Wireless Netw (2008) 14:715–729
DOI 10.1007/s11276-007-0045-6

Ad hoc networks beyond unit disk graphs

Fabian Kuhn · Roger Wattenhofer · Aaron Zollinger

Published online: 13 July 2007

Abstract In this paper, we study an algorithmic model for wireless ad hoc and sensor networks that aims to be sufficiently close to reality as to represent practical real-world networks while at the same time being concise enough to promote strong theoretical results. The quasi unit disk graph model contains all edges shorter than a parameter d between 0 and 1 and no edges longer than 1. We show that—in comparison to the cost known for unit disk graphs—the complexity results of geographic routing in this model contain the additional factor $1/d^2$. We prove that in quasi unit disk graphs flooding is an asymptotically message-optimal routing technique, we provide a geographic routing algorithm being most efficient in dense networks, and we show that classic geographic routing is possible with the same asymptotic performance guarantees as for unit disk graphs if $d \geq 1/\sqrt{2}$.

Keywords Algorithmic analysis · Cost metrics · Geographic routing · Network models · Wireless ad hoc networks

F. Kuhn
Microsoft Research, Mountain View, CA 94043, USA
e-mail: kuhn@microsoft.com

R. Wattenhofer
ETH Zurich, 8092 Zurich, Switzerland
e-mail: wattenhofer@tik.ee.ethz.ch

A. Zollinger (✉)
Department of Electrical Engineering and Computer Science, University of California, 513 Soda Hall #1776, Berkeley, CA 94720-1776, USA
e-mail: zolling@eecs.berkeley.edu

1 Introduction

One manifestation of the currently observed and continuing miniaturization of electronics in general and wireless communication technology in particular is mobile ad hoc *networks*. Ad hoc networks are formed by mobile devices consisting of, among other components, a processor, some memory, a radio communication unit, and a power source, due to physical constraints commonly a weak battery or a small solar cell.

Typically, wireless ad hoc networks are intended to be employed where no communication infrastructure is present before the deployment of the ad hoc network or where reliance on previously available infrastructure is not desired or not possible. Common scenarios for ad hoc networks include communication among rescue teams, police squads, or during fire fighting or other disaster relief actions. Another often mentioned scenario involves cars forming an ad hoc network for professional, entertainment, or informational purposes. Car-mounted radio broadcast warning systems automatically alerting approaching automobiles of accidents or other unexpected traffic events are frequently envisioned. Also for meetings or conferences ad hoc networks may have their value. Furthermore, ad hoc networks may find their application for security and—inevitably—for military purposes.[1]

Sensor networks can be considered a specialization of ad hoc networks in which nodes are equipped with sensors measuring certain physical values, such as humidity, brightness, temperature, acceleration, or vibration. Usually, the sensor nodes are designed to report measured information to a data sink node. Among the most common

[1] This raises the interesting question whether there actually exists any technology that cannot be applied for military purposes.

scenarios for sensor networks are environmental monitoring tasks, for instance to warn of imminent natural disasters or for the purpose of biological or other scientific observations. Typically, sensor networks will be deployed in areas difficult to access or, more generally, where human presence or stationary monitoring infrastructure is undesired or impossible.

A widely employed model for the study of ad hoc and sensor networks is the so-called *unit disk graph* (UDG) model: Nodes are located in the Euclidean plane and are assumed to have identical (unit) transmission radii. Consequently an edge between two nodes—representing that they are in mutual transmission range—exists if and only if their Euclidean distance is not greater than one. Accordingly, a unit disk graph models a flat environment with network devices equipped with wireless radio, all having equal transmission ranges. Edges in the UDG correspond to radio devices positioned in direct mutual communication range. On the one hand, clearly, this is a glaring simplification of reality, since, even if all network nodes are homogeneous, this model does not account for the presence of obstacles, such as walls, buildings, mountains—or also weather conditions—which might obstruct signal propagation. On the other hand, unit disk graphs are simple enough to allow of strong theoretical results (see Sect. 2).

In this paper we study a graph model, originally introduced in [5], which is considerably closer to reality. We maintain the assumption that all mobile nodes are placed in the plane (that is, they have coordinates in $\mathbb{R}^2$). In a *quasi unit disk graph*, two nodes are connected by an edge if their distance is at most d, d being a parameter between 0 and 1. Furthermore, if the distance between two nodes is greater than 1, no edge exists between them. In the range between d and 1, the existence of an edge is not specified.

In contrast to unit disk graphs, the quasi unit disk graph model aims to account for the possible presence of obstacles to radio signal propagation: Nodes having a distance larger than d but at most 1 can or cannot be connected by an edge, depending on how unhinderedly the signals propagate between the two nodes. This model however does not account explicitly for such obstacles; it is basically a formalization of the fact that a given node can communicate with sufficiently close nodes while it cannot do so with distant nodes, where ''close'' and ''distant'' are characterized in a more general way than with the unit disk graph's single sharp threshold value of exactly one unit of length. In other words, the quasi unit disk graph relies on two guarantees: Edges of length at most d are always present; edges of length greater than 1 never exist.

We are aware that the quasi unit disk graph model is a static communication model that does not take into consideration interference among transmitting nodes. Recent work shows that explicit consideration of interference among nodes leads to a considerably different class of algorithmic problems (see Sect. 2). This paper in contrast focuses on and confines itself to the study of a generalization of the unit disk graph model that is at the same time also a closer representation of reality.

We first establish a constructive lower bound for quasi unit disk graphs showing that basically any algorithm without routing tables requires sending of $\Omega((\frac{c}{d})^2)$ messages to route from a source s to a destination t, where c is the length of the shortest path between s and t. We show that, with the aid of a topology control graph structure, a restricted flooding algorithm is guaranteed not to perform worse and that this technique is consequently asymptotically message-optimal.

A more subtle approach than flooding of the network is possible if we additionally make the basic assumptions of *geographic routing*—that is attributing the network nodes with information about their own and their neighbors' positions and assuming that the message source knows the position of the destination. We present a combination of greedy routing and restricted flooding. This yields a routing algorithm that is still asymptotically optimal in the worst case but also efficient in the average case, as previous work on average-case efficiency of geographic ad hoc routing algorithms suggests [10, 44, 47]. Finally we show that, if we assume d to be at least $1/\sqrt{2}$, it is possible to locally introduce virtual edges and perform the classic variations of geographic routing while preserving performance guarantees known from unit disk graphs.

After discussing related work in the following section, we state the model and provide definitions in Sect. 3. In Sect. 4, we establish a lower bound for the message complexity of so-called volatile memory routing algorithms. Section 5 contains the description of the topology control structure forming the basis for the subsequent algorithms. Section 6 provides the analysis of flooding algorithms with respect to message and time complexity. Section 7 discusses the combination of flooding with a greedy approach for geographic routing, whereas Sect. 8 shows that for large enough d, classic geographic routing can be employed. Section 9 draws the conclusions of the paper.

2 Related work

So far, the most popular network structure to model mobile ad hoc networks has been the unit disk graph. The underlying assumption of this model is that the nodes are placed in the plane, all of them having the same transmission range—normalized to a radius of one unit of length. In the unit disk graph model a considerable number of theoretical results have been found with respect to aspects of wireless ad hoc and sensor networks as diverse as topology control

[11, 20, 70, 72, 73], the construction of dominating sets [2, 22, 23, 26, 28, 37, 43, 71], network initialization [40, 55, 56] and deployment [54], geographic routing [44, 45, 47], or positioning [60].

A more general model is provided by disk graphs. In contrast to unit disk graphs, disk graphs allow nodes to have different transmission ranges. Besides unit disk graphs, also disk graphs have been widely used to model wireless networks. However, while for unit disk graphs a large number of theoretical results have been achieved, most of the knowledge on disk graphs is based on simulations, [52] being the exception that proves the rule. If disk graphs provide a simple method to analyze unidirectional links, it is however not possible to model any kind of obstacles using this model.

Another example of a more general modeling of wireless networks are growth-bounded graphs [36, 38, 42], basically capturing and generalizing the observation that a wireless network node usually cannot have an arbitrary number of non-neighboring neighbors.

Yet another approach to the modeling of wireless networks explicitly considers interference among nodes. While some work tries to characterize interference by static UDG-based models [11, 20, 57, 70], other researchers focus on the interplay of scheduling and transmission power based on a realistic physical signal propagation model [13, 15, 16, 24, 27, 34, 53, 58, 59]. A more general and thorough analysis and comparison of different approaches to algorithmic modeling of wireless ad hoc and sensor networks can be found in [66].

In this paper, we go beyond unit disk graphs by allowing that certain sufficiently long edges may or may not exist in the considered network graph. A model has been described in [5] which is—up to scaling—identical to our quasi unit disk graph model. The authors of [5] focused on geographic routing with guaranteed message delivery for certain instances of the quasi unit disk graph model. In [46] and this paper—in particular in Sect. 8—we generalize and extend these results towards algorithm efficiency. The quasi unit disk graph model was later studied in the context of network deployment [39] and graph embedding [41].

