An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks

Fabian Kuhn, *Member, IEEE*, Roger Wattenhofer, and Aaron Zollinger, *Member, IEEE*

***Abstract*—The one type of routing in ad hoc and sensor networks that currently appears to be most amenable to algorithmic analysis is geographic routing. This paper contains an introduction to the problem field of geographic routing, presents a specific routing algorithm based on a synthesis of the greedy forwarding and face routing approaches, and provides an algorithmic analysis of the presented algorithm from both a worst-case and an average-case perspective.**

***Index Terms*—Algorithmic analysis, routing, stretch, wireless networks.**

I. Introduction

AD HOC and sensor networks consist of autonomous devices communicating via radio equipment. Common scenarios for ad hoc networks include survivable, efficient, dynamic communication networks for emergency and rescue operations, disaster relief efforts, and similar tasks where typically no communication infrastructure is present prior to the deployment of the ad hoc network. In sensor networks, nodes are additionally equipped with sensors, performing the task of sensing a certain physical value, such as temperature, humidity, brightness, or motion, and periodically reporting the sensed data to a designated sink node for monitoring purposes.

Since ad hoc and sensor network nodes are generally assumed to be autonomous and operate for a considerable period of time, in the case of sensor networks up to several years, energy conservation is one of the central issues in this research context. On the other hand, many scenarios assume a high degree of dynamics, particularly based on node mobility.

Routing in a communication network is the process of forwarding a message from a source host to a destination host via intermediate nodes. In wired networks, routing is commonly a task performed by routers, special fail-safe network hosts particularly designed for the purpose of forwarding messages with high performance. In ideal wireless ad hoc networks, in contrast, every network node may act as a router, as a relay node forwarding a message on its way from its source node to its destination node. This process is particularly important in ad hoc networks, as network nodes are assumed to have restricted power resources and therefore try to transmit messages at low transmission power, leading to the effect that the destination of a message can typically not be reached directly from the source. The importance of this task also becomes manifest in the popular term *multihop routing*, expressing the essential role of network nodes as relay stations.

In wired networks, routing almost always takes place in relatively stable conditions; at least the main neighborhood topology remains identical over weeks, months, or even years. The primary focus of routing in wired networks is on high-performance forwarding of messages; reaction latency in the face of network topology changes, caused by failing hosts or connections, is generally of secondary importance. Considering the stability of wired networks, prompt reaction to topology changes or rapid propagation of according information is often not required, as such events are relatively rare.

Wireless ad hoc networks are of a fundamentally different character: To begin with, wireless connections are by nature significantly less stable than wired connections. Effects influencing the propagation of radio signals, such as shielding, reflection, scattering, and interference, inevitably require routing systems in ad hoc networks to be able to cope with comparatively low link communication reliability. More importantly, many scenarios for ad hoc networks assume that nodes are potentially mobile. These two factors, above all in high node mobility, cause ad hoc networks to be inherently more dynamic than wired networks. Traditional routing protocols designed for wired networks therefore generally fail to satisfy the requirements of wireless ad hoc networks.

A considerable number of routing protocols specifically devised for operation in ad hoc networks have consequently been invented. These protocols are usually classified into two groups: *proactive* and *reactive* routing protocols. *Proactive* routing protocols resemble protocols for wired networks in that they collect routing information ahead of time. A request for a message to be routed can be serviced without any further preparative actions. As every node keeps a table specifying how to forward a message, information on topology changes is propagated whenever they occur. Similar to routing protocols in wired networks, proactive routing protocols are efficient only if links are stable and node mobility is low compared to the rate of communication traffic. Already if node mobility reaches a reasonable degree, the routing overhead incurred by table update messages can become unacceptably high [1]. Another question is whether lightweight ad hoc network nodes with scarce resources can be expected to maintain routing tables potentially for all possible destinations

Manuscript received May 12, 2006; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Das.

F. Kuhn is with Microsoft Research, Mountain View, CA 94043 USA, and also with the Institute of Theoretical Computer Science, ETH Zurich, 68092 Zurich, Switzerland (e-mail: kuhn@microsoft.com).

R. Wattenhofer is with the Distributed Computing Group, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 358092 Zurich, Switzerland (e-mail: wattenhofer@tik.ee.ethz.ch).

A. Zollinger is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: zolling@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TNET.2007.900372

in the network.[1] *Reactive* routing protocols, on the other hand, try to delay any preparatory actions as long as possible. Routing occurs on demand, only. In principle, a node wishing to send a message has to flood the network in order to find the destination. Although there are many tricks to restrict flooding or to cache information overheard by nodes, flooding can consume a considerable portion of the network bandwidth. Attempting to combine the advantages of both concepts, proposals have also been made to incorporate both approaches in hybrid protocols, adapting to current network conditions.

Most of these routing protocols have been described and studied from a system-centric point of view. Simulation appears to be the preferred method of assessment. It appears, however, that a global evaluation of protocols is difficult. Ad Hoc networks have many parameters, such as transmission power, signal attenuation, interference, physical obstacles, node density and distribution, degree and type of node mobility, just to mention a few; therefore simulation cannot cover all the degrees of freedom. For a given set of parameters, certain protocols appear superior to others; for other parameters, the ranking may be reversed. One possible answer to this problem may be found in trying to rigorously analyze the efficiency of proposed protocols and algorithms. However, analyzing the complexity of ad hoc routing algorithms appears to be not only intricate, but virtually impossible. Accordingly, only few attempts have been made to analyze ad hoc routing in a general setting from an algorithmic perspective.

One specific type of ad hoc routing, in contrast, appears to be more easily accessible to algorithmic analysis: geographic routing. Geographic routing, sometimes also called directional, geometric, location-based, or position-based routing, is based on two principal assumptions. First, it is assumed that every node knows its own and its network neighbors' positions. Second, the source of a message is assumed to be informed about the position of the destination. The former assumption is currently becoming more and more realistic with the advent of inexpensive and miniaturized positioning systems. It is also conceivable that position information could be attained by local computation and message exchange with stationary devices. In order to come up to the latter assumption, that is to provide the source of a message with the destination position, several so-called location services have been proposed [2]–[5]. For some scenarios it can also be sufficient to reach *any* destination currently located in a given area, sometimes called "geocasting". These are only briefly summarized explanations why the two basic assumptions of geographic routing are reasonable. This issue is discussed in more depth for instance in [6, Ch. 11].

Geographic routing is particularly interesting, as it operates without any routing tables whatsoever. Furthermore, once the position of the destination is known, all operations are strictly local, that is, every node is required to keep track only of its direct neighbors. These two factors—absence of necessity to keep routing tables up to date and independence of remotely occurring topology changes—are among the foremost reasons why geographic routing is exceptionally suitable for operation in ad hoc networks. Furthermore, in a sense, geographic routing can be considered a lean version of source routing appropriate for dynamic networks: While in source routing the complete hop-by-hop route to be followed by the message is specified by the source, in geographic routing the source simply addresses the message with the position of the destination. As the destination can generally be expected to move slowly compared to the frequency of topology changes between the source and the destination, it makes sense to keep track of the position of the destination instead of maintaining network topology information up to date; if the destination does not move too fast, the message is delivered regardless of possible topology changes among intermediate nodes. Finally, from a less technical perspective, it can be hoped that by studying geographic routing it is possible to gain insights into routing in ad hoc networks in general, without availability of position information.

