Virtual-Force-Based Geometric Routing Protocol in MANETs

Cong Liu and Jie Wu, *Fellow*, *IEEE*

Abstract—Routing is the foremost issue in mobile ad hoc networks (manets). To guarantee delivery and improve performance, most position-based routing protocols, e.g., *greedy-face-greedy* (GFG), forward a message in greedy routing mode until the message is forwarded to a *local minimum* where greedy forwarding is impossible. They then switch to a less efficient mode known as face routing. Face routing requires the underlying network to be a planar graph which makes geometric routing only theoretically feasible. To remove this constraint, this paper tackles the local minimum problem with two new methods. First, we construct a virtual small-world network by adding virtual long links to the network to reduce the number of local minima. Second, we use the virtual force method to recover from local minima without relying on face routing. Combining these two methods, we propose a purely greedy routing protocol, the *small-world iterative navigation greedy* (SWING+) routing protocol. Simulations are conducted to evaluate SWING+ against existing geometric routing protocols. Simulation results show that SWING+ guarantees delivery, and that its performance is comparable to that of the state-of-the-art *greedy other adaptive face routing* (GOAFR+) routing protocol.

Index Terms—Position-based (geometric) routing, mobile ad hoc networks (manets), simulation, small-world model, virtual force.

1 Introduction

A mobile ad hoc network (manet) is comprised solely of wireless stations. The communication between source and destination nodes may require a traversal of multiple hops due to a limited radio range. Existing routing algorithms can be broadly classified into topology-based and position-based routing protocols. Topology-based routing determines a route based on complete network topology as state information.

This paper focuses on *position-based routing* (also called *geometric or geographic routing*), which is attractive as a routing protocol in relatively static manets because they have lower route discovery overhead than the proactive/reactive topology-based routing protocols, which employ some kind of flooding. In geometric routing, it is assumed that each node is aware of its position and the positions of their neighbors, and that the source is aware of the destination's position. Efficient position-based routing protocols use greedy forwarding as their basic operation. In greedy forwarding, if a node knows its neighbors' positions, the locally optimal choice for the next hop is the neighbor that is geographically closest to the destination of the message. Greedy forwarding, however, fails in a local minimum where the only route to the destination requires a packet to move temporarily farther in geometric distance from the destination.

In order to recover from a local minimum, most existing protocols switch to a less efficient mode known as face routing [3]. Face routing runs on a planar graph (a graph without crossing edges), in which the message is routed around the perimeter of the face (an area surrounded by the edges in the planar graph) which intersects the line between the source and the destination. Most efficient geometric routing protocols are greedy-face combinations, such as *greedy-face-greedy* (GFG) [2], and its variants *greedy perimeter stateless routing* (GPSR) [8], *greedy other adaptive face routing* (GOAFR) [13], and GOAFR+ [12]. Though these algorithms guarantee delivery, they rely on the intrinsic faces of the planar graph which can only be constructed locally in a unit disk graph (UDG). Unlike the greedy-face combinations, our proposed algorithm does not rely on face routing. Thus, it removes the constraint of planar graph and UDG, and is applicable in any radio model and 3D networks.

This paper tackles the local minimum problem using two new methods other than face routing. The first method is to construct a virtual small-world network. Specifically, each node in the network constructs some remote contacts connected by *virtual long links* (VLLs). Each VLL consists of multiple consecutive physical links. To make the approach scalable, the lengths of the VLLs (in terms of hop-count) conform to a 2-exponent power-law distribution, which is analogous to [9]. One purpose of introducing VLLs is to reduce the number of local minima for each destination and increase the success rate of the greedy routing.

The second method is the *virtual force* (VF)-based greedy routing. It can recover from local minima without resorting to face routing. In this method, a message is forwarded along the increasing gradient of the *composition of the VFs*. Each VF has a source. The destination is the only source of an attractive (positive) VF. Whenever the greedy method fails in a local minimum under the composition of the VFs, a new source of a repulsive (negative) VF is added to the local minimum to expel the message from it. Thus, the local

- *The authors are with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431.*
E-mail: cliu8@fau.edu, jie@cse.fau.edu.

Manuscript received 23 May 2007; revised 8 Jan. 2008; accepted 21 May 2008; published online 10 June 2008.
Recommended for acceptance by S. Das.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-05-0162.
Digital Object Identifier no. 10.1109/TPDS.2008.103.

minimum is removed, and greedy routing is recovered. This method is an iteration of greedy forwarding and local minimum removal. We call it the *iterative navigation greedy* (ING) protocol. In ING, a list of the past local minima needs to be piggybacked in the message.

Our new, purely greedy protocol, which combines the virtual small-world network and the ING, is called the *small-world iterative navigation greedy* (SWING+) protocol. SWING+ is an extension of our previous works [15], [16]. We prove that SWING+ guarantees delivery and derive a bound for the worst-case route discovery delay in SWING+. SWING+ is a purely greedy routing algorithm where a message is always forwarded greedily under the composition of the VFs leading it to the destination. Simulation results confirm that SWING+ guarantees delivery, and that the performance of SWING+ is comparable to that of the state-of-the-art position-based routing protocol GOAFR+ [12] in terms of route discovery delay and route length.

