Sensor Networks Continue to Puzzle: Selected Open Problems

Thomas Locher, Pascal von Rickenbach, and Roger Wattenhofer

Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland
{lochert, pascalv, wattenhofer}@tik.ee.ethz.ch

Abstract. While several important problems in the field of sensor networks have already been tackled, there is still a wide range of challenging, open problems that merit further attention. We present five theoretical problems that we believe to be essential to understanding sensor networks. The goal of this work is both to summarize the current state of research and, by calling attention to these fundamental problems, to spark interest in the networking community to attend to these and related problems in sensor networks.

Introduction

Algorithmic sensor network research has been around for almost a decade now,[1] and it has meanwhile reached a semi-mature state: Many essential questions have been studied; some exemplary ones such as, e.g., min-energy [1,2] and min-time [3,4,5] broadcasting or geo-routing [6,7,8] are understood to a pleasing degree, belying those who accuse the sensor networking community of not producing any rigid results.

However, sensor networks continue to puzzle as many fundamental aspects are not well understood; in this paper we present five brainteasers in the sensor network domain, covering various areas such as scheduling, topology control, clustering, positioning, and time synchronization. The five open problems have in common that they all pertain to data gathering, an important task in sensor networking. As it is often essential to know when and where data has been collected, the data needs to be enriched with time (Section 5) and position (Section 4) information. Additionally, the structure of the network has to be tuned in order to gather data in an energy-efficient manner. In Section 3 we save energy by turning off unneeded nodes, in Section 2 by reducing interference. Finally, in Section 1 we study the capacity of sensor networks, i.e., the achievable throughput of scheduling algorithms.

The five problems have in common that they all allow for a precise "zero parameters" definition. This is probably rare in a research area that still mostly revolves around the question which questions to ask. In that sense, these five problems are prototypical for an algorithmic approach to networking. However, primarily they have in common that the authors of this article are familiar with them. Our five open problems are by no means the most important problems that remain to be solved in the sensor network domain.

[1] Alas, there is no clear date of birth of this research area; however, some of the first workshops such as, e.g., DialM or MobiHoc were started about a decade ago.

We are sure that three other authors would come up with a completely different set of open problems, at least equally worthy of being studied. Nevertheless, we do believe that advancing the state of the art of any of the problems discussed in this paper will not only advance sensor networks but also networking and distributed computing in general.

1 Scheduling

Spatial reuse is fundamental in wireless networking. Due to channel interference, concurrent transmissions may hinder a successful reception at the intended destinations. Thus, it is vital to coordinate channel access in order to prevent collisions and to increase network throughput. The task of a scheduling algorithm is to order a given set of transmission requests such that the correct reception of messages is not prevented due to interference caused by concurrent transmissions. Apart from timing message transmissions, scheduling algorithms have another degree of freedom to optimize their schedule: They can adjust the transmission power for each message individually to fully benefit from spatial reuse in order to minimize the total time needed to successfully complete all requests. This is important since a successful message reception depends on the ratio between received signal strength on the one hand and interference and ambient noise on the other hand (also known as SINR).[2]

More formally, consider the network nodes $X = \{x_1, \ldots, x_n\}$. Furthermore, let P_r be the signal power received by a node x_r and let I_r denote the amount of interference generated by other nodes. Finally, let N be the ambient noise power level. Then, a node x_r receives a transmission if and only if $\frac{P_r}{N+I_r} \geq \beta$, where β denotes the minimum signal-to-noise-plus-interference ratio that is required for a message to be successfully received.

In wireless networks, the value of received signal power P_r is a decreasing function of the distance $d(x_s, x_r)$ between transmitter node x_s and receiver node x_r. More specifically, given the distance $d(x_s, x_r)$ between sender and receiver, the decay of the signal power is proportional to $d(x_s, x_r)^{-\alpha}$. The so-called path-loss exponent α is a constant between 2 and 6 and depends on external conditions of the medium, as well as the exact sender-receiver distance [9]. Let P_i be the power level assigned to node x_i. A message transmitted from a node $x_s \in X$ is successfully received by a node x_r if

$$\frac{\frac{P_s}{d(x_s,x_r)^\alpha}}{N + \sum_{x_i \in X \setminus \{x_s\}} \frac{P_i}{d(x_i,x_r)^\alpha}} \geq \beta.$$

In [10,11] the scheduling complexity of basic network structures, namely strongly connected networks, is studied. It is shown that adjusting the transmission power gives an exponential advantage over uniform or linear power