We will start our analysis of geographic routing by describing a simple greedy routing approach in Section IV. The main drawback of this approach is that it cannot guarantee to always reach the destination. Geographic routing algorithms that, in contrast, always reach the destination, are based on faces, contiguous regions separated by the edges of planar network subgraphs. It may, however, happen that these algorithms take $\Omega(n)$ steps before arriving at the destination, where n is the number of network nodes. In other words, they basically do not perform better than an algorithm visiting every node in the network. In Section V we will describe the concept of face routing and describe algorithms that not only always find the destination, but are also guaranteed to do so with cost at most $O(c^2)$, where c is the cost of a shortest path connecting the source and the destination. The next section will show that, given an instance of a class of lower bound graphs, no geographic routing algorithm will be able to perform better; in this sense, the presented face routing algorithms are asymptotically optimal in worst-case networks. Despite their asymptotic optimality, these algorithms are relatively inflexible in that they follow the boundaries of faces also in dense average-case networks where greedy routing would reach the destination much faster. Section VII will outline how greedy routing and face routing can be combined, resulting in the GOAFR$^+$ algorithm, which preserves the worst-case guarantees of its face routing components. In addition, simulations, as mentioned in the same section, showed that the GOAFR$^+$ algorithm is—to the best of our knowledge—the currently most efficient geographic routing algorithm also in average-case networks. GOAFR$^+$ particularly outperforms other routing algorithms in a critical node density range, where the network is just about to become connected and which forms a challenge to any routing algorithm, also non-geographic routing algorithms.

II. Related Work

As mentioned earlier, routing protocols for ad hoc networks can be classified as proactive and reactive protocols. Proactive protocols, such as DSDV [7], TBRPF [8], and OLSR [9], distribute routing information ahead of time in order to be able to react immediately whenever a message needs to be forwarded. On the other hand, reactive protocols, such as AODV [10], DSR [11], or TORA [12] do not try to anticipate communication and

[1]Many routing protocols in wired networks employ hierarchic addressing schemes to reduce the size of routing tables. Such an approach is generally not possible in ad hoc networks, since the addressing of nodes in such networks is commonly assumed not to necessarily follow specific coordination.

initiate route discovery as late as possible, as a reaction to a message requested to be routed. As the performance and incurred routing overhead of such protocols highly depend on the type and extent of network mobility, also hybrid protocols, such as [13]–[15], have been proposed. Further reviews of routing algorithms in mobile ad hoc networks in general can be found in [1] and [16]. Most of these protocols have been described and studied from a system perspective; performance and efficiency assessment was commonly carried out by means of simulation. To date, only few attempts have been made to analyze routing in ad hoc networks in a general setting from an analytical algorithmic perspective [17]–[19].

The early proposals of geographic routing, suggested over a decade ago, were of purely greedy nature: At each intermediate network node the message to be routed is forwarded to the neighbor closest to the destination [20]–[22]. This can, however, fail if the message reaches a local minimum with respect to the distance to the destination, that is a node without any "better" neighbors. Also a "least deviation angle" approach (*Compass Routing* in [23]) cannot guarantee message delivery in all cases.

The first geographic routing algorithm that does guarantee delivery was *Face Routing* introduced in [23] (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, it 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 [24], [25], however, without better worst-case performance than original Face Routing. Yet other geographic routing algorithms have been shown to reach the destination on special planar graphs without any runtime guarantees [26]. Ref. [27] proposed an algorithm competitive with the shortest path between source and destination on Delaunay triangulations; this is, however, not applicable to ad hoc networks, as Delaunay triangulations may contain arbitrarily long edges, whereas transmission ranges in ad hoc networks are limited. Accordingly, [28] proposed local approximation of the Delaunay Graph, however, without improving performance bounds for routing. A more detailed overview of geographic routing can be found in [29].

In [30], *Adaptive Face Routing* (AFR) was proposed. The execution cost of this algorithm—basically enhancing Face Routing by the employment of an ellipse restricting the searchable area—is bounded by the cost of the optimal route. In particular, the cost of AFR is not greater than the squared cost of the optimal route. It was also shown that this is the worst-case optimal result any geographic routing algorithm can achieve.

Face Routing and also AFR are 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 [24], [25], [31], however, without competitive worst-case guarantees. In [32], the GOAFR algorithm was introduced; to the best of our knowledge, this was the first algorithm to combine greedy and face routing in a worst-case optimal way; the GOAFR$^+$ algorithm [33] remains asymptotically worst-case optimal while improving GOAFR's average-case efficiency by employing a counter technique for falling back as soon as possible from face to greedy routing.

Lately, first experiences with geographic and in particular face routing in practical networks have been made [34], [35] 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 [36], [37]. Also more theoretical observations were made with respect to the message delivery guarantee of different algorithms employing face routing on various types of graphs [38].

The results in this paper partly rely on the $\Omega(1)$-model, the assumption that the distance between any pair of network nodes is at least a (possibly small) constant. In [33] it was shown that equivalently a clustering technique can be employed for graphs that do not comply with the $\Omega(1)$-model assumption. Clustering for the purpose of ad hoc routing has been proposed by various researchers [39], [40]. A closely related approach is the construction of *dominating sets* (see [41] and related references therein) for instance for employment as routing backbones.

In the context of routing in ad hoc and sensor networks the assumption is commonly made that—if the network nodes know their own positions—they are also informed about their neighbors' positions practically for free, that is by local exchange of according messages. [42]–[45] studied what can be done without such regularly exchanged beacon messages.

The question what is possible if no position information at all is available to the network nodes was addressed in [46] and [47] by computation of virtual node coordinates and in [48] and [49] with the focus on geographic routing.

III. Models and Preliminaries

At the beginning of every theoretical analysis stands the question of how to model the considered system. An obvious abstraction of a communication network is a graph with nodes representing networking devices and edges standing for network connections. The study of ad hoc networks in this paper assumes that network nodes are placed in the Euclidean plane. We furthermore model ad hoc networks as unit disk graphs [50]. A unit disk graph (UDG) is defined as follows:

Definition 3.1: (Unit Disk Graph): Let $V \subset \mathbb{R}^2$ be a set of points in the two-dimensional plane. The graph with edges between all nodes with distance at most 1 is called the unit disk graph of V.

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. Clearly, the unit disk graph model forms a highly idealistic abstraction of ad hoc networks. Nevertheless it admits certain insights based on algorithmic analysis. Discussions of routing in a model that more closely captures the connectivity characteristics of wireless networks can be found in [51] and [52].

To measure the quality of a routing algorithm, we attribute to each edge e a cost which is a function of the Euclidean length of e.

Definition 3.2: (Cost Function): A cost function $c :]0, 1] \mapsto \mathbb{R}^+$ is a nondecreasing function which maps any possible edge length d $(0 < d \leq 1)$ to a positive real value $c(d)$ such that $d' > d \Longrightarrow c(d') \geq c(d)$. For the cost of an edge $e \in E$ we also use the shorter form $c(e) := c(d(e))$.

Note that]0,1] really is the domain of a cost function $c(\cdot)$, that is, $c(\cdot)$ has to be defined for all values in this interval and in particular, $c(1) < \infty$. The cost model thus defined includes all popular cost measures such as the link (or hop) distance metric $(c_\ell(d) :\equiv 1)$, the Euclidean distance metric $(c_d(d) := d)$, energy $(c_E(d) := d^2$, or more generally d^α for $\alpha \geq 2)$, as well as hybrid measures which are positive linear combinations of the above metrics.

For convenience we also define the cost of a path, a sequence of contiguous edges, and of algorithms. The cost $c(p)$ of a path p is defined as the sum of the cost values of its edges. Analogously, the cost $c(\mathcal{A})$ of an algorithm $\mathcal{A}$ is defined as the summed-up cost of all edges which are traversed during the execution of an algorithm on a particular graph. The question whether a node can send a message to several neighbors simultaneously does not affect our results, as the considered algorithms do not send messages in parallel to more than one recipient.

For the sake of simplicity, we assume that the distance between any two nodes may not be arbitrarily small.

Definition 3.3: ($\Omega(1)$-Model): If the distance between any two nodes is bounded from below by a term of order $\Omega(1)$, i.e., there is a positive constant d_0 such that d_0 is a lower bound on the distance between any two nodes, this is referred to as the $\Omega(1)$-model.

Graphs with this restriction have also been called *civilized* [53] or λ-*precision* [54] graphs in the literature. As a consequence of the $\Omega(1)$-model, the above-mentioned three metrics are equivalent up to a constant factor with respect to the cost of a path. As shown in the following lemma, this holds for all metrics defined according to Definition 3.2.

Lemma 3.1: Let $c_1(\cdot)$ and $c_2(\cdot)$ be cost functions according to Definition 3.2 and let G be a unit disk graph in the $\Omega(1)$-model. Further let p be a path in G. We then have

$$c_1(p) \leq \alpha \cdot c_2(p)$$

for a constant α.