In this paper, we give different complexity results concerning the quasi-UDG model. We show how to construct a subgraph of the network graph G which enables cost-optimal flooding and we show how the flooding overhead can be reduced (in practice) by using geographic routing for $d \geq 1/\sqrt{2}$ and a combination of geographic routing and flooding for arbitrary d. Constructing a sub-structure G' of G such that G' features some desirable properties is often termed topology control. Topology control is used to reduce the number of nodes and the number of edges involved in protocols such as routing. An important issue of such precomputation is the reduction of interference effects, message complexity, or energy consumption. Sometimes, algorithms need network graphs with special properties; for example all face-routing based geographic routing algorithms [10, 30, 35, 44, 45, 47] need a planar graph to operate correctly. For unit disk graphs, a number of different ideas in order to reduce the complexity of the network topology have been proposed. Many of them are based on dominating sets [2, 22, 23, 26, 28, 37, 43, 71], angle of arrival [73], or geographic clustering [19]. For finding planar subgraphs of unit disk graphs, various constructions of different quality and complexity have been conceived [7, 21, 23, 51, 68]. Other proposals strive to combine more and more beneficial complexity-reducing properties [72], while yet others focus on interference reduction [11, 20, 70] in unit disk graphs. Surveys on topology control algorithms for ad hoc networks in general can be found in [64] and [65].

Flooding—an essential ingredient of many ad hoc routing algorithms, such as DSR [29] or AODV [63]—is one of the main techniques employed in this paper. It is therefore crucial to reduce the number of messages sent in this process. One way to reduce the cost of flooding is to lower the complexity of the network by using appropriate topology control mechanisms, the approach chosen in this paper. Apart from this, there are other approaches which try to optimize flooding performance by using geographic information about the destination [6, 33]. These algorithms differ from the greedy routing/flooding approach presented in this paper in that they only try to flood into the right direction without actually applying geographic routing whenever possible.

Geographic routing (also known as location-based, position-based, or geometric routing) has also mainly been studied in unit disk graphs. Greedy routing algorithms have been studied in [17, 25, 35, 67] (an early approach combining greedy routing and flooding techniques being studied in [17]). Greedy routing behaves well in practice, but no guarantee can be given about the arrival of messages. The first algorithm with guaranteed delivery was *Face Routing* in [35] (called *Compass Routing II* there). *Face Routing* walks along faces of planar graphs and proceeds along the line connecting the source and the destination. Besides guaranteeing to reach the destination, this algorithm does so with O(n) messages, where n is the number of network nodes. However, 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. There have been later suggestions for algorithms with guaranteed message delivery [10, 14]; at least in the worst case, however, none of them outperforms the original Face Routing. Yet other geographic routing algorithms have been shown to reach the destination in special planar graphs without any or restricted runtime guarantees [8, 9, 23]. A more detailed overview of geographic routing can be found in [69].

The first geographic routing algorithm whose cost is bounded by a function of the cost c of an optimal path is *Adaptive Face Routing (AFR)* as introduced in [45]. It was also shown that this is the worst-case optimal result any geographic routing algorithm can achieve. Face Routing and also AFR are however not applicable for practical purposes due to their strict employment of face traversal. There have been proposals for practical purposes to combine greedy routing with face routing [10, 14, 30], however without competitive worst-case guarantees. The GOAFR algorithm (*Greedy Other Adaptive Face Routing*) introduced in [47] was the first algorithm to combine greedy and face routing in a worst-case optimal way; the GOAFR$^+$ algorithm [44] remains asymptotically worst-case optimal while improving GOAFR's efficiency in average-case networks.

Lately, first experiences with geographic and in particular face routing in practical networks have been made [31, 32]. More specifically, problems in connection with graph planarization that can occur in practice were observed, documented, and tackled. Another strand of research approaches these issues by allowing the routing algorithm to store certain limited information in the network nodes [48, 49].

Geographic routing is based on two assumptions: First, that every network node knows its own and its direct neighbors' current positions, and second, that the source of a message is informed about the current location of the destination. Above all the second assumption is algorithmically interesting and has been conceptually realized and analyzed in the context of so-called *location services*. Among the most prominent examples of location services are the GLS Grid Location System [50], which describes a hierarchical system of location information servers allowing for efficient position lookup in many cases, the Locality-Aware Location Service LLS [1], which extends this concept to worst-case considerations, and the MLS location service [18], which additionally takes into account node mobility. A more detailed discussion of the basic assumptions of geographic routing can be found in [74, Chapter 11].

3 Model

This section provides definitions of the model employed in this paper. We first give a formal definition of our ad hoc network model:

Definition 3.1 (*Quasi Unit Disk Graph*) Let V be a set of nodes in the 2-dimensional plane $\mathbb{R}^2$ and $d \in [0, 1]$ be a parameter. The symmetric Euclidean graph (V, E), such that for any pair of nodes $u,v \in V$

- $(u,v) \in E$ if $|u\ v| \leq d$ and
- $(u,v) \notin E$ if $|u\ v| > 1$,

where $|u\ v|$ is the Euclidean distance between the nodes u and v, is called a quasi unit disk graph (quasi-UDG) with parameter d.

In the subsequent section, we establish a lower bound for the message complexity of so-called volatile memory routing algorithms. With this model nodes are attributed with a short-term memory in which for each message a constant number of bits may be stored temporarily.

Definition 3.2. (*Volatile memory routing algorithm*) The task of a volatile memory routing algorithm is to transmit a message from a source s to a destination t in a graph, where each node of the graph holds a memory in which O(log n) bits may be stored as long as the message is en route, where n is the number of network nodes.

In particular this model allows the nodes to store message identifiers—having a bit length logarithmic in the number of nodes—for flooding (cf. Sect. 6).

The second important algorithm model discussed in this paper is *geographic* routing [45, 74]:

Definition 3.3. (*Geographic Routing Algorithm*) The task of a geographic routing algorithm is to transmit a message from a source s to a destination t in a graph while observing the following rules:

- Every node is informed about its own and all of its neighbors' positions.
- The source of a message knows the position of the message destination.
- A message may contain control information about at most O(1) nodes.
- A node is only allowed to temporarily store a message before retransmission; no other memory is available.

As stated above, in original geographic routing a node is allowed to store messages only temporarily before relaying them. In order to enable an algorithm to employ flooding, this restriction has to be relaxed:

Definition 3.4. (*Geographic Volatile Memory Routing Algorithm*) A geographic volatile memory routing algorithm is a volatile memory routing algorithm additionally observing the first three rules of the definition of geographic routing algorithms.

In the following, we provide a concise overview of basic concepts of distributed computing vital for the understanding of this paper. More detailed descriptions can be found in textbooks, such as in [61].

At certain points of the paper, we have to distinguish between the *synchronous* and the *asynchronous* model of distributed computation. In the *synchronous* model, communication delays are assumed to be bounded. As a consequence it can also be assumed that all processes running on different network nodes perform their message sending and receiving operations in simultaneous and globally clocked rounds. In the *asynchronous* model, message delays are unbounded. No assumptions can be made on the duration of single process operations.

Two fundamental measures in distributed computing are *message* and *time complexity*. The *message complexity* of a distributed algorithm is the total number of messages sent during its execution. The definition of *time complexity* depends on the synchrony model: In the synchronous model, time complexity is the total number of rounds elapsed between algorithm start and algorithm termination. In the asynchronous model such a simple time model cannot naturally be obtained, since the transmission delay of a message is unbounded. The common solution to this is the assumption that the message delay is at most one time unit.

Finally, since we consider message complexities in this paper, we define the *cost of a path* according to the link distance metric, that is, the cost of a path is the number of edges on the path. Similarly, we consider *spanner* graphs with respect to the link distance metric: A graph $G' = (V, E')$ is a spanner of a graph $G = (V, E)$ with stretch factor k if and only if for any pair of nodes (u,v) the cost of the shortest path in G' is at most k times the cost of the shortest path in G.

4 A lower bound in quasi unit disk graphs

Before discussing particular routing algorithms, we present in this section a lower bound on the message complexity of any volatile memory routing algorithm. This result is established constructing a family of graphs.

Theorem 4.1. *Let c be the cost of a shortest path from s to t. There exist graphs in which any (randomized) volatile memory routing algorithm has (expected) message complexity $\Omega((\frac{c}{d})^2)$.*

Proof We provide a constructive proof by describing a class of graphs for which the theorem holds.