[2] The communication model adopting this notion of signal-to-noise-plus-interference ratio is also known as the physical model [9].

assignment schemes. This gives an interesting complement to the more pessimistic bounds for the capacity in wireless networks [9]. The authors of [12] define a measure called disturbance that comprises the intrinsic difficulty of finding a short schedule for a problem instance. Furthermore, they propose an algorithm that achieves provably efficient performance in any network and request setting that exhibits a low disturbance. For the special case of many-to-one communication with data aggregation in relaying nodes, [13] derives a scaling law describing the achievable rate in arbitrarily deployed sensor networks. It is show that for a large number of aggregation functions a sustainable rate of $1/\log^2 n$ can be achieved.

In the context of routing, [14] studies the problem of constructing end-to-end schedules for a given set of routing requests such that the delay is minimized. That is, each node is assigned a distinct power level, the paths for all requests are determined, and all message transmissions are scheduled to guarantee successful reception in the SINR model. In this setting, [14] presents a polynomial-time algorithm with provable worst-case performance for the problem.

Despite all the work discussed in this section considering transmission scheduling problems with specific constraints, the basic problem is still not fully understood.

Problem 1. *A communication request consists of a source and a destination, which are arbitrary points in the Euclidean plane. Given n communication requests, assign a color (time slot) to each request. For all requests sharing the same color specify power levels such that each request can be handled correctly, i.e., the SINR condition is met at all destinations. The goal is to minimize the number of colors.*

While uniform power assignment is understood well [15], it is unknown how difficult the problem is if nodes can adapt their transmission power. This is indisputably a most fundamental problem in the field of sensor networks. A deeper understanding of scheduling will potentially shed new light also on other advanced open problems.

2 Topology Control

Energy is a scarce resource in wireless sensor networks. In a very general way, topology control can be considered as the task of, given a network communication graph, constructing a subgraph with certain desired properties while minimizing energy consumption. The subgraph needs to meet some requirements, the minimum requirement being to maintain connectivity. However, sometimes one has stronger demands, e.g., the subgraph should not only be connected but a spanner of the original graph. At the same time the subgraph should be sparse as low node degrees allow for simpler neighborhood management at the nodes; additionally, symmetric links are desired as they permit simpler higher-layer protocols, and, if the constructed graph is planar, geo-routing protocols can be used. The most important goal however is energy-efficiency. Energy is saved by several means, the simplest being to eliminate distant neighbors, and thereby energy-inefficient connections, since the energy consumption of a transmission is believed to grow at least quadratically with distance.[3] Almost as a side effect, this reduction also results in less interference. Confining interference additionally lowers the power consumption by reducing the number of collisions and consequently the number of packet retransmissions on the media access layer.

Early work focused on topology control algorithms emphasizing locality while exhibiting more and more desirable properties [16,17,18,19], sometimes presenting distributed algorithms that optimize various design goals concurrently. All these approaches have in common, however, that they address interference reduction only implicitly. The intuition was that a low (constant) node degree at all nodes would solve the interference issue automatically. This intuition was proved wrong in [20], starting a new thread that explicitly studies interference reduction in the context of topology control [21,22,23]. The interference model introduced in [24] in the context of data-gathering structures, which is generalized in [25], proposes a natural way to define interference in sensor networks. The general question is: How can one connect the nodes such that as few nodes as possible disturb each other? In the following, we discuss the network and interference model presented in [25].

The wireless network is modeled as a geometric graph. The graph consists of a set of nodes represented by points in the Euclidean plane; we want to connect these nodes by choosing a set of edges. In order to prevent already basic communication between neighboring nodes from becoming unacceptably cumbersome [26], it is required that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. In other words, only undirected edges are considered. A node is able to adjust its transmission power to any value between zero and its maximum power level to reach other nodes. An edge exists if and only if the maximum transmission range of both incident nodes mutually include their counterpart. The minimum requirement of a topology control algorithm reducing transmission power levels is then to compute a subgraph of the given network graph that preserves connectivity. The interference of a node v is then defined as the number of other nodes that potentially affect message reception at node v.[4] The maximum interference of a graph is then defined as the maximum node interference.

So far, not many results have been published in the context of explicit interference minimization. For networks restricted to one dimension the authors in [25] present a $\sqrt[4]{n}$-approximation of the optimal connectivity-preserving topology that minimizes the maximum interference. For the two

[3] In sensor networks, one has to be careful about this model, as generally transmission distances are short, and the base transmission or even reception energy washes this quadratic behavior out.