Proof: Assume without loss of generality that p consists of k edges, that is, $c_\ell(p) = k$. As $d_0 \leq c_d(e) \leq 1$ for all edges $e \in E$ and the cost functions being nondecreasing, we have $c_1(p) \leq c_1(1) \cdot k$ and $c_2(d_0) \cdot k \leq c_2(p)$. Since—according to Definition 3.2—both $c_1(1)$ and $c_2(d_0)$ are constants greater than 0, the lemma holds with $\alpha = c_1(1)/c_2(d_0)$. ∎

Also the distance in a graph of a pair of nodes u and v—defined to be the cost of the shortest path connecting u and v—differs only by a constant factor for the different cost metrics:

Lemma 3.2: Let G be a unit disk graph with node set V in the $\Omega(1)$-model. Further let $s \in V$ and $t \in V$ be two nodes and let p_1^* and p_2^* be optimal paths from s to t on G with respect to the metrics induced by the cost functions $c_1(\cdot)$ and $c_2(\cdot)$, respectively. It then holds that

$$c_1(p_2^*) \leq \alpha \cdot c_1(p_1^*) \text{ and } c_1(p_2^*) \geq \beta \cdot c_1(p_1^*)$$

for two constants α and β, that is, the cost values of optimal paths for different metrics only differ by a constant factor.

Proof: By the optimality of p_2^* we have

$$c_2(p_2^*) \leq c_2(p_1^*). \tag{1}$$

Applying Lemma 3.1 we obtain

$$c_1(p_2^*) \leq \gamma \cdot c_2(p_2^*) \text{ and } c_2(p_1^*) \leq \delta \cdot c_1(p_1^*) \tag{2}$$

for two constants γ and δ. Combining (1) and (2) yields $c_1(p_2^*) \leq \alpha \cdot c_1(p_1^*)$ for $\alpha = \gamma \cdot \delta$. Furthermore, by the optimality of p_1^*, we have $c_1(p_2^*) \geq c_1(p_1^*)$, and therefore the second equation of the lemma holds with $\beta = 1$. ∎

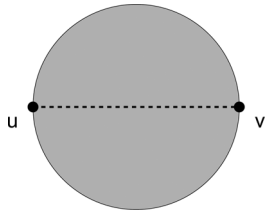

Fig. 1. An edge (u, v) in the Gabriel Graph exists if and only if the shaded disk (including its boundary) does not contain any third node.

As this equivalence of cost metrics applies not only to the link, the Euclidean, and the energy metrics, but to all cost functions according to Definition 3.2, we sometimes refer to the "cost" of an edge and mean any cost metric belonging to the above class of cost functions. In [33] it was shown that employing clustering techniques a similar result can be achieved without the $\Omega(1)$-model assumption. [33] also describes the existence of two classes of cost functions and discusses their implications on routing. In this paper we will, however, adhere to the $\Omega(1)$-model for simplicity.

For our routing algorithms the network graph is required to be *planar*, that is without intersecting edges.[2] A planar graph features *faces*, contiguous regions separated by the edges of the graph. In order to achieve planarity on the unit disk graph G, the *Gabriel Graph* is employed. A Gabriel Graph contains an edge between two nodes u and v if and only if the disk (including its boundary) having $\overline{uv}$ as a diameter does not contain a "witness" node w (cf. Fig. 1). Besides being planar, GG_G, the Gabriel Graph on the unit disk graph G, features two important properties:

- It can be computed locally: A network node can determine all its incident nodes in GG_G by mere inspection of its neighbors' locations (since G is a unit disk graph).
- The Gabriel Graph is a constant-stretch spanner for the energy metric: The construction of the Gabriel Graph on

[2]More precisely, the considered planar graphs are planar *embeddings* in the Euclidean plane.

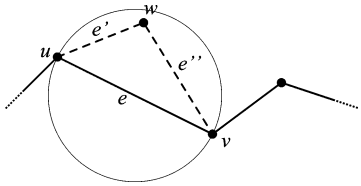

Fig. 2. The Gabriel Graph contains an energy-optimal path.

G preserves an energy-minimal path between any pair of network nodes. Together with the $\Omega(1)$-model it follows that the distance in GG_G between any pair of nodes is equal (up to constant factors) to their distance in G for all considered metrics. This is shown in the following lemma.

Lemma 3.3: In the $\Omega(1)$-model the shortest path for any of the metrics according to Definition 3.2 on the Gabriel Graph intersected with the unit disk graph is only by a constant factor longer than the shortest path on the unit disk graph for the respective metric.

Proof: We first show that at least one best path with respect to the energy metric on the UDG is also contained in $\mathrm{GG} \cap \mathrm{UDG}$. Suppose that $e = (u, v)$ is an edge of an energy-optimal path p on the UDG. For the sake of contradiction suppose that e is not contained in $\mathrm{GG} \cap \mathrm{UDG}$. Then there is a node w in or on the circle with diameter $\overline{uv}$ (see Fig. 2). The edges $e' = (u, w)$ and $e'' = (v, w)$ are also edges of the UDG and because w lies in the described circle, we have $c_d(e')^2 + c_d(e'')^2 \leq c_d(e)^2$. If w is inside the circle with diameter $\overline{uv}$, the energy for the path $p' := p \setminus \{e\} \cup \{e', e''\}$ is smaller than the energy for p, and p is therefore not an energy-optimal path, contradicting the above assumption. If w lies exactly on the above circle, p' is an energy-optimal path as well and the argument applies recursively.

According to the optimality of $p^*_{\mathrm{GG}\cap\mathrm{UDG}}$—defined to be a shortest path on $\mathrm{GG} \cap \mathrm{UDG}$ with respect to a cost function $c(\cdot)$—we have $c(p^*_{\mathrm{GG}\cap\mathrm{UDG}}) \leq c(p) \leq \alpha \cdot c_E(p)$ for a constant α, the last inequality holding due to Lemma 3.1. Employing Lemma 3.2, we furthermore obtain $c_E(p) \leq \beta \cdot c(p^*_{\mathrm{UDG}})$ for a constant β, where p^*_{UDG} is a shortest path with respect to $c(\cdot)$ on the unit disk graph, which concludes the proof. ∎

Unless stated otherwise, it is assumed that every node locally computes its neighbors in the Gabriel Graph prior to the start of routing algorithms.

The geographic ad hoc routing algorithms we consider in this paper can be defined as follows.

Definition 3.4: (Geographic Ad Hoc Routing Algorithm): Let $G = (V, E)$ be a Euclidean graph. The task of a geographic ad hoc routing algorithm $\mathcal{A}$ is to transmit a message from a source $s \in V$ to a destination $t \in V$ by sending packets over the edges of G while complying with the following conditions:

- All nodes $v \in V$ know their geographic positions as well as the geographic positions of all their neighbors in G.
- The source s is informed about the position of the destination t.
- The control information which can be stored in a packet is limited by $O(\log n)$ bits, that is, only information about a constant number of nodes is allowed.
- Except for the temporary storage of packets before forwarding, a node is not allowed to maintain any information.

In the literature, geographic ad hoc routing has been given various other names, such as $O(1)$-memory routing algorithms in [26] and [27], local routing algorithms in [23], geometric, position-based, or location-based routing. Due to these storage restrictions, geographic ad hoc routing algorithms are inherently local. In particular, nodes do not store any routing tables, eliminating a possible source of outdated information.

Finally, it is assumed that routing takes place much faster than node movement: A routing algorithm is modeled to run on temporarily stationary nodes. The issues faced when easing or giving up this assumption were discussed in [6, Ch. 12].

IV. Greedy Routing

The probably most straightforward approach to geographic routing—which has also been studied as the first type of geographic routing algorithms in the related work—is *greedy forwarding*: Every node relays the message to be routed to its neighbor located "best" with respect to the destination. If "best" is interpreted as "closest to the destination", greedy forwarding can be formulated as follows:

Greedy Routing GR

0) Start at s.
1) Proceed to the neighbor closest to t.
2) Repeat step 1 until either reaching t or a local minimum with respect to the distance from t, that is a node v without any neighbor closer to t than v itself.