The basic element used for the construction of these graphs is formed by k nodes (k to be determined later) equidistantly placed on a line, such that the distance between two adjacent nodes is $d + \varepsilon$ for a small $\varepsilon > 0$ (cf. vertical chains in Fig. 1). There exists an edge between every pair of nodes (u,v), such that $(\lceil\frac{1}{d}\rceil - 1)\, d < |uv| \le 1$, that is, the nodes are connected by all the edges with maximum Euclidean length not greater than 1. In addition there is a head node having an edge to each one of the first $\lceil\frac{1}{d}\rceil - 1$ nodes on the line (the head node is located such that all additional edges have length at most 1). As shown in Fig. 1, k such vertical chains are placed side by side with distance $d + \varepsilon$ such that the nodes form a matrix. The head nodes of these chains are interconnected in a way that they have the same chain structure among themselves (uppermost row in Fig. 1) with their head node (of second order) denoted by s. The node t—located near the bottom right corner of the node matrix—is connected to one of the end nodes of exactly one of the vertical chains by a simple chain of nodes. Note that the constructed graph is a quasi unit disk graph.

The main property posing a problem for a routing algorithm is that a matrix column consists of $\lceil\frac{1}{d}\rceil - 1$ interleaved chains which are only connected via the head node. (The same also holds for the first matrix row.) Consequently only one of the neighbors of s leads to h, the head node of the column connected to t, and only one of the neighbors of h leads to the bottom node connected to t. Since a volatile memory routing algorithm has no a priori information about the graph structure, a deterministic algorithm has to explore every matrix node before finding the path to t. (For a randomized algorithm, t can be connected to the matrix such that the algorithm has to explore roughly half of the matrix nodes in expectation.) A volatile memory routing algorithm therefore has to send $\Omega(n)$ messages, where n is the total number of nodes. The optimal path on the other hand—almost exclusively using edges of length nearly 1—has cost about $2k \cdot d$, which—together with $k \approx \sqrt{n}$—establishes the theorem. □

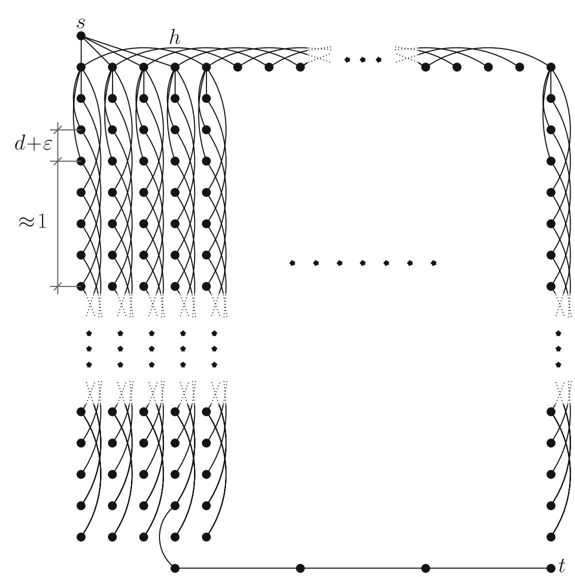

Fig. 1 Message-complexity lower bound for volatile memory routing algorithms in quasi unit disk graphs

5 Topology control

In the previous section we introduced a lower bound graph class in which any volatile memory routing algorithm cannot find the destination with message complexity less than $\Omega((\frac{c}{d})^2)$. In this section we now describe how to obtain a subgraph of a given quasi unit disk graph which forms the basis for our algorithms matching the lower bound. This *Backbone Graph* features two important properties exploited for routing: (1) It contains in a given area A at most $O(\frac{A}{d^2})$ nodes and (2) it is a $O(\log(\frac{1}{d}))$-spanner.

Given a quasi unit disk graph G, the Backbone Graph is constructed in three steps. Steps 1 and 2 can be performed by a standard distributed algorithm (as described in [2]) by having the nodes send `dominator` and `connector` messages. The details of this algorithm are omitted, as such discussion would go beyond the scope of this paper.

In particular, the Backbone Graph is constructed in the following steps:

1. The first step consists of a clustering process. We construct a Maximal Independent Set *MIS* of nodes in G. Note that since *MIS* is an independent set in G, any two nodes in *MIS* have distance greater than d and consequently a given area A contains at most $O(\frac{A}{d^2})$ nodes in *MIS*. For the purpose of routing, the nodes in *MIS* will later become cluster heads: Since the nodes in *MIS* also form a Dominating Set, any node in G will have at least one node from *MIS* within its neighborhood and will choose one of these as its cluster head.
2. In a second step the cluster heads are linked together by connector nodes, connecting all pairs of nodes in *MIS* that are at most three hops apart in G. This results in the *Dense Backbone Graph* G_{DBG}. Since *MIS* is a Dominating Set, the cluster heads can be connected by bridges consisting of at most two nodes. Furthermore, G_{DBG} is a constant-stretch spanner of G.
3. G_{DBG} can contain $\Omega(\frac{A}{d^4})$ nodes in a given area A, which exceeds the lower bound by a factor of $\frac{1}{d^2}$. The size of *MIS* alone matching the lower bound, the third step now reduces the number of connecting bridges between cluster heads. Let $G_{DBG}^{(v)}$ denote the graph with node set *MIS* and (virtual) edges between all nodes connected by bridges in G_{DBG}. Our objective is now to construct a subgraph $G_{BG}^{(v)}$ of $G_{DBG}^{(v)}$ with $O(\frac{A}{d^2})$ (virtual) edges within the area A. It eventually follows that the final Backbone Graph G_{BG}—where the (virtual) edges in $G_{BG}^{(v)}$ have again been replaced by connector nodes and their adjacent edges—contains at most $O(\frac{A}{d^2})$ nodes within the area A. In order to obtain a graph $G_{BG}^{(v)}$ with the desired property, the plane is divided by a grid into square cells of side length 6. In each cell z all nodes and edges completely contained within z temporarily form a local network. (Note that we assume for this operation that the nodes are informed about their positions.) The number of nodes contained within z is at most $O(\frac{1}{d^2})$. We now apply an algorithm constructing a sparse spanner [3, 61, 62] to reduce the number of edges contained in z to $O(\frac{1}{d^2})$.[2] This procedure is repeated three times on grids with their origin shifted by (3,0), (0,3), and (3,3), respectively, relative to the origin of the first grid (cf. Fig. 2). Note that these are local operations since the subgraphs are of bounded size. The edge set of graph $G_{BG}^{(v)}$ is finally formed by the union of all edges resulting from the edge reduction steps on all four grids.

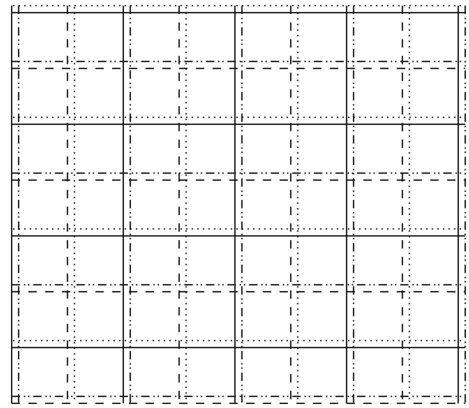

Fig. 2 Grid structure employed in the construction of a sparse spanner

The following lemma proves the first essential property of this subgraph of the given quasi unit disk graph.

Lemma 5.1 *In a given area A (with constant extension in each direction) the number of nodes and the number of edges in the Backbone Graph are both bounded by* $O(\frac{A}{d^2})$.

Proof The grids employed for the edge reduction steps are chosen to have two properties: (1) Every edge in $G_{DBG}^{(v)}$ is completely contained in at least one cell and (2) any region (with constant extension in each direction) is intersected by at most a constant number of grid cells (for instance a square of side length 3 can be intersected in total by at most nine grid cells). Property (1) guarantees that every edge is considered at least with one of the four grids: Together with the fact that the edge reduction step does not alter the number of components in a cell subgraph, it follows that the number of components in the complete graph is not altered either. Since each resulting subgraph contains at most $O(\frac{1}{d^2})$ edges and together with Property (2), it follows that also the union of all remaining edges—that is the number of edges in $G_{BG}^{(v)}$—is not

[2] The mentioned algorithm constructs for a constant $\kappa \geq 1$ an $O(\kappa)$-spanner with at most $n^{1+1/\kappa}$ edges. Setting $\kappa = \log n$ and since $n = \frac{1}{d^2}$ holds, we obtain a graph with the required properties.

greater than $O(\frac{1}{d^2})$ for a constant region. The fact that each edge in $G_{BG}^{(v)}$ corresponds to at most two nodes and three edges in G_{BG} and $G_{BG}^{(v)}$ having at most $O(\frac{A}{d^2})$ nodes for an area A (the nodes in *MIS*) finally leads to the lemma. □

The second essential property of G_{BG} is shown in the following lemma.

Lemma 5.2 *The Backbone Graph G_{BG} is a spanner of G_{DBG} with stretch factor* $O(\log(\frac{1}{d}))$.