[4] In practice, the shape of a node's interference region is not restricted to be circular. In particular, it depends on the antenna in use; the interference range is typically larger than the reception range.

dimensional case, the authors in [27] propose an algorithm that bounds the maximum interference to $O(\sqrt{n})$. If average interference of a graph is considered, there is an asymptotically optimal algorithm achieving an approximation ratio of $O(\log n)$ [28]. This leads us to the open problem:

Problem 2. *Given n nodes in the plane. Connect the nodes by a spanning tree. For each node v we construct a disk centering at v with radius equal to the distance to v's furthest neighbor in the spanning tree. The interference of a node v is then defined as the number of disks that include node v. Find a spanning tree that minimizes the maximum interference.*

This problem is still not understood well. We do not know the complexity of the problem (solvable optimally in polynomial time, or $\mathcal{NP}$-complete), and it is unknown whether efficient approximation algorithms exist. Once we understand interference, we can try to combine it with other optimization goals such as planarity or constant node degree. And once we understand these, we can start looking for distributed (or even local) algorithms for the problem. Furthermore, we can abandon the strict geometric representation of interference and think about more general interference models [28].

Clearly, there is a relation between Problem 2 and the scheduling problem studied in Section 1 [10], as in both problems the goal is to increase spatial reuse by understanding interference. However, we do not believe that solving one problem would help solving the other, as the scheduling problem allows for a more general power control approach. It was shown in [11] that there is an exponential difference between these two models. The next section is related to this one as well: The goal is also to reduce energy consumption, however with a different approach.

3 Dominating Set

An alternative method to ensure an efficient operation in dense graphs is to completely "shut down" a large fraction of all nodes and delegate their responsibilities to a few neighboring nodes. This is in stark contrast to the approach taken in topology control algorithms where all nodes continue to handle messages themselves. Naturally, it must be guaranteed that every node has a neighbor that is in the position to take over its tasks. Ideally, this set of nodes that remain awake and handle all tasks is as small as possible in order to minimize energy consumption. New sets of nodes that must stay awake can be constructed periodically in order to even out the burden of communication among all nodes in the network.

More formally, we again model the network as a graph where edges between nodes indicate that these nodes can communicate directly. A set of nodes S for which it holds that every node that is not in S has a direct neighbor in S has to be found. Such a set is commonly referred to as a *dominating set.* The goal of the *minimum dominating set* (MDS) problem is to find the dominating set of minimum size. For certain applications, it is mandatory or at least beneficial if the nodes in the dominating set are connected. Thus, a variation of the MDS problem is the problem of finding a minimum *connected dominating set*.

Computing a minimum dominating set is a hard problem. It has been shown that the MDS problem is $\mathcal{NP}$-complete not only for arbitrary graphs [29], but also for special topologies such as *unit disk graphs* (UDGs) [30,31]. Moreover, dominating sets cannot be approximated in polynomial time to within a factor of $(1 - o(1)) \ln n$ [32] unless $\mathcal{NP}$ has quasi-polynomial time algorithms. However, this bound only holds for general graphs, and in various special cases, constant approximations can be computed efficiently. For example, there is a simple constant approximation algorithm for dominating sets in UDGs [33]. Note that a DS can trivially be extended to a connected dominating set by means of a spanning tree with only a constant overhead. This result has been generalized in [34], where it is shown that a constant-factor approximation is even possible if all nodes are *weighted*, and the goal is to find a (connected) dominating set that minimizes the sum of the weight of all nodes in the dominating set. In the unweighted case, there is a PTAS for the minimum dominating set problem in unit disk graphs [35].

Distributed algorithms for the MDS problem have also been studied extensively. The algorithms in the following papers belong to the class of *local algorithms* in which all nodes are allowed to communicate k times, for a particular value k, with their neighboring nodes. In this model, nodes can basically gather information about nodes in their k-neighborhood and can thus base their decisions on this information only. Similarly to the centralized case, it has also been shown that once a dominating set has been built, this set can be used to construct a connected dominating set in a distributed fashion [36].