This formulation clearly reflects the simplicity of such an approach with respect to both concept and implementation. However, as indicated in Step 2 of the algorithm, it shows a big drawback: It is possible that the message runs into a "dead end", a node without any "better" neighbor. If backtracking techniques can overcome local minima in some cases, they fail to serve as a general solution to this problem, especially together with the strict message size limitations imposed on geographic routing (cf. Definition 3.4). Also alternative interpretations of "best neighbor" fail to reach the destination; in a "least deviation angle" approach for instance the message can end up in an infinite path loop [23].

If greedy routing, however, reaches the destination, it generally does so efficiently. Informally, this is due to the fact that—except in degenerate cases—the message stays relatively close to the line connecting the source and the destination. As discussed later in Section VII, employment of greedy routing whenever possible is beneficial above all in densely populated average-case networks. But also in worst-case networks the cost expended by greedy routing cannot become arbitrarily high:

Lemma 4.1: If GR reaches t, it does so with cost $O(d^2)$, where $d := |\overline{st}|$ denotes the Euclidean distance between s and t.

Proof: A detailed proof of this lemma can be found in [28]. For completeness we just give a proof sketch. Let $p := v_1, \ldots, v_k$ be the sequence of nodes visited during greedy routing. According to the definition of greedy routing, no two nodes v_i, v_j with odd indexes i, j are neighbors. Further, since the distance to t is decreasing along the path p, all nodes v_i are inside $D(t, d)$, the disk with center t and radius d. $D(t, d)$

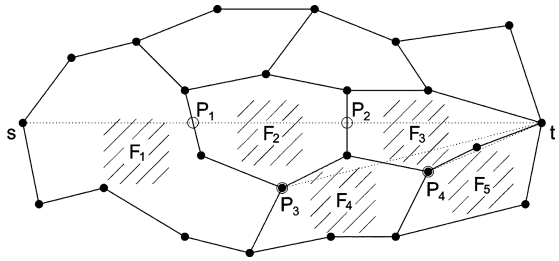

Fig. 3. Face routing starts at s, explores face F_1, finds P_1 on $\overline{st}$, explores F_2, finds P_2, and switches to F_3 before reaching t. OFR, in contrast, finds P_3, the point on F_1's boundary closest to t, continues to explore F_4, where it finds P_4, and finally reaches t via F_5.

contains at most $O(d^2)$ nodes with pairwise distance at least 1. It follows that p consists of $O(d^2)$ nodes. ■

In the following sections, greedy routing will be employed as a routing algorithm component for its efficiency in both worst-case and average-case networks.

V. Routing With Faces

In the previous section, we observed that greedy routing is not guaranteed to always reach the destination. This section introduces a type of geographic routing that, in contrast, always finds the destination if the network contains a connection from the source: routing based on faces.

A. Face Routing

The first geographic routing algorithm shown to always reach the destination was Face Routing introduced in [23]. Although we will formally describe a variant of Face Routing slightly adapted for our purposes, we will now give a brief overview of the original Face Routing algorithm.

At the heart of Face Routing lies the concept of faces, contiguous regions separated by the edges of a planar graph, that is a graph containing no two intersecting edges. The algorithm proceeds by exploration of face boundaries employing the local *right hand rule* in analogy to following the right hand wall in a maze (cf. Fig. 3). On its way around a face, the algorithm keeps track of the points where it crosses the line $\overline{st}$ connecting the source s and the destination t. Having completely surrounded a face, the algorithm returns to the one of these intersections lying closest to the destination. From here, it proceeds by exploring the next face closer to t. If the source and the destination are connected, Face Routing always finds a path to the destination. It thereby takes at most $O(n)$ steps, where n is the total number of nodes in the network.

B. AFR

Where the Face Routing algorithm can take up to $O(n)$ steps to reach the destination irrespective of the actual distance between the source and the destination in the given network, the main contribution of the Adaptive Face Routing algorithm AFR—as presented in [30]—consists in limiting the expended cost with respect to the length of the shortest path between s and t. Although the results discussed in the subsequent sections of this paper go beyond AFR, we will first provide a summary of this algorithm for completeness and continue to give an overview of the employed technique.

As mentioned, the main problem with respect to the performance of Face Routing lies in the necessity of exploring the *complete* boundary of faces. It is thus impossible to bound the cost of this algorithm by the cost of an optimal path between s and t. If, however, we know the length of an optimal path connecting the source and the destination, Face Routing can be extended to *Bounded Face Routing BFR*: The exploration of faces is restricted to a searchable area, in particular an ellipse whose size is chosen such that it contains a complete optimal path. If the algorithm hits the ellipse, it has to "turn back" and continue its exploration of the current face in the opposite direction until hitting the ellipse for the second time, which completes the exploration of the current face. Briefly put, the details will be explained later, since BFR does not traverse an edge more than a constant number of times, and since the bounding ellipse (together with the $\Omega(1)$-model and graph planarity) does not contain more than $O(|\overline{st}|^2)$ edges, the cost of BFR is in $O(c^2(p^*))$, where p^* is an optimal path connecting s and t.

In most cases, however, a prediction of the length of an optimal path will not be possible. The solution to this problem finally leads to *Adaptive Face Routing AFR*: BFR is started with the ellipse size set to an initial estimate of the optimal path length. If BFR fails to reach the destination, which will be reported to the source, BFR will be restarted with a bounding ellipse of doubled size. (It is also possible to double the ellipse size directly without returning to the source.) If s and t are connected, AFR will eventually find a path to t. This iteration is asymptotically dominated by the cost of the algorithm steps performed in the last ellipse, whose area is at most proportional to the squared cost of an optimal path. Consequently, also the cost of AFR is bounded by $O(c^2(p^*))$.

Section VI will show that in a lower-bound graph no local geographic routing algorithm can perform better: AFR is asymptotically optimal.

C. OAFR

As described in Section IV, greedy routing promises to find the destination with low cost in all cases where it arrives at the destination. A natural approach to leveraging the potential of greedy routing above all for practical purposes therefore consists in combining greedy routing and face routing. In a first attempt Greedy Routing and AFR can be literally combined: Proceed in a greedy manner and use AFR to escape from potential local minima. It has, however, been shown in [32] that, employing greedy routing, this algorithm loses AFR's asymptotic optimality. Nevertheless, a variant of AFR, named OAFR, was found whose combination with greedy routing does finally yield an algorithm that is both average-case efficient and asymptotically optimal.

Similarly to the above description of AFR, we will explain the OAFR algorithm in three steps: OFR, OBFR, and OAFR.

Other Face Routing (OFR) differs from Face Routing in the following way: Instead of changing to the next face at the "best" *intersection* of the face boundary with $\overline{st}$, OFR returns—after completing the exploration of the boundary of the current face—to the boundary point (or one of the points) closest to the destination (Fig. 3). Conserving the headway made towards the

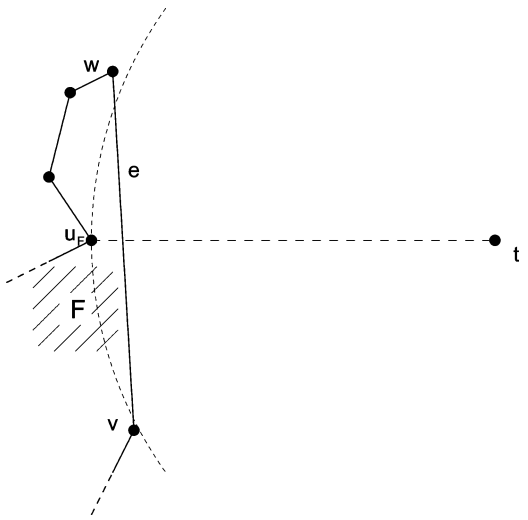

Fig. 4. OFR modified to switch to the next face at the *node* closest to t (instead of the *point* closest to t) progresses with each face switch if it runs on the Gabriel Graph. In particular it cannot happen that the algorithm is caught in an infinite loop: Having arrived at u_F, the node of face F located closest to t, the algorithm always switches to a face other than F. The only possible way of constructing a counterexample fails: A constellation forcing the modified OFR algorithm to again select F as the next face at u_F has at least one edge $e = (v, w)$ on F's boundary intersecting the line segment $\overline{u_F t}$; otherwise t would lie inside F, implying that s and t would be disconnected. The fact that both v and w are not closer to t than u_F—u_F is the node on F's boundary closest to t—implies that at least one of u_F and t are located within the disk with diameter $\overline{vw}$, which contradicts the existence of the edge e in the gabriel graph.

destination on each face, OFR in a sense uses a more natural approach than Face Routing.