Proof Every edge in $G_{DBG}^{(v)}$ is contained in at least one grid cell and consequently also considered in at least one of the respective subgraphs. Since the edges retained in each subgraph form a $O(\log(\frac{1}{d}))$ -spanner (on the subgraph), this property also holds for the union of all subgraphs, $G_{BG}^{(v)}$. Finally, each edge in $G_{BG}^{(v)}$ resulting in at most three edges in G_{BG}, the lemma follows. □

In distributed computing, a distinction is made between the *one-hop broadcast* model and the *point-to-point communication* model: In the one-hop broadcast model a node can simultaneously send a message to all its neighbors, whereas in the point-to-point communication model, a message is sent over an edge to one distinct neighbor. The algorithms described in the remaining sections are assumed to execute on G_{BG}. Since in this graph the number of nodes and the number of edges are asymptotically equal in a given area, the two models can be employed interchangeably—depending on whether we argue over the number of nodes or edges in the graph—with respect to both asymptotic time and asymptotic message complexity.

When routing a message m from a source s' to a destination t', the nodes s' and t' will in general not be cluster heads. The complete process of routing therefore consists of

1. s' sending m to its associated cluster head s,
2. routing m from s to t, the cluster head associated to t', and
3. t sending m to t'.

Since steps 1 and 3 incur only constant cost with respect to both message and time complexity, we exclusively consider step 2 in the remaining part of the paper. Whenever mentioning a source s or a destination t, we therefore assume that s and t are cluster heads.

6 Message-optimal flooding

In this section, we discuss the message and time complexities of the Echo algorithm in quasi unit disk graphs. For succinctness, we only give a short outline of the algorithm execution; more detailed information can be found for instance in [12, 61]. The Echo algorithm consist of a flooding phase and an echo phase.

– The flooding phase is initiated by the source s by sending a flooding message—containing a time-to-live (TTL) counter τ—to all its neighbors. Each node receiving the flooding message for the first time decrements the TTL counter by one and retransmits the message to all its neighbors (with the exception of the neighbor it received the message from). In the synchronous model, this flooding phase constructs a Breadth First Search (BFS) tree.
– From the leaves of this tree—the nodes where the τ counter reaches zero—echo messages are sent back to the source along the BFS tree constructed during the flooding phase. An inner node in the BFS tree can decide locally when to send an echo message to its parent in the tree by awaiting receipt of echo messages from all of its children.

By initiating the first flooding phase with τ set to 1 and relaunching a flooding phase with doubled τ whenever the echo messages indicate that the destination has not yet been reached, both time and message complexities can be bounded:

Theorem 6.1 *Employed on G_{BG} in the synchronous model, the Echo algorithm reaches the destination with message complexity* $O((\frac{c}{d})^2)$ *and time complexity* $O(c \cdot \log(\frac{1}{d}))$, *where c is the cost of a shortest path between s and t. This is asymptotically optimal with respect to message complexity.*

Proof The Echo algorithm floods the complete network with message complexity $O(m)$ and time complexity $O(D)$, where m is the number of edges in the network and D is the diameter of the network. Since no edge in G_{BG} is longer than 1, all nodes reached with a certain τ lie within the circle centered at s with radius τ. The number of edges within this circle is bounded by $O((\frac{\tau}{d})^2)$. Note that the destination is reached at the latest for the maximal τ less than $2 \cdot c$. Since τ is doubled after each failure, the total number of visited edges is formed by a geometric series and consequently asymptotically dominated by the number of edges in the circle with maximum τ, from which the message complexity follows. Asymptotic optimality is a consequence of the lower bound established in Sect. 4.

The time complexity follows from the fact that the BFS tree constructed during the flooding phase contains a shortest path from s to t. Since G_{BG} is a $\log(\frac{1}{d})$ -spanner of G, the shortest path in G_{BG}, on which the algorithm is executed, is $c \cdot \log(\frac{1}{d})$. The time complexity of a single flooding-echo round being proportional to τ and again the total time complexity being asymptotically dominated by the maximum τ used, the time complexity follows. □

In the asynchronous model, the synchronizer construction introduced in [4] can be employed.

Theorem 6.2 *When employed on G_{BG} in the asynchronous model, the Echo algorithm reaches the destination with message complexity* $O((\frac{c}{d})^2 \cdot \log^3(\frac{c}{d}))$ *and time complexity* $O(c \cdot \log(\frac{1}{d}) \cdot \log^3(\frac{c}{d}))$, *where c is the cost of the shortest path between s and t.*

Proof The synchronizer construction introduced in [4] incurs an additional cost factor of $O(\log^3(\frac{1}{d}))$, where n is the number of involved network nodes, with respect to both message and time complexity. As in our case the number of involved nodes is in $O((\frac{c}{d})^2)$, this cost factor is in $O(\log^3(\frac{c}{d}))$. Plugging in the above Echo algorithm for the synchronous model yields the lemma. □

For geographic routing as discussed in the following section, a variant of Echo can be defined by replacing the time-to-live counter by a geometric argument: The flooding message is retransmitted only by nodes located within a circle centered at s with a certain radius r.

Lemma 6.3 *The geographic Echo algorithm reaches t with message and time complexity* $O((\frac{c}{d})^2)$, *where c is the link cost of the shortest path. This holds for both the synchronous and the asynchronous model and is asymptotically optimal with respect to message complexity.*

Proof In contrast to the above Echo algorithm using TTL, all nodes located within the restricting circle centered at s with radius r participate in the execution of the geographic algorithm (except for nodes that are not connected to the source by any path entirely contained in this circle). This circle containing at most $O((\frac{r}{d})^2)$ nodes, the message complexity follows, where the remaining reasoning is analogous to the one in the proof of Theorem 6.1. The time complexity follows from the fact that time complexity cannot be greater than message complexity. □

7 Greedy Echo routing

Although asymptotically message-optimal, a flooding-based algorithm is prohibitively expensive in most networks for practical purposes. Previous work showed how this problem can be tackled by combining a correct routing algorithm (which is guaranteed to find the destination) with a greedy routing scheme [10, 47, 74]. In this section we follow this example by describing a geographic volatile memory routing algorithm that tries to leverage the advantages of a greedy routing approach with respect to both conceptual simplicity and message-efficiency: In order to route a message, a node simply forwards it to its neighbor closest to the destination. Greedy routing can however run into a local minimum with respect to the distance to the destination, that is a node without any neighbors closer to t. In the algorithm described below, such a local minimum is overcome by employment of restricted flooding, in particular by the aid of the geographic Echo algorithm as described in the previous section. In this section we therefore refer by Echo to the geographic Echo algorithm. We denote by Echo_r the subalgorithm of geographic Echo consisting of the flooding and the corresponding echo phase for the radius r.

Our algorithm GEcho combines both greedy routing and flooding in two modes: Generally the message is forwarded in greedy mode as long as possible. Whenever running into a local minimum, the algorithm switches to echo mode. In order to keep the cost of flooding-based echo low, the algorithm tries to fall back to greedy mode as early as possible. The fallback criterion is chosen such that the combined routing algorithm is asymptotically optimal with respect to message complexity. In particular, the Echo algorithm does not terminate only when finding t, but already when finding a node v which is significantly closer to t than the local minimum, as described in step 2 of the GEcho algorithm:

GEcho The value q is a constant parameter chosen prior to algorithm execution such that $0 < q \le 1$.

0. Start at s.
1. **(Greedy Mode)** Forward the message to the neighbor in G closest to t. If t is reached, terminate. If a local minimum is reached, continue with step 2, otherwise repeat step 1 at the next node.
2. **(Echo Mode)** Execute algorithm Echo starting at the local minimum u (with an initial flooding radius of 1) until either reaching t—in which case the algorithm terminates—or finding a node v, such that $|u\,t| - |v\,t| \ge q \cdot r$, where r is the currently chosen radius in Echo_r, the subalgorithm of Echo using radius r. Proceed to v and continue with step 1.

In the following, we obtain a statement on the asymptotic complexity of the algorithm. We first show that the number of messages sent in greedy mode is bounded:

Lemma 7.1 *The number of messages sent in greedy mode is bounded by* $O((\frac{c}{d})^2)$.

Proof Let us exclusively consider the sequence U of nodes sending messages in greedy mode or receiving messages sent in greedy mode during the execution of the algorithm. Note that the distance to t is strictly decreasing within U. Since the algorithm stays in greedy mode until reaching a local minimum, U is partitioned into subsequences $U_1, U_2, \ldots, U_k, k \ge 1$ of nodes by the occurrence of local minima: A local minimum only receives a greedy message without being able to send it to a subsequent node in greedy mode. Within a subsequence $U_i = u_1, u_2, \ldots, u_{\ell_i}$, $\ell_i \ge 2$ any two nodes $u_j, u_{j+2}, 1 \le j \le \ell_i - 2$ have distance greater than d (otherwise u_j would have sent the greedy message directly to u_{j+2}). On the other hand also the distance between a local minimum u_{ℓ_i} and the first node in the following subsequence U_{i+1} have distance greater than d (otherwise u_{ℓ_i} would not be a local minimum). Together with the fact that all nodes in U are located within the circle C centered at t with radius $|st|$, the number of nodes in the total sequence U is therefore bounded by twice the maximum number of nodes with relative distance greater than d—or likewise the maximum number of nonintersecting disks of radius $d/2$—that can be placed within C. With $|st| \leq c$, the lemma follows. □

We now confine ourselves to the number of messages sent in echo mode. Note that after each *round*, defined to be one execution of step 1 or step 2, the algorithm is strictly closer to t than before that round.