In general graphs, a *maximum independent set* (MIS) can be constructed using a randomized algorithm in $O(\log n)$ time [37]. Naturally, a MIS is also a dominating set, but the constructed MIS does not guarantee any bounds on the approximation ratio. The algorithm presented in [38] computes an $O(\log \Delta)$-approximation in $O(\log n \log \Delta)$ rounds with high probability, where Δ denotes the maximum node degree. The first constant-time distributed algorithm achieving a non-trivial approximation ratio is presented in [39]: An $O(k\Delta^{2/k} \log \Delta)$-approximation is computed in $O(k^2)$ rounds for an arbitrary (constant) k. By setting $k = \Theta(\log \Delta)$, the algorithm achieves an approximation ratio of $O(\log^2 \Delta)$ in $O(\log^2 \Delta)$ rounds. This result was later improved to an $O(\log \Delta)$-approximation algorithm also requiring $O(\log^2 \Delta)$ rounds [40].

There has also been a lot of work on computing dominating or maximum independent sets in unit disk graphs. Note that in unit disk graphs a maximum independent set is a good approximation of the optimal dominating set, thus the two problems are basically equivalent. A PTAS for UDGs is also achievable by means of a local algorithm [41]. If the nodes know the distance to all other nodes, a MIS can be constructed in $O(\log^* n)$ time in unit disk graphs and also in a large class of bounded independence graphs [42], which matches a MIS lower bound of $\Omega(\log^* n)$ [43]. The fastest *deterministic* algorithm for the MIS problem in unit disk graphs—in fact, in any growth-bounded graph—requires $O(\log \varDelta \log^* n)$ time [44] to construct a MIS. A MIS can be constructed faster using a randomized algorithm whose running time is only $O(\log \log n \log^* n)$ with high probability [45].

It is, however, still unclear if a dominating set that is only a constant factor larger than the smallest possible dominating set can be constructed very quickly in unit disk graphs.

Problem 3. *Let each node in a unit disk graph know its k-neighborhood for a constant k, i.e., each node knows all nodes up to distance k including their interconnections. Given this information, each node must decide locally without any further communication whether it joins the dominating set or not. Is it possible to construct a valid dominating set that is only a constant factor larger than the optimal dominating set?*

While there are lower bounds to find a MIS or a coloring, there is no lower bound for the MDS problem. It is unclear if a constant-time algorithm can compute a dominating set in UDGs, and conversely if a constant-factor approximation requires $\omega(1)$ time. There are many related open problems such as the problem of finding a MIS or a coloring with a small approximation ratio as quickly as possible.

4 Embedding

Many envisioned application scenarios in the field of wireless sensor networks rely on positioning information: sensing the environment is only useful if one knows where the data has actually been measured. Knowledge of location information can also improve the performance of routing algorithms because it allows the use of geo-routing techniques [6,7]. Equipping all sensor nodes with specific hardware such as GPS receivers would be one option to gain position information at the nodes. However, GPS reception might be obstructed by climatic conditions or in-door environments. Another solution is to provide only a few nodes (so-called anchor or landmark nodes) with GPS and have the rest of the nodes compute their position by using the known coordinates of the anchor nodes [46,47]. One characteristic inherent to all these approaches is that the solution quality is determined by the anchor density and their actual placement.

Obviously, in the absence of anchors, nodes are clueless about their real coordinates. However, recent work has pointed out that for many applications it is not necessary to have real coordinates but it suffices to have virtual coordinates—two nodes having similar coordinates implies that they are physically close together. Moreover, a deeper understanding of anchorless positioning would likely advance the state of the art of anchor-based positioning algorithms. A mapping of all the nodes to virtual coordinates, in this case coordinates in the Euclidean plane, is called an *embedding*.

Sensor networks are typically modelled as *unit disk graphs* in which there is an edge between two nodes if and only if the Euclidean distance between them is less or equal to 1. It has been shown that the problem of deciding whether a given graph is a unit disk graph is $\mathcal{NP}$-hard [48]. A more general model for sensor networks is given by *d-quasi unit disk graphs.* A graph is called a d-quasi unit disk graph (d-QUDG, $d \leq 1$) if there is an embedding that respects the following two rules: If two nodes are connected, the distance between their respective coordinates must be at most 1, and if there is not edge between two nodes, the distance between their coordinates must exceed d. Note that a 1-QUDG corresponds to a UDG graph and that the definition of a d-QUDG does not specify whether there is an edge between two nodes at a distance in the range $(d, 1]$ for $d < 1$. In that sense, a d-QUDG is a relaxed version of a UDG. A QUDG can generally be regarded as a more realistic model for sensor networks since nodes at a critical distance may or may not be able to communicate. The quality $q(e)$ of an embedding e in this model is defined as

$$q(e) = \frac{\max_{\{u,v\} \in E} dist(u, v)}{min_{\{u,v\} \notin E} dist(u, v)}.$$