Other Face Routing OFR

0) Begin at s and start to explore the face F containing the connecting line $\overline{st}$ in the immediate environment of s.
1) Explore the complete boundary of the face F based on local decisions employing the *right hand rule*.
2) Having accomplished F's exploration, advance to the point p closest to t on F's boundary. Switch to the face containing $\overline{pt}$ in p's environment and continue with step 1. Repeat these two steps until reaching t.

The number of steps taken by OFR is bounded as shown in the following lemma:

Lemma 5.1: OFR always terminates in $O(n)$ steps, where n is the number of nodes. If s and t are connected, OFR reaches t; otherwise, disconnection will be detected.

Proof: Let $F_1, F_2, \ldots, F_k$ be the sequence of the faces visited during the execution of OFR. We will first assume s and t to be connected. Since the switch between two faces always happens at the point on the face boundary closest to t and because the next face is chosen such that it always contains points which are nearer to t, no face is visited twice. Let further $p_0, p_1, p_2, \ldots, p_t$ be the *trace* of OFR's execution, where p_i, $i \geq 1$ is the point with minimum distance from t on the boundary of F_i. Because no face is visited more than once, we have that $\forall i > j : |\overline{p_i t}| < |\overline{p_j t}|$. Hence, if s and t are connected, we eventually arrive at a face with t on its boundary. (Otherwise, there is an i for which $p_i = p_{i+1}$, which means that the graph is disconnected.)

Since each face is explored at most once, each edge is visited at most four times. As every planar graph corresponds to the projection of a polyhedron on the plane, Euler's polyhedron formula can be employed: $n - m + f = 2$, where n, m, and f stand for the number of nodes, edges, and faces in the graph, respectively. Furthermore, the observations that (for $n > 3$) every face is delimited by at least three edges and that each edge is adjacent to at most two faces yield $3f \leq 2m$. Using Euler's formula we have $3m - 3n + 6 = 3f \leq 2m$ and therefore $m \leq 3n - 6$. Thus, OFR terminates after $O(n)$ steps. ■

If the algorithm detects graph disconnection (finding $p_i = p_{i+1}$ for some $i \geq 0$), this can be reported to the source by again using OFR in the reverse direction.

Remark (Gabriel Graph): When applying OFR on a Gabriel Graph, as we will do for the routing on unit disk graphs, OFR can be simplified in the following way: Instead of changing faces at the *point* on the face boundary which is closest to t, it is possible to take the *node* which is closest to t. This modification leaves the property described in Lemma 5.1 unchanged, as also the modified OFR algorithm always switches to a new face in Step 2 if it is run on the Gabriel Graph. This is illustrated in Fig. 4. Since definitions and explanations become clearer, we will use this modified form of the OFR algorithm for the description of the subsequent algorithms. Equivalent results can be achieved with the original version of the algorithm.

When trying to formulate a statement on OFR's cost, the main problem arising is its traversal of complete boundaries of faces: Informally put, OFR can meet an incredibly big face whose total exploration is prohibitively expensive compared to an optimal path from s to t. In order to solve this, AFR's trick to bound the searchable area by an ellipse containing an optimal path can be borrowed. Consequently, we obtain *Other Bounded Face Routing (OBFR)*.

For the sake of simplicity, we assume for the following description of OBFR that s and t are connected. If $\tilde{c}$ is an estimate of the Euclidean length of a shortest path between s and t, let $\mathcal{E}$ be the ellipse with foci s and t and with the length of the major axis being $\tilde{c}$ (in other words, $\mathcal{E}$ contains all paths from s to t of Euclidean length at most $\tilde{c}$).

Other Bounded Face Routing (OBFR)

0) Step 0 of OFR.
1) Step 1 of OFR, but do not leave $\mathcal{E}$: When hitting $\mathcal{E}$, continue the exploration of the current face F in the opposite direction. F's exploration will afterwards be complete when hitting $\mathcal{E}$ for the second time.
2) Step 2 of OFR with one modification: If the node closest to t on F's boundary is the same one as in the previous iteration, that is, no progress has been made in Step 1, report failure back to s by means of OBFR.

Figs. 5 and 6 illustrate the execution of OBFR if the ellipse is chosen too small and if the ellipse contains a path from s to

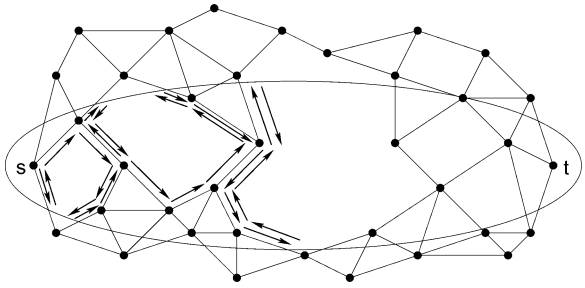

Fig. 5. Execution of OBFR if the ellipse is not chosen sufficiently large.

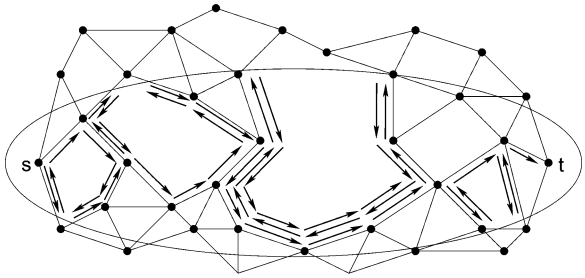

Fig. 6. Execution of OBFR if the ellipse is chosen large enough to contain a path from s to t.

t, respectively. The cost expended by OBFR can be bounded as follows:

Lemma 5.2: If the length $\tilde{c}$ of the major axis of $\mathcal{E}$ is at least the length of a—with respect to the Euclidean metric—shortest path between s and t, OBFR reaches the destination. Otherwise OBFR reports failure to the source. In any case, OBFR expends cost at most $O(\tilde{c}^2)$.

Proof: We first assume that $\tilde{c}$ is at least the length of a shortest (Euclidean) path p^*, that is, p^* is completely contained in $\mathcal{E}$. Since OBFR stays within $\mathcal{E}$ while routing a message, we only look at the part of the graph which lies inside $\mathcal{E}$. We define the faces to be those contiguous regions which are separated by the edges of the graph and by the boundary of $\mathcal{E}$. (Hence, if a face is cut into several pieces by the boundary of $\mathcal{E}$, now each such piece is denoted a face.) Assume for the sake of contradiction that OBFR reports failure, that is, the algorithm does not make progress in Step 2. This is only possible if the currently traversed face boundary cuts the area enclosed by the ellipse into a region containing s and a second region containing t (cf. Fig. 5). In this case, however, $\mathcal{E}$ does not contain any path connecting s and t, which contradicts our assumption and therefore proves the first claim of the lemma.

If no path connects s and t within $\mathcal{E}$, a face boundary separating s from t as described in the previous paragraph exists. This is detected by OBFR, making no progress beyond a node v. As OBFR reached v starting from s, $\mathcal{E}$ contains a path from v to s and OBFR can be restarted in the opposite direction with the same ellipse, eventually reaching s and reporting failure.

Finally, it remains to be shown that the cost expended does not exceed $O(\tilde{c}^2)$. If $\mathcal{E}$ contains a path connecting s and t, every face is, for the same reasons as for OFR, visited at most once. Otherwise, every face is visited at most twice (the face where failure is detected can be visited an additional time). Furthermore, during the traversal of a face boundary, each edge can be visited at most four times. Consequently, any edge is traversed at most a constant number of times during the complete execution of OBFR.