Lemma 7.2 *For a given r, the subalgorithm* Echo$_r$ *is executed at most* $\lceil \frac{|st|}{qr} - 1 \rceil$ *times.*

Proof According to the criterion described in step 2, an echo round initiated at node u terminates—unless arriving at t—only if it finds a node v such that $|ut| - |vt| \geq q \cdot r$. For any r (also if at a particular node Echo$_r$ fails and r is doubled) such progress can be made at most $\lceil \frac{|st|}{qr} - 1 \rceil$ times, since after each round the algorithm is strictly closer to t than before. □

With this property we can obtain the total number of messages sent in echo mode during algorithm execution.

Lemma 7.3 *The total number of messages sent in echo mode is at most* $\mathrm{O}((\frac{c}{d})^2)$.

Proof We obtain the total number of messages sent in echo mode by summing up over all nodes ever contained in a circle bounding Echo$_r$. Since the number of nodes contained in a given circular area is asymptotically proportional to the size of the area, it is sufficient to compute the total area covered by all Echo$_r$ bounding circles. Let $r_i = 2^i$, $i = 1, 2, 3, \ldots$ denote the radii of the echo-bounding circles. The maximum r_i can be found by the observation that (1) all echo-restricting circles have their centers at a node not farther from t than s and (2) the circle centered at any node not farther from t than s having radius 2 c completely contains the shortest path. Since the value of r in Echo$_r$ is obtained by doubling, the maximum r_i used overall is less than 4 c; the maximum i reached is consequently $\lceil \log(4c) \rceil$. With R_i being the total number of bounding circles used with radius r_i, we obtain

$$A = \sum_{i=0}^{\lceil \log(4c) \rceil} R_i \cdot \pi r_i^2$$

for the total covered area A. Using Lemma 7.2, we obtain

$$\begin{aligned} A &\leq \pi \cdot \sum_{i=0}^{\lceil \log(4c) \rceil} \left\lceil \frac{|st|}{qr_i} - 1 \right\rceil \cdot r_i^2 \\ &< \pi \cdot \sum_{i=0}^{\lceil \log(4c) \rceil} \frac{|st|}{q} \cdot r_i \underset{(|st| \leq c)}{\leq} \frac{\pi c}{q} \cdot \sum_{i=0}^{\lceil \log(4c) \rceil} 2^i \\ &= \frac{\pi c}{q} \cdot \left(2^{\lceil \log(4c) \rceil + 1} - 1\right) \in \mathrm{O}(c^2). \end{aligned}$$

The Area A containing at most $\mathrm{O}(\frac{A}{d^2})$ nodes (cf. Sect. 5), the lemma follows. □

In total, the complexity of the GEcho algorithm can be bounded as follows:

Lemma 7.4 *The algorithm GEcho finds the destination with both message and time complexity* $\mathrm{O}((\frac{c}{d})^2)$, *where c is the link cost of the shortest path.*

Proof The message complexity bound follows directly from the previous two lemmas. The time complexity bound follows from the fact that time complexity cannot be greater than message complexity. □

Theorem 7.5 *The algorithm GEcho is asymptotically optimal with respect to message complexity.*

Proof Follows from Lemma 7.4 and Sect. 4. □

8 Large *d*-values

This section treats the special case where the parameter d of the quasi unit disk graph G is $d \geq 1/\sqrt{2}$. This case was already considered by Barrière et al. [5]. It is shown there that for $d \geq 1/\sqrt{2}$ standard geographic routing is possible. Here we extend these results and present a geographic routing algorithm which is asymptotically optimal, that is, whose cost is quadratic in the cost of an optimal path (cf. [45]).

The structural difference between quasi-UDGs for $d < 1/\sqrt{2}$ and quasi-UDGs for $d \geq 1/\sqrt{2}$ lies in the local environment of intersecting edges. If $d \geq 1/\sqrt{2}$, all intersections can be detected locally. This is shown by the following two lemmas.

Lemma 8.1 *Let* $e = (u, v)$ *be an edge and w be a node which is in the disk with diameter* (u,v). *In this configuration w has an edge to at least one of the nodes u or v.*

Proof The following proof is illustrated by Fig. 3. Because $|uv| \leq 1$, the regions of the points whose distances to u and v are greater than $1/\sqrt{2}$ (shaded areas in Fig. 3) do not intersect inside $\mathcal{C}$, the disk with diameter (u, v). Thus $|uw| \leq 1/\sqrt{2}$ and/or $|vw| \leq 1/\sqrt{2}$. In the figure, this holds

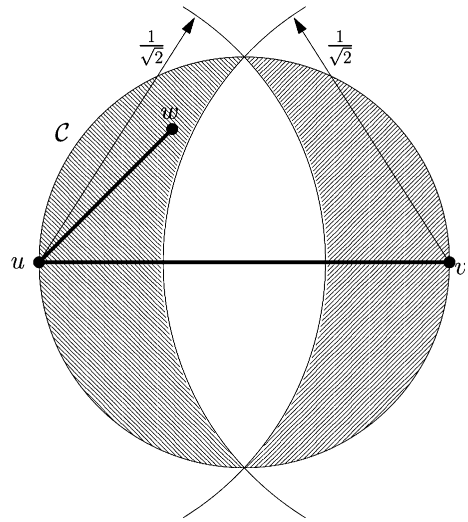

Fig. 3 For $d \leq 1/\sqrt{2}$, w is connected by an edge to u and/or to v (cf. proof of Lemma 8.1)

for u and w, implying that G contains an edge between these two nodes. □

Lemma 8.2 *Let $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ be two intersecting edges in a quasi-UDG G with parameter $d \geq 1/\sqrt{2}$. Then at least one of the edges (u_1, u_2), (u_1, v_2), (v_1, u_2), or (v_1, v_2) exists in G.*

Proof We have to show that one of the four sides of the quadrangle (u_1, u_2, v_1, v_2) is shorter than $1/\sqrt{2}$. Because the sum of the interior angles of the quadrangle is 2π, at least one of the angles has to be greater or equal to $\pi/2$. Assuming without loss of generality that this is the angle at node u_2, u_2 lies in the disk with diameter (u_1, v_1), and the lemma follows from Lemma 8.1. □

We will now give an overview of the results of [5]. The presented algorithm consists of three steps. In a first step, the quasi-UDG G is extended by adding virtual edges. Whenever there is an edge (u, v) and a node w which is inside the circle with diameter (u, v), for at least one of the nodes u and v—without loss of generality let it be u—the distance to w is smaller than or equal to $1/\sqrt{2}$ (Lemma 8.1.), and therefore u has a connection to w. If there is no edge between v and w, a virtual edge is added. Sending a message over this virtual edge is done by sending the message via node u. This process is done recursively, that is, also if (u, v) is a virtual edge. The graph obtained by adding the virtual edges to G is called the super-graph $S(G)$. Barrière et al. prove that on $S(G)$ the Gabriel Graph $GG(S(G))$ can be constructed yielding a planar subgraph of $S(G)$. In the Gabriel Graph construction, an edge (u, v) is removed if and only if there is a node w in the disk with diameter (u, v) [21]. Then any geographic routing algorithm guaranteed to reach the destination is applied on $GG(S(G))$.

In order to obtain an optimal geographic routing algorithm, we have to change the algorithm of [5] in two ways: (i) The planar graph which we need for geographic routing should be a constant-stretch spanner and the number of nodes in a given area A should not exceed O(A). (ii) We have to replace the geographic routing algorithm by a more elaborate variant such as AFR [45] or one of its successors GOAFR [47] and GOAFR$^+$ [44]. These algorithms apply area doubling strategies—similar to the GEcho algorithm—together with graph planarization and face routing techniques.

One of the bounding factors for the spanning property is given by the recursive depth of the virtual edge construction, that is the length of paths corresponding to virtual edges. From [5], we have the following result.

Lemma 8.3 *Let λ be the minimum Euclidean distance between any two nodes. If $d \geq 1/\sqrt{2}$, the length of the route in G corresponding to a virtual edge in $S(G)$ is at most $1 + \frac{1}{2\lambda^2}$.*

Proof The lemma follows directly from Property 1 in Sect. 5 of [5]. □

As shown later in the section, the assumption that there is a minimum Euclidean distance λ between any two nodes is sufficient to allow for the formulation of asymptotically optimal geographic routing algorithms. However, even without this assumption, but employing the Backbone Graph G_{BG} (cf. Sect. 5), we obtain a quasi-UDG with bounded degree, a property which we will prove to be equivalent to the minimum distance assumption.