To sum up, the contributions in this paper are the following:

1. Based on the concept of the small-world network, we increase the virtual connectivity of the network and reduce the number of local minima in the network.
2. We propose a VF-based geometric routing which recovers a greedy algorithm from local minima without relying on a planar graph and a UDG. To the best of our knowledge, it is the first delivery-guaranteed geometric routing that is applicable in nonplanar and non-unit-disk graphs.
3. We prove that our algorithm guarantees delivery in connected networks and derive a bound for its route discovery delay.
4. We conduct simulations to evaluate the routing performance of SWING+ compared to the existing ones and their extensions (with VLLs).

Simulation results show that SWING+ has similar performance results to the state-of-the-art algorithm in 2D networks in terms of route discovery time and route length.

To simplify our discussion, we adopt the following common assumptions in the context of geometric routing. We assume that 1) all nodes know their own positions, probably from a GPS device [4]; 2) a location registration and lookup service that precisely maps node addresses to locations [14] and queries to this system can use the same geographic routing system as data packets; 3) each data transmission has two phases: route discovery and data delivery, which are reasonable when the data to be transmitted is of large volume.

The rest of this paper is organized as follows: Section 2 briefly summarizes the related works. We present SWING+ in Sections 3 and 4, where we detail the formation of the virtual small-world network, the greedy routing in a small-world network, and the ING protocol. Simulation results are shown in Section 5. Overhead and scalability of SWING+ are analyzed in Section 6. Finally, Section 7 concludes this paper.

2 Related Works

SWING+ is a geometric routing which is built on the insights from some works on the small-world theory. In this section, we will review some related works in geometric routing and in small-world theory, respectively.

2.1 Position-Based Routing

Localized algorithms are desirable in manets, in which nodes make routing decisions based only on the information about their neighboring nodes and the position of the destination. One such method, the greedy routing algorithm, in which each node forwards the message to its neighbor closest to the destination, is based on the location information supplied by GPS [4].

In GFG [2] and its variant GPSR [8], a message starts with greedy routing until it reaches a local minimum where greedy forwarding is impossible. The algorithm then changes to face routing mode which forwards the message along the perimeter of the face in the planar graph of the network which is next to the local minimum in the direction of the destination. Face routing can forward the message in either direction along the perimeter of the face using the left-hand rule or the right-hand rule. In the left (right)-hand rule traversal, the traveler travels around the perimeter placing its left (right) hand on the perimeter. Face routing mode stops and reverts back to greedy routing when the message is forwarded to a node that is closer to the destination than the local minimum. GFG switches between greedy mode and face mode in order to make sure that the message is continuously getting closer to the destination.

The connectivity graphs of wireless networks typically contain many crossing edges. Face routing must run on a planar subgraph. Localized methods for obtaining a planar subgraph include Gabriel Graph (GG) [6], Relative Neighborhood Graph (RNG) [20], and Localized Delaunay Triangulation (LDT) [7]. To reduce transmissions, Datta et al. [5] proposed to run GFG in its face routing mode on the network of internal nodes which is defined as a connected dominating set (CDS). *Greedy-void-greedy* (GVG) [18] has its face forwarding section run over arbitrary nonplanar graphs by adding virtual nodes at the crosses of the links.

GOAFR [13] and GOAFR+ [12] use a distance-bounded face traversal. This traversal iteratively traverses both sides of the face for a bounded distance with the right-hand rule and the left-hand rule, respectively, and increases the bound if the condition to return to greedy forwarding mode is not satisfied after some iterations. The resulting paths of GOAFR and GOAFR+ are asymptotically optimal and are, on average, the shortest among the proposed geometric routing protocols.

Similar to [5], Zhang et al. [22] proposed a face tracing geometric routing based on an embedding network of clusterheads. This method allows geometric routing in nonplanar networks. However, it uses complicated methods to deal with the nonplanar cluster graph, and it is not applicable to general 3D networks. Geometric routings using virtual positioning, such as the one presented in [11], aim at reducing local minima and preventing the face routing mode to be invoked. However, these methods usually converge very slowly that they need a long heat-up time and that they cannot tolerate the frequent topology changes in manets very well.

2.2 The Small World Model

The small-world model [17] corresponds to a phenomenon in a social network where any two people have “six degrees of separation,” which means that any two people can be connected through six intermediate acquaintances. More recently, it has been shown in [21] that this phenomenon is pervasive in many natural and artificial complex networks, and is captured by two measurements: small average path length and high clustering coefficient (defined as the average fraction of pairs of neighbors of a node that are also neighbors of each other).