A good embedding has a *small* value for its quality. It has been shown that it is also $\mathcal{NP}$-hard to find an embedding such that $q(e) < \sqrt{3/2}$ [49]. In the same work, it has further been proven that it is $\mathcal{NP}$-hard to decide whether a graph can be realized as a d-quasi unit disk graph with $d \geq 1/\sqrt{2}$. Surprisingly, the problem remains hard even if additional information is available. For example, each node might know the exact distance to each of its neighbors. Given this distance information, it is still $\mathcal{NP}$-hard to decide whether the graph is a UDG or not [50]. Instead of having distance information, the nodes might be aware of the angle between itself and any two of its neighbors. The problem remains $\mathcal{NP}$-hard also in this context [51].

In [52], the first approximation algorithm for this problem is presented, which heavily borrows techniques introduced by Vempala [53], claiming an $O(\log^{2.5} n\sqrt{\log \log n})$-quality embedding in polynomial running time.[5] The currently best known algorithm for this problem is due to Pemmaraju and Pirwani [54], which computes a $O(\log^{2.5} n)$-quality embedding of a given unit disk graph.

In practice, many heuristics are used to compute embeddings efficiently. Various approaches based on, e.g., distance measurements [55], using eigenvectors [56] or linear programming [51] etc. have been shown to produce acceptable results. Still, in theory the problem is not well understood.

Problem 4 *Given the adjacency matrix of a unit disk graph, find positions for all nodes in the Euclidean plane such that the ratio between the maximum distance between any two adjacent nodes and the minimum distance between any two non-adjacent nodes is as small as possible.*

Apparently, there is a large gap between the best known lower bound, which is a constant, and the polylogarithmic upper bound. It is a challenging

[5] A subsequent paper [54] corrects the bound on the quality to $O(\log^{3.5} n\sqrt{\log \log n})$.

task to either come up with a better approximation algorithm or prove a stronger lower bound.

5 Time Synchronization

Many protocols require that the participants be closely *synchronized* in order to guarantee an efficient and successful execution. It is therefore mandatory to provide a distributed clock synchronization algorithm whose objective is to ensure that the nodes are able to acquire a common notion of time. As the state of the system is distributed, the participating nodes can synchronize their clocks by exchanging messages with their neighboring nodes and thereby learn about the current state of other nodes.

We consider distributed clock synchronization algorithms in the following setting. Given an arbitrary graph $G = (V, E)$ in which nodes can communicate directly with all other node to which they are directly connected in G. The nodes that are directly connected to a node v are referred to as the neighboring nodes of v. The communication between neighboring nodes is assumed to be reliable, but all messages can have variable delays in the range $[0, 1]$. The *distance* between nodes i and j is defined as the length of the shortest path between i and j, and the *diameter* D of G is the maximum distance between any two nodes.

We assume that each node is equipped with a *hardware clock* $H(\cdot)$ whose value at time t is $H(t) := \int_0^t h(\tau)\ d\tau$, where $h(\tau)$ is the *hardware clock rate* at time τ. Furthermore, we make the assumption that the hardware clocks have bounded drift, i.e., there is a constant $0 \leq \epsilon < 1$ such that $1-\epsilon \leq h(t) \leq 1+\epsilon$ at all times t.

In addition to the hardware clock, each node i is further equipped with a second, so-called *logical clock* $L(\cdot)$. The logical clock also increases steadily, just like the hardware clock, but potentially at a different rate. However, the deviation between the hardware and the logical clock rate is lower and upper bounded by specific constants, e.g., the logical clock rate has to be at least half and at most twice the hardware clock rate at any given time. This restriction ensures that the logical clock can neither be slowed down nor sped up arbitrarily, which would trivialize the problem and destroy the relation between the hardware and the logical clock.

Due to different clock rates the hardware clocks of different nodes might drift apart. As the hardware clocks cannot be manipulated, the goal is therefore to minimize the clock skew of the logical clocks. At any point in time, a node may inform its neighboring nodes about its current logical time. A node receiving such an update can decide to increase its own logical clock in order to counterbalance the skew between the clocks. However, the logical clock is not allowed to run backwards.

A desirable goal is to guarantee that the clock skew between any two nodes in the network is as small as possible. The bound achievable for this goal is denoted the *global property* of the clock synchronization algorithm. It can be shown that the skew between two nodes at distance d cannot be synchronized better than $\Omega(d)$ by using simple indistinguishability type arguments. Srikanth and Toueg [57] presented a clock synchronization algorithm, which is asymptotically optimal in the sense that it guarantees a clock skew of at most $O(D)$ between any two nodes in a network of diameter D. However, there are executions of this algorithm causing a clock skew of $\Theta(D)$ even between neighboring nodes.