Due to the planarity of the considered graph, the number of edges is linear in the number of nodes (cf. proof of Lemma 5.1). Furthermore, according to the employed $\Omega(1)$-model, the circles of radius $d_0/2$ around all nodes do not intersect each other. Since the length of the semimajor axis a of the ellipse $\mathcal{E}$ is $\tilde{c}/2$, and since the area of $\mathcal{E}$ is smaller than πa^2, the number of nodes n' inside $\mathcal{E}$ is bounded by

$$n' \leq \frac{\pi a^2}{\pi \left(\frac{d_0}{2}\right)^2} = \frac{\tilde{c}^2}{d_0^2} \in O(\tilde{c}^2).$$

Having thus found an upper bound for the number of messages sent, the last statement of the lemma follows with Lemma 2 for all cost metrics defined according to Definition 3.2. ■

Note that the above specification of OBFR omits an important point for clarity of representation: The algorithm can distinguish between the case where the chosen ellipse is not sufficiently large to contain a path connecting s with t and the case where s and t are disconnected. As described above, the first case is detected if no progress is made after hitting the ellipse. Also in the second case there exists a node beyond which no progress is made; however, this node is detected as such without hitting the ellipse, that is, after traversing the complete boundary of the network component containing the source.

Since there is usually no *a priori* information on the optimal path length, initially, in analogy to AFR, a small estimate for the ellipse size is used and iteratively enlarged until reaching the destination.

Other Adaptive Face Routing OAFR

0) Initialize $\mathcal{E}$ to be the ellipse with foci s and t the length of whose major axis is $2 \cdot |\overline{st}|$.
1) Start OBFR with $\mathcal{E}$.
2) If the destination has not been reached, double the length of $\mathcal{E}$'s major axis and go to step 1.

Exploiting that OBFR is able to distinguish between insufficient ellipse size and graph disconnection between s and t, also OAFR detects graph disconnection. Furthermore, OAFR's cost is bounded as follows:

Theorem 5.3: If s and t are connected, OAFR reaches the destination with cost $O(c^2(p^*))$, where p^* is an optimal path. If s and t are disconnected, OAFR detects so and reports to s.

Proof: We denote the first estimate $\tilde{c}$ on the optimal path length by $\widetilde{c_0}$ and the consecutive estimates by $\widetilde{c_i} := 2^i \widetilde{c_0}$. Furthermore, we define k such that $\widetilde{c_{k-1}} < c_d(p^*) \leq \widetilde{c_k}$. For $c(\mathrm{OBFR}[\tilde{c}])$, the cost of OBFR with the length of $\mathcal{E}$'s major axis set to $\tilde{c}$, we have $c(\mathrm{OBFR}[\tilde{c}]) \in O(\tilde{c}^2)$ and therefore

$$c_\ell\left(\mathrm{OBFR}[\tilde{c}]\right) \leq \lambda \cdot \tilde{c}^2$$

for a constant λ (and sufficiently large $\tilde{c}$). The total cost of OAFR can therefore be bounded by

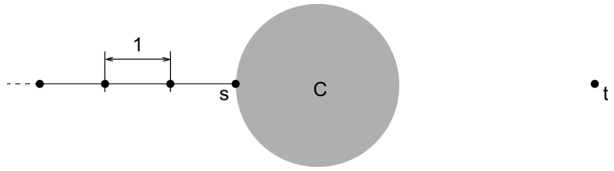

Fig. 7. If $n'/2$ nodes are located in the node cluster C (represented as a gray disk) and $n'/2$ nodes form the spike on the left, OAFR executes its component OBFR $\Theta(\log n')$ times, each OBFR execution having cost in $\Theta(n')$ if the nodes in C are placed in a maze-like structure, before detecting that s and t are disconnected.

$$\begin{aligned} c_\ell(\mathrm{OAFR}) &\leq \sum_{i=0}^{k} c_\ell\left(\mathrm{OBFR}[\tilde{c}_i]\right) \leq \sum_{i=0}^{k} \lambda(2^i\tilde{c}_0)^2 \\ &= \lambda\tilde{c}_0^2\frac{4^{k+1}-1}{3} < \frac{16}{3}\lambda(2^{k-1}\tilde{c}_0)^2 \\ &< \frac{16}{3}\lambda\cdot \mathrm{c}_\mathrm{d}^2(p^*) \in O\left(\mathrm{c}_\mathrm{d}^2(p^*)\right). \end{aligned}$$

■

Remark: It can be shown that for OAFR (and also AFR) the cost of detecting disconnection between s and t is bounded by $O(n'\log n')$, where n' is the number of nodes in the network component containing s: The number of OBFR executions is at most in $O(\log n')$, while the cost expended by OBFR in each of these executions is at most linear in n'. As illustrated in Fig. 7, there exist graphs for which OAFR expends cost $\Theta(n'\log n')$.

VI. A Lower Bound

As presented in the previous section, the OAFR algorithm reaches the destination with cost $O(c^2)$, where c is the cost of the shortest path between the source and the destination. A natural question arising is whether this guarantee is good or if there are algorithms that can perform better. The following will give an answer to this question by showing that no geographic routing algorithm according to Definition 3.4 can find the destination with lower cost. In particular a constructive lower bound is given:

Theorem 6.1: Let the cost of a best route for a given source-destination pair be c. There exist graphs where any deterministic (randomized) geographic ad hoc routing algorithm has (expected) cost $\Omega(c^2)$ for any cost metric according to Definition 3.2.

Proof: A family of networks is constructed as follows. We are given a positive integer k and define a Euclidean graph G (see Fig. 8): On a circle we evenly distribute $2k$ nodes such that the distance between two neighboring points is exactly 1; thus, the circle has radius $r \approx k/\pi$. For every second node of the circle we construct a chain of $\lceil r/2\rceil - 1$ nodes. The nodes of such a chain are arranged on a line pointing towards the center of the circle; the distance between two neighboring nodes of a chain is exactly 1. Node w is one arbitrary circle node with a chain: The chain of w consists of $\lceil r\rceil$ nodes with distance 1. The last node of the chain of w is the center node; the edge to the center node does not need to have length 1.

The unit disk graph consists of the edges on the circle and the edges on the chains only. In particular, there is no edge between two chains because all chains except the w chain end strictly outside radius $r/2$. The graph has k chains with $\Theta(k)$ nodes each.

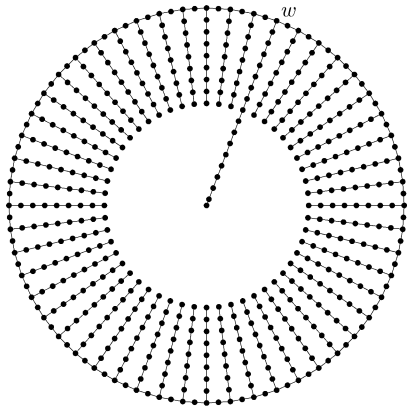

Fig. 8. Lower-bound graph.

We route from an arbitrary node on the circle (the source s) to the center of the circle (the destination t). An optimal route between s and t follows the shortest path on the circle until it hits node w, and then directly follows w's chain to t with link cost $c \leq k + r + 1 \in O(k)$. A routing algorithm with routing tables at each node will find this best route.

A geographic routing algorithm, in contrast, needs to find the "correct" chain w. Since there is no routing information stored at the nodes, this can only be done by exploring the chains. Any deterministic algorithm needs to explore the chains in a deterministic order until it finds the chain w. Thus, an adversary can always place w such that w's chain will be explored as the last one. The algorithm will therefore explore $\Theta(k^2)$ (instead of only $O(k)$) nodes.

The argument is similar for randomized algorithms. By placing w accordingly (randomly!), an adversary forces the randomized algorithm to explore $\Omega(k)$ chains before chain w with constant probability. Then the expected link cost of the algorithm is $\Omega(k^2)$.