Specifically, we start by constructing G_{BG}. This gives us a set of dominator nodes $D = MIS$ and a set of connector nodes C. We transform G_{BG} into a quasi-UDG $G'_{BG} = (V', E')$ by setting $V' = D \cup C$ and by including all possible edges of E in E' (all edges between nodes of V').[3]

Lemma 8.4 *The degree of each node in the quasi-UDG G'_{BG} is bounded by a constant.*

Proof Because the dominator nodes D have distance at least $1/\sqrt{2}$ from each other, the number of dominators which are within three hops from a node $v \in V'$ is bounded by a constant. Only these dominators can add connector nodes which are neighbors of v. Each of them can only add a constant number of connector nodes; therefore, the degree of node v has to be constant. A more detailed proof can be done analogously to the proof for the same lemma for unit disk graphs [2, 71]. □

[3] Note that G'_{BG} can contain more edges than G_{DBG} introduced in Sect. 5, as G'_{BG} contains *all* edges between nodes in V'.

G'_{BG} is now used for the Gabriel Graph construction. First, virtual edges are added as in the algorithm of [5], resulting in a super-graph $S(G'_{BG})$. Then $GG(S(G'_{BG}))$ is constructed. In analogy to Lemma 8.3, we can state a bound on the maximum route length for any virtual edge.

Lemma 8.5 *Let $G = (V, E)$ be a quasi-UDG with maximum node degree Δ. If $d \geq 1/\sqrt{2}$, the length of the route in G corresponding to a virtual edge in $S(G)$ is at most* $O(\Delta^2)$.

Proof Let $(u, v) \in E$ be an edge of G. Further let $w_1, \ldots, w_k$ be a sequence of nodes which recursively force the creation of new virtual edges e_i for which the corresponding route contains (u, v). Let $\ell_0 := |uv|$ and ℓ_i be the Euclidean length of the virtual edge e_i (see Fig. 4 as an illustration). λ_i is the length of the edge which together with e_{i-1} provides the route for e_i ($\lambda_i \leq d$). For the length ℓ_i of the ith virtual edge e_i we obtain

$$\ell_i \leq \sqrt{\ell_{i-1}^2 - \lambda_i^2} = \sqrt{1 - \frac{\lambda_i^2}{\ell_{i-1}^2}} \cdot \ell_{i-1} \leq \sqrt{1 - \lambda_i^2} \cdot \ell_{i-1}.$$

The last inequality follows from $\ell_{i-1} \leq 1$. We therefore have

$$\ell_k \leq \prod_{i=1}^{k} \sqrt{1 - \lambda_i^2} \cdot \ell_0 \leq \prod_{i=1}^{k} \sqrt{1 - \lambda_i^2}. \tag{1}$$

We define $\lambda := 1/k \sum_{i=1}^{k} \lambda_i$ to be the average length of the edges corresponding to the λ_i. As we will show in Lemma 8.6, the expression of Eq. 1 can be upper-bounded by replacing each λ_i by λ:

$$\ell_k \leq \left(\sqrt{1 - \lambda^2}\right)^k = \left(1 - \lambda^2\right)^{k/2}. \tag{2}$$

In a quasi-UDG all nodes in a disk with radius $d/2$ are direct neighbors. Therefore, when starting at a node u, after at most $\Delta + 1$ hops, a cycle-free path must leave the disk with radius $d/2$ around u. Thus, the sum of the lengths of $\Delta + 1$ successive edges on a cycle-free path is greater than $d/2$; in particular, for $d \geq 1/\sqrt{2}$ this is a constant. The average edge length of any cycle-free path is thus at least $\Omega(1/\Delta)$. As illustrated in Fig. 4, the λ_i form two paths. Therefore, the average λ_i must be on the order of $\lambda \in \Omega(1/\Delta)$. As $(1 - 1/n)^n \leq 1/e$, we can set $k = 2/\lambda^2 \in O(\Delta^2)$ in (2) and obtain

$$\ell_k \leq (1 - \lambda^2)^{1/\lambda^2} \leq 1/e \leq 1/\sqrt{2}.$$

According to the definition of a virtual edge, the length of such an edge e_k is at least $\ell_k \geq 1/\sqrt{2}$. Accordingly, k cannot be chosen greater than in $O(\Delta^2)$, which concludes the proof. □

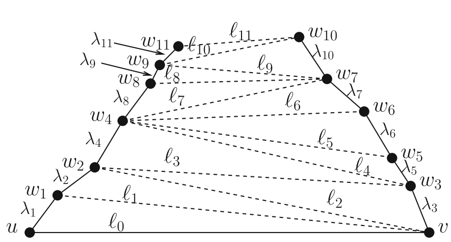

Fig. 4 Recursive depth of virtual edges in $S(G)$ (cf. proof of Lemma 8.5)

In Eq. 2 we used that ℓ_k can be upper-bounded by replacing each λ_i by the average edge length λ. This is proved in the following lemma:

Lemma 8.6 *Given k real numbers $\lambda_1, \ldots, \lambda_k$ with $|\lambda_i| \leq 1$,*

$$\prod_{i=1}^{k} \sqrt{1 - \lambda_i^2} \leq \left(\sqrt{1 - \lambda^2}\right)^k$$

holds, where $\lambda := 1/k \sum_{i=1}^{k} \lambda_i$.

Proof To prove this lemma, it is sufficient to show that the inequality holds for replacement of two values λ_i and λ_j by their average. It then follows that replacing $\lambda_{\min} := \min \lambda_i$ and $\lambda_{\max} := \max \lambda_i$ by $\lambda_{\text{avg}} := (\lambda_{\min} + \lambda_{\max})/2$ does not make the product $\prod_{i=1}^{k} \sqrt{1 - \lambda_i^2}$ smaller. Repeated application of this substitution of λ_{avg} for $\lambda_{\min}$ and $\lambda_{\max}$ to the newly obtained set of λ_i values results in a chain of inequalities in which $\lambda_{\min}$ and $\lambda_{\max}$ (in every respectively updated set of λ_i values) converge to λ (defined to be the average of all *initial* λ_i values) and at the end of which stands

$$\ldots \leq \left(\sqrt{1 - \lambda^2}\right)^k.$$

We will first show that

$$\sqrt{1 - \lambda_i^2} \sqrt{1 - \lambda_j^2} \leq 1 - \overline{\lambda}_{ij}'^2, \tag{3}$$

where $\overline{\lambda}_{ij}' := \sqrt{(\lambda_i^2 + \lambda_j^2)/2}$. In other words—setting $x_i := \lambda_i^2$ and $x_j := \lambda_j^2$ —we show that $\sqrt{1 - x_i}\sqrt{1 - x_j} \leq 1 - (x_i + x_j)/2$. Squaring this equation and subtracting the left hand side from the right hand side, we obtain

$$0 \leq \frac{x_i^2}{4} - \frac{x_i x_j}{2} + \frac{x_j^2}{4} = \frac{(x_i - x_j)^2}{4},$$

which holds for any real x_i and x_j and therefore implies the correctness of Eq. 3.

Defining $\overline{\lambda}_{ij} := (\lambda_i + \lambda_j)/2$, subtraction of $\overline{\lambda}_{ij}^2$ from $\overline{\lambda}_{ij}'^2$ leads to

$$\overline{\lambda}_{ij}'^2 - \overline{\lambda}_{ij}^2 = \frac{\lambda_i^2}{4} - \frac{\lambda_i \lambda_j}{2} + \frac{\lambda_j^2}{4} = \frac{(\lambda_i - \lambda_j)^2}{4} \geq 0,$$

the last inequality holding again for all real λ_i and λ_j, which implies $\overline{\lambda}_{ij}^2 \leq \overline{\lambda}_{ij}'^2$. Together with Eq. 3,

$$\sqrt{1-\lambda_i^2}\sqrt{1-\lambda_j^2} \leq 1-\overline{\lambda}_{ij}'^2 \leq 1-\overline{\lambda}_{ij}^2$$

holds, which proves that the product $\prod_{i=1}^{k}\sqrt{1-\lambda_i^2}$ does not become smaller after replacement of both λ_i and λ_j by $\overline{\lambda}_{ij}$. Consequently, this also holds for $\lambda_{\min}$, $\lambda_{\max}$, and λ_{avg}, which—together with the observation made at the beginning of the proof—establishes the lemma. □

Having thus completely proved the correctness of Lemma 8.5, we can now employ it to show that shortest paths are longer on $GG(S(G'_{BG}))$ than on G by at most a constant factor.