For several distributed applications, such as, e.g., media access control or event detection, it is mandatory that the clocks between any node and particularly all nodes in its vicinity are closely synchronized. This is known as the *gradient property* of the algorithm that requires a minimal clock skew between all neighboring nodes. This property was introduced in [58] where a surprising lower bound on the worst-case clock skew of $\Omega(\frac{\log D}{\log \log D})$ between neighboring nodes is proven. If the logical clocks are allowed to remain constant for a certain period of time, the clock skew between neighboring nodes can in fact be kept constant [59]. In general, the best known clock synchronization algorithm with a non-trivial gradient property guarantees that the worst-case skew between any two neighbors at distance d is at most $O(d + \sqrt{D})$ [60]. Obviously, the gap between the lower and the upper bound is still fairly large and the goal is to close this gap.

Problem 5 *Nodes in an arbitrary graph are equipped with an unmodifiable hardware clock and a modifiable logical clock. The logical clock must make progress roughly at the rate of the hardware clock, i.e., the clock rates may differ by a small constant. Messages sent over the edges of the graph have delivery times in the range* $[0, 1]$*. Given a bounded, variable drift on the hardware clocks, design a message-passing algorithm that ensures that the logical clock skew of adjacent nodes is as small as possible at all times.*

The algorithm that guarantees a skew of $O(\sqrt{D})$ [60] between neighboring nodes requires that a large amount of messages are sent. Another natural question is whether a good gradient property can also be ensured if bounds on the message complexity are imposed. Further future work might include faulty or even byzantine nodes which deliberately try to hinder the correct nodes from synchronizing their clocks.

6 Conclusions

In this paper, we presented five open problems in the field of sensor networks, all with an algorithmic flavor. Craving for progress, we offer a bag of Swiss chocolate to anybody who solves one of our problems. As stated before, our selection is rather random, and other authors for sure would promote other problems at least equally worthy of being studied. Actually, we would also be quite keen to learn about these other problems and encourage you to tell us about them. An official repository of open problems could ignite a fresh way of organizing research in this area—a way that actually uses the Internet—and could help keeping track of progress.

References

1. Wan, P.J., Calinescu, G., Li, X.Y., Frieder, O.: Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks. In: Proc. of the 20^{th} Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). (2001)
2. Ambühl, C.: An Optimal Bound for the MST Algorithm to Compute Energy Efficient Broadcast Trees in Wireless Networks. In: Proc. of the 32^{th} Int. Colloquium on Automata, Languages and Programming (ICALP). (2005)
3. Alon, N., Bar-Noy, A., Linial, N., Peleg, D.: A Lower Bound for Radio Broadcast. J. Comput. Syst. Sci. **43**(2) (1991) 290–298
4. Czumaj, A., Rytter, W.: Broadcasting Algorithms in Radio Networks with Unknown Topology. J. Algorithms **60**(2) (2006) 115–143
5. Peleg, D.: Time Efficient Broadcasting in Radio Networks: A Review. In: Proc. of the 4^{th} 4th Int. Conf. on Distributed Computing and Internet Technology (ICDCIT). (2007)
6. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with Guaranteed Delivery in ad hoc Wireless Networks. In: Proc. of the 3^{rd} Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M). (1999)
7. Kuhn, F., Wattenhofer, R., Zollinger, A.: Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In: Proc. of the 4^{th} ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2003)
8. Durocher, S., Kirkpatrick, D., Narayanan, L.: On Routing with Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks. Personal communication (2007)
9. Gupta, P., Kumar, P.R.: The Capacity of Wireless Networks. IEEE Trans. Information Theory **46**(2) (2000) 388–404
10. Moscibroda, T., Wattenhofer, R.: The Complexity of Connectivity in Wireless Networks. In: Proc. of the 25^{th} Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). (2006)
11. Moscibroda, T., Wattenhofer, R., Zollinger, A.: Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies. In: Proc. of the 7^{th} ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (May 2006)
12. Moscibroda, T., Oswald, Y.A., Wattenhofer, R.: How Optimal are Wireless Scheduling Protocols? In: Proc. of the 26^{th} Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). (2007)
13. Moscibroda, T.: The Worst-Case Capacity of Wireless Sensor Networks. In: Proc. of the 6^{th} Int. Conf. on Information Processing in Sensor Networks (IPSN). (2007)
14. Chafekar, D., Kumar, V.S.A., Marathe, M., Pathasarathy, S., Srinivasan, A.: Cross-Layer Latency Minimization in Wireless Networks with SINR constraints. In: Proc. of the 8^{th} ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2007)
15. Goussevskaia, O., Oswald, Y.A., Wattenhofer, R.: Complexity in Geometric SINR. In: Proc. of the 8^{th} ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2007)
16. Wattenhofer, R., Li, L., Bahl, P., Wang, Y.M.: Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In: Proc. of the 20^{th} Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). (2001)
17. Santi, P.: Topology Control in Wireless Ad Hoc and Sensor Networks. Wiley (2005)
18. Li, X.Y., Song, W.Z., Wan, W.: A Unified Energy Efficient Topology for Unicast and Broadcast. In: Proc. of the 11^{th} Int. Conf. on Mobile Computing and Networking (MOBICOM). (2005)
19. Damian, M., Pandit, S., Pemmaraju, S.: Local Approximation Schemes for Topology Control. In: Proc. of the 25^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC). (2006)
20. Burkhart, M., von Rickenbach, P., Wattenhofer, R., Zollinger, A.: Does Topology Control Reduce Interference? In: Proc. of the 5^{th} ACM Int. Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc). (2004)