As all edges (but one) in our construction have length 1, the cost values in the Euclidean distance, the link distance, and the energy metrics are equal. As for any fixed cost metric $c'(\cdot)$ according to Definition 3.2 $c'(1)$ is also a constant, the $\Omega(c^2)$ lower bound holds for all according metrics. ■

Note that this lower bound holds generally, not only for $\Omega(1)$-graphs. (As shown in [33], however, a similar lower bound proves that in general graphs the cost metrics according to Definition 3.2 fall into two classes.)

Given this lower bound, we can now state that OAFR is asymptotically optimal for unit disk graphs in the $\Omega(1)$-model.

Theorem 6.2: Let c be the cost of an optimal path for a given source–destination pair on a unit disk graph in the $\Omega(1)$-model. In the worst case, the cost for applying OAFR to find a route from the source to the destination is $\Theta(c^2)$. This is asymptotically optimal.

Proof: This theorem is an immediate consequence of Theorems 5.3 and 6.1. ■

VII. Combining Greedy and Face Routing

A greedy routing approach as presented in Section IV is not only worth being considered due to its simplicity in both concept

and implementation. Above all, in dense networks such an algorithm can also be expected to find paths of good quality; here, the straightforwardness of a greedy strategy contrasts highly the inflexible exploration of faces inherent to face routing. For practical purposes, it is inevitable to improve the performance of a face routing variant, for instance by leveraging the potential of a greedy approach.

In this section, we will briefly outline the GOAFR^{+}[3] algorithm combining greedy and face routing [33]. GOAFR^{+} is a combination of greedy routing and face routing in the following sense: Whenever possible, the algorithm tries to route in a greedy manner; in order to overcome local minima with respect to the distance from the destination, face routing is employed. In face routing mode, GOAFR^{+} restricts the searchable area in a similar way as OAFR. Simulations showed that choosing a circle centered at the destination t instead of an ellipse and above all gradually reducing its radius while the message approaches t improves the average-case performance.

More importantly, for average-case considerations, the algorithm should fall back to greedy routing as soon as possible after escaping the local minimum. This is suggested by the observation that greedy forwarding is, especially in dense networks, more efficient than face routing in the average case. Accordingly, GOAFR^{+} tries to return to greedy routing as early as possible. However, it was shown [32], [33] that this must not be done too simplistically, such as whenever the algorithm in face routing mode is closer to the destination than the escaped from local minimum, as this would happen at the expense of the algorithm's asymptotic optimality. In order to preserve this property, the GOAFR^{+} algorithm employs two counters p and q to keep track of how many of the nodes visited during the current face routing phase are located closer to the destination (counted with p) and how many are at least as far from the destination (counted with q) than the starting point of the current face routing phase; as soon as a certain fallback condition holds (whose discussion would exceed the scope of this paper), GOAFR^{+} continues in greedy mode.

It can be proved that the GOAFR^{+} algorithm retains OAFR's asymptotic optimality, in particular, that it is guaranteed to reach the destination with cost at most $O(c^2(p^*))$, where p^* is an optimal path between the source and the destination [33].

On the other hand it was shown by simulation in networks generated by randomly placing nodes in a given field that GOAFR^{+}'s combination of face routing with greedy forwarding is beneficial for routing also in average-case networks [6], [32], [33]. Particularly, in order to judge the practicability of a geographic routing algorithm, the *normalized cost* of an algorithm A in a network N given a source s and a destination t, defined as

$$cost_A(N,s,t) := \frac{s_A(N,s,t)}{c_\ell\left(p^*_\ell(N,s,t)\right)}$$

was measured, where $s_A(N,s,t)$ is the number of steps taken by algorithm A in network N finding a route from s to t;

[3]Expressing the combination of the greedy and face routing approaches, the acronym GOAFR stands for Greedy Other Adaptive Face Routing. The "+" sign indicates that GOAFR^{+} (pronounced as "gopher-plus") is an improvement over a previously defined similar algorithm with the name GOAFR.

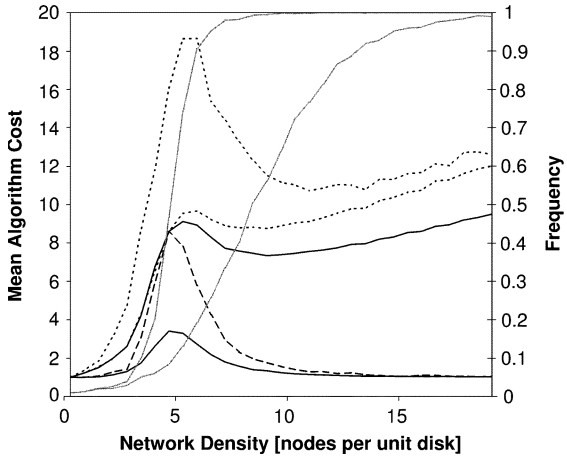

Fig. 9. Algorithm performance in network densities including the critical density range around 4.5 nodes per unit disk. Mean cost values of FR (upper dotted line), AFR (lower dotted), OAFR (upper solid), GFG/GPSR (dashed), and GOAFR^{+} (lower solid). The network connectivity (upper gray line) and greedy success rates (lower gray) are plotted against the right y-axis.

$c_\ell(p^*_\ell(N,s,t))$ is the (hop) length of the shortest path (with respect to the hop metric) between the source s and the destination t in N.

The GOAFR^{+} algorithm was compared with the FR algorithm (Face Routing, see Section V-A), AFR, OAFR, and GFG/GPSR (a combination of greedy and face routing introduced in [24] and [31]). Fig. 9 shows for each simulated algorithm A its mean cost value, that is

$$\overline{cost}_A := \frac{1}{k}\sum_{i=1}^{k} cost_A(N_i,s_i,t_i)$$

over $k = 2000$ generated triples (N_i,s_i,t_i) for network densities ranging from 0.3 to 20 nodes per unit disk.

In the context of routing in networks formed by randomly placed nodes, node density plays an important role. For very low network densities, the network will almost always be disconnected; for high densities, on the other hand the network can be expected to consist of one connected component with high probability. The transition between these two extremes is astonishingly narrow. This can be observed in Fig. 9, where, in addition to the algorithm cost values, the network connectivity and greedy success rates are plotted against the right y-axis. The network connectivity rate is defined as the portion of triples (N_i,s_i,t_i) in which s_i and t_i are connected in network N_i of a given node density; the greedy success rate represents the portion of randomly generated triples (N_i,s_i,t_i) in which—starting from s_i—the destination t_i can be reached using greedy forwarding alone.

In Fig. 9, the cost value series for both combinations of greedy and face routing display favorable values for low and high values: With very low densities, if the source and the destination are in the same connected component, they are very likely close together; with high densities, greedy routing alone will efficiently reach the destination with high probability. In between, all simulated algorithms are more or less bell-shaped around the critical density range of about 4.5 nodes per unit disk. This behavior is due to the fact that typically in networks of critical density the shortest path connecting the source and

the destination is significantly longer than their Euclidean distance. As reflected by the cost values, this critical density range appears to be a challenge for any routing algorithm. Not only this effect is best tolerated by the GOAFR$^+$ algorithm. More generally, the simulation results show that GOAFR$^+$ clearly outperforms all previously known geographic routing algorithms over a broad network density range.

VIII. Conclusion

The GOAFR$^+$ geographic routing algorithm presented in this paper forms a combination of the greedy forwarding and face routing approaches. Employing face routing, enhanced by a limitation to a searchable area and a counter technique, the algorithm is proved to require at most $O(c^2)$ steps where c is the cost of a shortest path connecting the source and the destination; together with a corresponding lower bound graph, this guarantee is shown to be asymptotically optimal. Using greedy forwarding, the algorithm also becomes efficient in average-case networks, as shown by simulation and comparison with similar algorithms. In this sense, the GOAFR$^+$ algorithm can be considered a synthesis of simplicity and average-case efficiency on the one hand and correctness as well as asymptotic worst-case optimality on the other hand.

More generally, the GOAFR$^+$ algorithm may stand as an example that it is possible to design algorithms with theoretically proved worst-case guarantees and more practically relevant average-case efficiency. Maybe, even the general conclusion can be drawn that accounting for worst-case behavior before studying the average case appears to be easier than conversely, which may serve as a design principle beyond the scope of ad hoc and sensor networks.