Lemma 8.7 *The Gabriel Graph $GG(S(G'_{BG}))$ is a constant-stretch spanner for the quasi-UDG G.*

Proof From Lemma 8.5, we see that the virtual edges only impose a constant factor on the cost of a path. We can therefore proceed as if all virtual edges were normal edges of G. Further, it is well known that the Gabriel Graph construction retains an energy-optimal path (if the edge cost corresponds to the square of the Euclidean edge length, see for instance [45]). As the average edge length of $S(G'_{BG})$ is a constant (cf. proof of Lemma 8.5), the number of hops and the energy cost of a path only differ by a constant factor. Therefore, the minimum energy path is only by a constant factor longer than the shortest path connecting two nodes. For further details, we refer to the analysis for unit disk graphs [44]. □

We can now state the main result of this section.

Theorem 8.8 *Let G be a quasi unit disk graph with $d \geq 1/\sqrt{2}$. Applying AFR [45], GOAFR [47], or GOAFR$^+$ [44] on $GG(S(G'_{BG}))$ yields a geographic routing algorithm whose cost is in $O(c^2)$, where c is the cost of an optimal path. This is asymptotically optimal.*

Proof As G can be the unit disk graph—setting $d := 1$—the lower bound follows from the lower bound for unit disk graphs in [45]. The number of nodes as well as the number of edges of $GG(S(G'_{BG}))$ in a given area A is proportional to A; therefore the $O(c^2)$ cost also directly follows from the respective analyses in [44, 45, 47].□

8.1 Alternative construction

We conclude the section on quasi unit disk graphs for $d \geq 1/\sqrt{2}$ with the description of an alternative construction of a planar graph which can be used to perform geographic routing. By Lemma 8.2, all edge intersections of a quasi-UDG with $d \geq 1/\sqrt{2}$ can be detected locally (in one communication round). Instead of the virtual edges/Gabriel Graph construction, we can define virtual nodes at all intersections of two edges. These virtual nodes are managed by the endpoints of the intersecting edges; sending a message from or to a virtual node means sending a message from or to a (non-virtual) neighbor of the virtual node. If this is applied on G'_{BG}, we obtain a planar graph (by definition!) with only $O(A)$ nodes in any given area A. Because this planar graph is a spanner, we obtain a geographic routing algorithm with cost $O(c^2)$ by applying AFR, GOAFR, or GOAFR$^+$ [44, 45, 47].

9 Conclusion

What is the benefit of oversimplified models about which interesting properties can be proved, that however have barely anything in common with reality? But what if we adjust our model to imitate reality to the least detail and obtain nothing but a system far too complex for stringent reasoning? These are the two extremes for which we studied a potential way out in the field of wireless ad hoc and sensor network modeling: a model capturing the essence of ad hoc networks, yet concise enough to permit rigorous theoretical results.

We consider the quasi unit disk graph—having edges between all nodes with distance at most d, d lying between 0 and 1, and no edge of length greater than 1—a good example of such a model residing between pure theory and pure practice. For this model we constructed in this paper a message-complexity lower bound for any volatile memory routing algorithm. We furthermore showed that a flooding algorithm matches this lower bound and is consequently asymptotically optimal with respect to message complexity. We described a geographic routing algorithm combining greedy routing and geographic flooding, resulting in a message-optimal algorithm in the worst case—with nodes distributed in any configuration in the plane—and message-efficient algorithm in the average case—with nodes distributed randomly in the plane. We finally showed that classic geographic routing algorithms can be employed with the same performance guarantees as for unit disk graphs if d is at least $1/\sqrt{2}$.

References

1. Abraham, I., Dolev, D., & Malkhi D. (2004). LLS: A locality aware location service for mobile ad hoc networks. In *DIALM-POMC '04: Proceedings of the 2004 Joint workshop on Foundations of Mobile Computing*, pp. 75–84.
2. Alzoubi, K., Wan, P. -J., & Frieder, O. (2002). Message-optimal connected dominating sets in mobile ad hoc networks. In

Proceedings of the 3rd ACM International Symposium on Mobile ad hoc networking & computing (MOBIHOC).
3. Awerbuch, B. (1985). Complexity of Network Synchronization. *Journal of the ACM (JACM), 32*(4), 804–823.
4. Awerbuch, B., & Peleg, D. (1990). Network synchronization with polylogarithmic overhead. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 514–522.
5. Barrière, L., Fraigniaud, P., & Narayanan, L. (2001). Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *Proceedings of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)* (pp. 19–27). ACM Press.
6. Basagni, S., Chlamtac, I., Syrotiuk, V., & Woodward, B. (1998). A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking (MOBICOM)* (pp. 76–84). Dallas, TX, USA: ACM Press.
7. Boone, P., Chavez, E., Gleitzky, L., Kranakis, E., Opatrny, J., Salazar, G., & Urrutia, J. (2004). Morelia test: Improving the efficiency of the gabriel test and face routing in ad-hoc networks. In *Proceedings of the 11th Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Smolenice Castle, Slovakia.
8. Bose, P., Brodnik, A., Carlsson, S., Demaine, E., Fleischer, R., López-Ortiz, A., Morin, P., & Munro, J. (2000). Online routing in convex subdivisions. In *International Symposium on Algorithms and Computation (ISAAC)*, vol. 1969 of *Lecture Notes in Computer Science* (pp. 47–59). Springer.
9. Bose, P., & Morin, P. (1999). Online routing in triangulations. In *Proceedings of 10th International Symposium on Algorithms and Computation (ISAAC)*, Vol. 1741 of Springer LNCS, pp. 113–122.
10. Bose, P., Morin, P., Stojmenovic, I., & Urrutia, J. (1999). Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and communications (DIAL-M)*, pp. 48–55.
11. Burkhart, M., von Rickenbach, P., Wattenhofer, R., & Zollinger, A. (2004). Does topology control reduce interference? In *Proceedings of the 5th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, pp. 9–19.
12. Chang, E. (1982). Echo algorithms: Depth parallel operations on general graphs. *IEEE Transactions on Software Engineering*, *8*, 391–401
13. Cruz, R., & Santhanam, A. (2003). Optimal routing, link scheduling, and power control in multi-hop wireless networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM)*.
14. Datta, S., Stojmenovic, I., & Wu, J. (2002). Internal node and shortcut based routing with guaranteed delivery in wireless networks. In *Cluster computing 5* (pp. 169–178). Kluwer Academic Publishers.
15. ElBatt, T., & Ephremides, A. (2002). Joint scheduling and power control for wireless ad-hoc networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*.
16. Ephremides, A., & Truong, T. (1990). Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications, 38*, 456–460.
17. Finn, G. (1987). Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, USC/ISI, March 1987.
18. Flury, R., & Wattenhofer, R. (2006). MLS: An efficient location service for mobile ad hoc networks. In *7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Florence, Italy, May.
19. Frey, H. (2005). Geographical cluster based multihop ad hoc network routing with guaranteed delivery. In *2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Washington, DC, USA, November*.
20. Fussen, M., Wattenhofer, R., & Zollinger, A. (2005). Interference arises at the receiver. In *International Conference on Wireless Networks, Communications, and Mobile Computing (WIRELESSCOM), Maui, Hawaii, USA, June*.
21. Gabriel, K., & Sokal, R. (1969). A new statistical approach to geographic variation analysis. *Systematic Zoology, 8*, 259–278.
22. Gao, J., Guibas, L., Hershberger, J., Zhang, L., & Zhu, A. (2001). Discrete mobile centers. In *Proceedings 17th Annual Symposium on Computational Geometry (SCG)* (pp. 188–196). Medford, MA, USA: ACM Press.
23. Gao, J., Guibas, L., Hershberger, J., Zhang, L., & Zhu, A. (2001). Geometric spanner for routing in mobile networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing* (pp 45–55). Long Beach, CA, USA: ACM Press.
24. Grönkvist, J. (2005). *Interference-based scheduling in spatial Reuse TDMA*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden.
25. Hou, T., & Li, V. (1986). Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications, 34*(1), 38–44.
26. Huang, H., Richa, A. W., & Segal, M. (2002). Approximation algorithms for the mobile piercing set problem with applications to clustering in ad-hoc networks. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), Atlanta, Georgia, USA*, pp. 52–61.
27. Jain, K., Padhye, J., Padmanabhan, V., & Qiu, L. (2003). Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MOBICOM)*.
28. Jia L., Rajaraman R., & Suel R. (2001). An efficient distributed algorithm for constructing small dominating sets. In *Proceedings of the* 20*th ACM Symposium on Principles of Distributed Computing (PODC), 2001*, pp 33–42.
29. Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In Imielinski & Korth (Eds.), *Mobile computing*, Vol. 353. Kluwer Academic Publishers.
30. Karp, B., & Kung, H. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 243–254.
31. Kim, Y. -J., Govindan, R., Karp, B., & Shenker, S. (2005). Geographic routing made practical. In *Proceedings of the Second USENIX/ACM Symposium on Networked System Design and Implementation (NSDI 2005)*, Boston, Massachusetts, USA, May.
32. Kim, Y. -J., Govindan, R., Karp, B., & Shenker, S. (2005). On the pitfalls of geographic face routing. In *Proceedings of the ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Cologne, Germany, September.
33. Ko, Y. -B., & Vaidya, N. (1998). Location-aided routing (LAR) in mobile a hoc networks. In *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 66–75.
34. Kothapalli, K., Scheideler, C., Onus, M., & Richa, A. (2005). Constant density spanners for wireless ad-hoc networks. In *Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.
35. Kranakis, E., Singh, H., & Urrutia, J. (1999). Compass routing on geometric networks. In *Proceedings 11th Canadian Conference on Computational Geometry, Vancouver, August*, pp. 51–54.