21. Moaveni-Nejad, K., Li, X.Y.: Low-Interference Topology Control for Wireless Adhoc Networks. Ad Hoc & Sensor Wireless Networks: An International Journal **1**(1-2) (2005)
22. Johansson, T., Carr-Motyčková, L.: Reducing Interference in Ad hoc Networks through Topology Control. In: Proc of the *3*rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC). (2005)
23. Benkert, M., Gudmundsson, J., Haverkort, H., Wolff, A.: Constructing Interference-Minimal Networks. Computational Geometry: Theory and Applications (2007)
24. Fussen, M., Wattenhofer, R., Zollinger, A.: Interference Arises at the Receiver. In: Proc. of the Int. Conf. on Wireless Networks, Communications, and Mobile Computing (WIRELESSCOM). (2005)
25. von Rickenbach, P., Schmid, S., Wattenhofer, R., Zollinger, A.: A Robust Interference Model for Wireless Ad-Hoc Networks. In: Proc. of the *5*th Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN). (2005)
26. Prakash, R.: Unidirectional Links Prove Costly in Wireless Ad-hoc Networks. In: Proc. of the *3*rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM). (1999)
27. Halldórsson, M.M., Tokuyama, T.: Minimizing Interference of a Wireless Ad-Hoc Network in a Plane. In: Proc. of the *2*nd Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks(ALGOSENSORS). (2006)
28. Moscibroda, T., Wattenhofer, R.: Minimizing Interference in Ad Hoc and Sensor Networks. In: Proc. of the *3*rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC). (2005)
29. Karp, R.: Reducibility Among Combinatorial Problems. J. ACM **45**(4) (1998) 634–652
30. Masuyama, S., Ibaraki, T., Haseqawa, T.: The Computational Complexity of the M-Center Problems in the Plane. Trans. IECE Japan E64 (1981) 57–64
31. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit Disk Graphs. Discrete Math. **86**(1-3) (1990) 165–177
32. Feige, U.: A Threshold of ln n for Approximating Set Cover. J. ACM **45**(4) (1998) 634–652
33. Marathe, M., Breu, H., Ravi, H., Rosenkrantz, D.: Simple Heuristics for Unit Disk Graphs. Networks **25** (1995) 59–68
34. Ambühl, C., Erlebach, T., Mihalak, M., Nunkesser, M.: Constant-Factor Approximation for Minimum-Weight (Connected) Dominating Sets in Unit Disk Graphs. In: Proc. of the *9*th Int. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX). (2006)
35. Nieberg, T., Hurink, J.L.: A PTAS for the Minimum Dominating Set Problem in Unit Disk Graphs. Memorandum 1732, Universiteit Twente, Dep't of Applied Mathematics (2004)
36. Dubhashi, D., Mei, A., Panconesi, A., Radhakrishnan, J., Srinivasan, A.: Constant-Factor Approximation for Minimum-Weight (Connected) Dominating Sets in Unit Disk Graphs. In: Proc. of the *9*th Int. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX). (2006)
37. Luby, M.: A Simple Parallel Algorithm for the Maximal Independent Set Problem. SIAM Journal on Computing **15** (1986) 1036–1053
38. Jia, L., Rajaraman, R., Suel, R.: An Efficient Distributed Algorithm for Constructing Small Dominating Sets. In: Proc. of the *20*th ACM Symposium on Principles of Distributed Computing (PODC). (2001)
39. Kuhn, F., Wattenhofer, R.: Constant-Time Distributed Dominating Set Approximation. In: Proc. of the *22*nd ACM Symposium on Principles of Distributed Computing (PODC). (2003)
40. Kuhn, F., Moscibroda, T., Wattenhofer, R.: The Price of Being Near-Sighted. In: Proc. of the *17*th ACM-SIAM Symposium on Discrete Algorithms (SODA). (2006)
41. Kuhn, F., Moscibroda, T., Nieberg, T., Wattenhofer, R.: Local Approximation Schemes for Ad Hoc and Sensor Networks. In: Proc. of the *3*rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC). (2005)