References

[1] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. ACM MobiCom*, 1998, pp. 85–97 [Online]. Available: citeseer.nj.nec.com/broch98performance.html

[2] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. ACM MobiCom*, Aug. 2000, pp. 120–130.

[3] I. Abraham, D. Dolev, and D. Malkhi, "LLS: A locality aware location service for mobile ad hoc networks," in *DIALM-POMC '04: Proc. 2004 Joint Workshop on Foundations of Mobile Computing*, 2004, pp. 75–84.

[4] M. Grossglauser and M. Vetterli, "Locating nodes with EASE: Mobility diffusion of last encounters in ad hoc networks," in *Proc. 22nd Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[5] R. Flury and R. Wattenhofer, "MLS: An efficient location service for mobile ad hoc networks," in *Proc. 7th ACM MobiHoc*, Florence, Italy, May 2006.

[6] A. Zollinger, "Networking unleashed: Geographic routing and topology control in ad hoc and sensor networks," Ph.D. dissertation, ETH, Zurich, Switzerland, 2005, diss. ETH 16025.

[7] C. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM'94*, 1994, pp. 234–244.

[8] R. Ogier, F. Templin, and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," IETF RFC 3684, Feb. 2004.

[9] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF RFC 3626, Oct. 2003.

[10] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.

[11] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds. Boston, MA: Kluwer Academic, 1996, vol. 353 [Online]. Available: citeseer.nj.nec.com/johnson96dynamic.html

[12] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. 16th Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM)*, 1997, pp. 1405–1413 [Online]. Available: citeseer.ist.psu.edu/article/park97highly.html

[13] M. Joa-Ng and I.-T. Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1415–1425, Aug. 1999.

[14] N. Nikaein, C. Bonnet, and N. Nikaein, "HARP—Hybrid Ad Hoc Routing protocol," in *Int. Symp. Telecommunications (IST 2001)*, Tehran, Iran, Sep. 2001.

[15] V. Ramasubramanian, Z. Haas, and E. Sirer, "SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks," in *Proc. ACM MobiHoc*, 2003.

[16] E. Royer and C. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Commun.*, vol. 6, no. 2, pp. 46–55, Apr. 1999.

[17] C. Busch, S. Surapaneni, and S. Tirthapura, "Analysis of link reversal routing algorithms for mobile ad hoc networks," in *Proc. 15th Annu. ACM Symp. Parallel Algorithms and Architectures (SPAA)*, San Diego, CA, Jun. 2003.

[18] E. Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Trans. Commun.*, vol. 29, no. 1, pp. 11–18, Jan. 1981.

[19] R. Prakash, "Unidirectional links prove costly in wireless ad hoc networks," in *Proc. 3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999.

[20] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," USC/ISI, Tech. Rep. ISI/RR-87-180, Mar. 1987.

[21] T. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Trans. Commun.*, vol. 34, no. 1, pp. 38–44, Jan. 1986.

[22] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. Commun.*, vol. 32, no. 3, pp. 246–257, Mar. 1984.

[23] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. 11th Canadian Conf. Computational Geometry*, Vancouver, BC, Canada, Aug. 1999, pp. 51–54 [Online]. Available: citeseer.nj.nec.com/kranakis99compass.html

[24] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. 3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999, pp. 48–55 [Online]. Available: citeseer.nj.nec.com/bose00routing.html

[25] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," in *Cluster Computing 5*. Boston, MA: Kluwer Academic, 2002, pp. 169–178.

[26] P. Bose, A. Brodnik, S. Carlsson, E. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and J. Munro, "Online routing in convex subdivisions," in *Int. Symp. Algorithms and Computation (ISAAC)*, 2000, vol. 1969, Lecture Notes in Computer Science, pp. 47–59 [Online]. Available: citeseer.nj.nec.com/bose00online.html, Springer

[27] P. Bose and P. Morin, "Online routing in triangulations," in *Proc. 10th Int. Symp. Algorithms and Computation (ISAAC)*, 1999, vol. 1741, Springer LNCS, pp. 113–122 [Online]. Available: citeseer.nj.nec.com/article/bose99online.html

[28] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proc. ACM MobiHoc*, Long Beach, CA, Oct. 2001.

[29] J. Urrutia, "Routing with guaranteed delivery in geometric and wireless networks," in *Handbook of Wireless Networks and Mobile Computing*, I. Stojmenovic, Ed. New York: Wiley, 2002, ch. 18, pp. 393–406.

[30] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad hoc routing," in *Proc. 6th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 2002, pp. 24–33.

[31] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom*, 2000, pp. 243–254 [Online]. Available: citeseer.nj.nec.com/karp00gpsr.html

[32] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad hoc routing," in *Proc. ACM MobiHoc*, 2003.

[33] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric routing: Of theory and practice," in *Proc. 22nd ACM Symp. Principles of Distributed Computing (PODC)*, 2003.
[34] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proc. Second USENIX/ACM Symp. Networked System Design and Implementation (NSDI 2005)*, Boston, MA, May 2005.
[35] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "On the pitfalls of geographic face routing," in *Proc. ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Cologne, Germany, Sep. 2005.
[36] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing: Geographic routing with local face information," in *13th IEEE Int. Conf. Network Protocols (ICNP)*, Boston, MA, Nov. 2005.
[37] B. Leong, B. Liskov, and R. Morris, "Geographic routing without planarization," in *3rd Symp. Networked Systems Design & Implementation (NSDI)*, San Jose, CA, May 2006.
[38] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006.
[39] C. Chiang, H. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks," in *Proc. IEEE Singapore Int. Conf. Networks*, 1997, pp. 197–211.
[40] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan, "A cluster-based approach for routing in ad hoc networks," in *Proc. 2nd USENIX Symp. Mobile and Location-Independent Computing*, 1995, pp. 1–10.
[41] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing newly deployed ad hoc and sensor networks," in *Proc. ACM MobiCom*, Philadelphia, PA, Oct. 2004.
[42] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," Univ. of Virginia, Charlottesville, VA, Tech. Rep. CS-2003-11, 2003.
[43] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Elsevier's Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, Nov. 2003.
[44] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct.–Dec. 2003.
[45] M. Heissenbüttel, "Routing and broadcasting in ad hoc networks," Ph.D. dissertation, Univ. of Bern, Bern, Switzerland, 2005.
[46] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Unit disk graph approximation," in *Proc. ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Philadelphia, PA, Oct. 2005.
[47] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, "Virtual coordinates for ad hoc and sensor networks," in *Proc. ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, Philadelphia, PA, Oct. 2005.
[48] M. Wattenhofer, R. Wattenhofer, and P. Widmayer, "Geometric routing without geometry," in *12th Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Le Mont Saint-Michel, France, May 2005.
[49] R. Fonseca, S. Ratnasamy, J. Zhao, C.-T. Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon vector routing: Scalable point-to-point routing in wireless sensornets," in *Proc. 2nd Symp. Networked Systems Design and Implementation (NSDI)*, Boston, MA, May 2005.
[50] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.
[51] L. Barrière, P. Fraigniaud, and L. Narayanan, "Robust position-based routing in wireless ad hoc networks with unstable transmission ranges," in *Proc. 5th ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 2001, pp. 19–27.
[52] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," in *Proc. 1st ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, San Diego, CA, Sep. 2003.
[53] P. Doyle and J. Snell, *Random Walks and Electric Networks*, ser. The Carus Mathematical Monographs. Washington DC: The Mathematical Association of America, 1984.
[54] H. Hunt, III, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns, "NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs," *J. Algorithms*, vol. 26, no. 2, pp. 238–274, 1998.

Fabian Kuhn (M'07) 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 Prof. Roger Wattenhofer at ETH Zurich as a Ph.D. student and research assistant. In 2005, he received the Ph.D. degree for his work on locality phenomena in distributed algorithms.

He is currently with Microsoft Research, Mountain View, CA.

Roger Wattenhofer received the Doctorate in computer science from ETH Zurich, Switzerland, in 1998.

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, and since July 2004 as an Associate Professor at the Information Technology and Electrical Engineering Department. His 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 (M'07) received the 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 received the 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.