36. Kuhn, F., Moscibroda, T., Nieberg, T., & Wattenhofer, R. (2005). Fast deterministic distributed maximal independent set computation on growth-bounded graphs. In *9th International Symposium on Distributed Computing (DISC)*, Cracow, Poland, September.
37. Kuhn, F., Moscibroda, T., Nieberg, T., & Wattenhofer, R. (2005). Local approximation schemes for ad hoc and sensor networks. In *Proceedings of the 3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Cologne, Germany, September.
38. Kuhn, F., Moscibroda, T., Nieberg, T., & Wattenhofer, R. (2005). Local approximation schemes for ad hoc and sensor networks. In *3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Cologne, Germany, September.
39. Kuhn, F., Moscibroda, T., & Wattenhofer, R. (2004). Initializing newly deployed ad hoc and sensor networks. In *10th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, Philadelphia, Pennsylvania, USA, September.
40. Kuhn, F., Moscibroda, T., & Wattenhofer, R. (2004). Radio network clustering from scratch. In *Proceedings of the 12th Annual European Symposium on Algorithms (ESA)*, Bergen, Norway, September.
41. Kuhn, F., Moscibroda, T., & Wattenhofer, R. (2004). Unit disk graph approximation. In *2nd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Philadelphia, Pennsylvania, USA, October.
42. Kuhn, F., Moscibroda, T., & Wattenhofer, R. (2005). On the locality of bounded growth. In *24th ACM Symposium on the Principles of Distributed Computing (PODC)*, Las Vegas, Nevada, USA, July.
43. Kuhn, F., & Wattenhofer, R. (2003). Constant-time distributed dominating set approximation. In *Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, Boston, Massachusetts, USA.
44. Kuhn, F., Wattenhofer, R., Zhang, Y., & Zollinger, A. (2003). Geometric routing: Of theory and practice. In *Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, Boston, Massachusetts, USA.
45. Kuhn, F., Wattenhofer, R., & Zollinger, A. (2002). Asymptotically optimal geometric mobile ad-hoc routing. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)* (pp 24–33). Atlanta, GA, USA: ACM Press.
46. Kuhn, F., Wattenhofer, R., & Zollinger, A. (2003). Ad-Hoc networks beyond unit disk graphs. In *1st ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, San Diego, California, USA, September.
47. Kuhn, F., Wattenhofer, R., & Zollinger, A. (2003). Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of the 4th ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC)*, Annapolis, Maryland, USA.
48. Leong, B., Liskov, B., & Morris, R. (2006). Geographic routing without planarization. In *3rd Symposium on Networked Systems Design & Implementation (NSDI)*, San Jose, California, USA, May.
49. Leong, B., Mitra, S., & Liskov, B. (2005). Path vector face routing: Geographic routing with local face information. In *13th IEEE International Conference on Network Protocols (ICNP)*, Boston, Massachusetts, USA, November.
50. Li, J., Jannotti, J., De Couto, D., Karger, D., & Morris, R. (2000). A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 120–130, Aug.
51. Li X.-Y., Calinescu G., & Wan P.-J. (2002). Distributed construction of planar spanner and routing for ad hoc wireless networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York, NY, USA.
52. Li, X. -Y., Song, W. -Z., & Wang, Y. (2004). Localized topology control for heterogenous wireless ad hoc networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Fort Lauderdale, Florida, USA, October.
53. Moscibroda, T., Oswald, Y. A., & Wattenhofer, R. (2007). How optimal are wireless scheduling protocols? In *Proceedings of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Anchorage, Alaska, USA, May.
54. Moscibroda T., von Rickenbach P., & Wattenhofer R. (2006). Analyzing the energy-latency trade-off during the deployment of sensor networks. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, April.
55. Moscibroda, T., & Wattenhofer, R. (2005). Coloring unstructured radio networks. In *Proceedings of the 17th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, Las Vegas, Nevada, USA, July.
56. Moscibroda, T., & Wattenhofer, R. (2005). Maximal independent sets in radio networks. In *Proceedings of the 24th ACM Symposium on the Principles of Distributed Computing (PODC)*, Las Vegas, Nevada, USA, July.
57. Moscibroda, T., & Wattenhofer, R. (2005). Minimizing interference in ad hoc and sensor networks. In *Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pp. 24–33.
58. Moscibroda, T., & Wattenhofer, R. (2006). The complexity of connectivity in wireless networks. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, April.
59. Moscibroda, T., Wattenhofer, R., & Zollinger A. (2006). Topology control meets SINR: The scheduling complexity of arbitrary topologies. In *7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Florence, Italy, May.
60. O'Dell, R., & Wattenhofer, R. (2005). Theoretical aspects of connectivity-based multi-hop positioning. *Theoretical Computer Science, 344*(1), 47–68, November.
61. Peleg, D. (2000). *Distributed computing, a locality-sensitive approach*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.
62. Peleg, D., & Schäffer, A. A. (1989). Graph spanners. *Journal of Graph Theory, 13*(1), 99–116.
63. Perkins, C., & Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100.
64. Rajaraman, R. (2002). Topology control and routing in ad hoc networks: A survey. *SIGACT News 33*:60–73, June
65. Santi, P. (2005). Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys (CSUR), 37*(2), 164–194, June.
66. Schmid, S., & Wattenhofer, R. (2006). Algorithmic models for sensor networks. In *Proceedings of the 14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, Island of Rhodes, Greece, April.
67. Takagi, H., & Kleinrock, L. (1984). Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications, 32*(3), 246–257.
68. Toussaint, G. (1980). The relative neighborhood graph of a finite planar set. *Pattern Recognition, 12*(4), 261–268.
69. Urrutia, J. (2005). Routing with guaranteed delivery in geometric and wireless networks. In: Stojmenovic, I. (Ed.), *Handbook of*

Wireless Networks and Mobile Computing, chapter 18 (pp. 393–406). John Wiley & Sons.
70. von Rickenbach, P., Schmid, S., Wattenhofer, R., & Zollinger, A. (2005). A robust interference model for wireless ad-hoc networks. In *Proceedings of the 5th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Denver, Colorado, USA, April.
71. Wang, Y., & Li, X. -Y. (2002). Geometric spanners for wireless ad hoc networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*.
72. Wang, Y., & Li, X. -Y. (2003). Localized construction of bounded degree planar spanner. In *Proceedings of the 1st DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, San Diego, California, USA.
73. Wattenhofer, R., Li, L., Bahl, P., & Wang, Y.-M. (2001). Distributed topology control for power efficient operation in multi-hop wireless ad hoc networks. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 1388–1397
74. Zollinger, A. (2005). *Networking unleashed: Geographic routing and topology control in ad hoc and sensor networks*. PhD thesis, ETH Zurich, Switzerland, Diss. ETH 16025.

Author Biographies

Fabian Kuhn received his M.Sc. degree in computer science (Dipl. Informatik-Ing ETH) from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland in 2001. In January 2002 he joined the Distributed Computing Group of Professor Roger Wattenhofer at ETH Zurich as a Ph.D. student and research assistant. In 2005 he earned his Ph.D. degree for his work on locality phenomena in distributed algorithms. He is currently with Microsoft Research, Mountain View, CA, USA.

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. In 2002 he returned to ETH, first as an assistant professor in the Computer Science Department, since July 2004 as an associate professor at the Information Technology and Electrical Engineering Department. Roger Wattenhofer's research interests include a variety of algorithmic and systems aspects in networking and distributed computing; in particular, peer-to-peer computing, ad hoc and sensor networking.

Aaron Zollinger received his M.Sc. degree in Computer Science (Dipl. Informatik-Ing ETH) from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland in 2001. In the same year he joined the Distributed Computing Group of Professor Roger Wattenhofer at ETH Zurich as a Ph.D. student and research assistant. In 2005 he earned his Ph.D. degree for his work on geographic routing and topology control in wireless ad hoc and sensor networks. He is currently with the Department of Electrical Engineering and Computer Science at the University of California, Berkeley, CA, USA.