42. Kuhn, F., Moscibroda, T., Wattenhofer, R.: On the Locality of Bounded Growth. In: Proc. of the *24th* ACM Symposium on Principles of Distributed Computing (PODC). (2005)
43. Linial, N.: Locality in Distributed Graph Algorithms. SIAM Journal on Computing **21**(1) (1992) 193–201
44. Kuhn, F., Moscibroda, T., Nieberg, T., Wattenhofer, R.: Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs. In: Proc. of the *19th* Int. Symposium on Distributed Computing (DISC). (2005)
45. Gfeller, B., Vicari, E.: A Randomized Distributed Algorithm for the Maximal Independent Set Problem in Growth-Bounded Graphs. In: Proc. of the *26th* ACM Symposium on Principles of Distributed Computing (PODC). (2007)
46. Savvides, A., Han, C.C., Strivastava, M.B.: Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In: Proc. of the *7th* Annual Int. Conf. on Mobile Computing and Networking (MOBICOM). (2001)
47. O'Dell, R., Wattenhofer, R.: Theoretical Aspects of Connectivity-based Multi-hop Positioning. Theoretical Computer Science **344**(1) (2005) 47–68
48. Breu, H., Kirkpatrick, D.: Unit Disk Graph Recognition is NP-hard. Computational Geometry: Theory and Applications **9**(1–2) (1998) 3–24
49. Kuhn, F., Moscibroda, T., Wattenhofer, R.: Unit Disk Graph Approximation. In: Proc. of the *2nd* ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC). (2004)
50. Aspnes, J., Goldenberg, D., Yang, Y.R.: On the Computational Complexity of Sensor Network Localization. In: Proc. of the *1st* Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS). (2004)
51. Bruck, J., Gao, J., Jiang, A.: Localization and Routing in Sensor Networks By Local Angle Information. In: Proc. of the *6th* ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2005)
52. Moscibroda, T., O'Dell, R., Wattenhofer, M., Wattenhofer, R.: Virtual Coordinates for Ad hoc and Sensor Networks. In: Proc. of the *2nd* ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC). (2004)
53. Vempala, S.: Random Projection: A New Approach to VLSI Layout. In: Proc. of the *39th* Annual Symposium on Foundations of Computer Science (FOCS). (1998)
54. Pemmaraju, S., Pirwani, I.A.: Good Quality Virtual Realization of Unit Ball Graphs. In: Proc. of the *15th* Annual European Symposium on Algorithms (ESA). (2007)
55. Priyantha, N.B., Balakrishnan, H., Demaine, E.D., Teller, S.: Anchor-free Distributed Localization in Sensor Networks. Technical Report TR-892, MIT, LCS (2003)
56. Gotsman, C., Koren, Y.: Distributed Graph Layout for Sensor Networks. Pach, J. (ed.) GD 2004, LNCS **3383** (2005)
57. Srikanth, T.K., Toueg, S.: Optimal Clock Synchronization. J. ACM **34**(3) (1987) 626–645
58. Fan, R., Lynch, N.: Gradient Clock Synchronization. In: Proc. of the *23rd* Annual ACM Symposium on Principles of Distributed Computing (PODC). (2004)
59. Fan, R., Chakraborty, I., Lynch, N.: Clock Synchronization for Wireless Networks. In: Proc. of the *8th* Int. Conf. on Principles of Distributed Systems (OPODIS). (2004)
60. Locher, T., Wattenhofer, R.: Oblivious Gradient Clock Synchronization. In: Proc. of the *20th* Int. Symposium on Distributed Computing (DISC). (2006)