Kleinberg [9] defined an infinite family of random network models that seek a simple framework that encapsulates the paradigm of Watts and Strogatz—rich in local connections with a few long-range connections. Rather than using a ring as the basic structure, this model uses a 2D $n \times n$ grid and allows each node to have a directional long link to a remote contact with the distance in the r-exponent power-law distribution. It is proved in [21] that there is a unique “navigable” model $(r = 2)$ within the family for which decentralized algorithms are bounded by $O(\log^2 n)$. An extension to the navigable hierarchical network is discussed in [10].

Terminode [1] is based on the small-world model that does not always forward packets directly toward the destination. In order to optimize routing in case of voids in the network topology, a node finds a list of remote contacts distributed all over the network to which it maintains a good path. To find a route to the destination, a node asks its remote contacts which in turn ask their remote contacts, and so on. The right remote contacts found are added as a loose source path to the header of the data packets. Though Terminode finds short paths, it uses some sort of broadcast to discover routes.

3 Small World Networks

SWING+ improves routing performance by using a small-world network which provides a richer connectivity for making routing decisions. We follow the small-world model of Watts and Strogatz in which individual nodes are rich in local connections with a few long-range connections. In this section, we present the establishment and maintenance of small-world networks in manets in which VLLs are used to establish long-range connections.

3.1 Small World Network Formation

In this section, we will present our algorithm to construct a virtual small-world network by adding a number of VLLs to each node in the network. A VLL can be considered as a virtual path between an arbitrary pair of nodes in the network, rather than as a path for a specific pair of source and destination as in traditional routing such as DSR and AODV. Considering scalability, the length of each VLL is under the power-law distribution. In our algorithm, each node periodically sends out VLL discovery messages which go away and then come back to report a VLL. The first problem here is the determination of the maximum hop and direction of a message. The second problem is the selection of a subset of the most valuable VLLs when the storage in each node is limited.

When a VLL discovery message (referred to simply as a message in this section) is sent by a node, the message should go in a different direction from those of the previous messages to explore a different part of the network. Also, for each message, a maximum hop should be appointed in a way that is scalable to the network size. Our method is designed to associate a random maximum hop $MaxHop$ to each message, which conforms to the power-law distribution. $MaxHop$ is generated using the following equation:

$$MaxHop = MinHop + \log_2\left(\frac{1}{q}\right). \tag{1}$$

Here, q is a uniformly random value between 0 and 1, and $MinHop$ is a constant which specifies the minimum hop of any VLL. $MinHop$ is 2 in our experiment.

The reason for choosing the 2-exponent power-law distribution is twofold. First, an analytical study in [9] shows that there is an analogous small-world model (an $n \times n$ grid where the distances between remote contacts are 2-exponent power-law distributed) for which decentralized algorithms are bounded by $O(\log^2 n)$. The second reason to use the 2-exponent power-law distribution is for scalability: only when the exponent is less than or equal to 2 will the average VLL length not grow as the network size increases.

3.2 The Entropy of a Set of VLLs

We use the concept of *entropy* in the formation and the maintenance of our virtual small-world network. The Shannon information entropy is, simply speaking, a measure of the discrepancy of a set of data points in a system. Entropy is also used in other fields, such as data mining.

We say that a set of VLLs is good if they reach different parts of the network. This can be measured by the distances between the end points of these VLLs. Thus, we can use entropy to measure the position discrepancy of the end point of a set of VLLs. If the entropy is large, the set of VLLs is good, and vice versa. We used Renyi’s entropy [19], and for a set of nodes V, it is defined as follows:

$$G(v_i, v_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d(v_i,v_j)}{2\sigma^2}} \tag{2}$$

$$Entropy(V) = -\ln \frac{\sum_{v_i,v_j \in V, v_i \neq v_j} G(v_i, v_j)}{\frac{|V|(|V|-1)}{2}}. \tag{3}$$

In the above definition, d is the *euclidean distance function* $(d(v_i, v_j) = |v_i v_j|)$. G is a *Gaussian window function* whose value ranges between $[0, \frac{1}{\sigma\sqrt{2\pi}}]$ and whose value decreases as the distance between nodes v_i and v_j increases. The range of the entropy of a set of nodes is $[0, \infty]$, and its value increases as the pairwise distances of the nodes increases.

3.3 Small World Network Formation

The virtual small-world network is formed cooperatively by the nodes. We call a node that originates a message an *originator*. Since it is desirable that each message chooses a

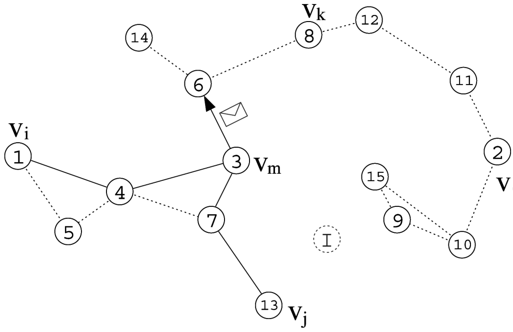


Fig. 1. Virtual small-world network formation. I is the imaginary point, and v_i and v_j are the end points of two established VLLs.

random direction to travel in and explore a different part of the network to establish a new VLL that is different from the established VLLs. We use information of the end points of the established VLLs and an imaginary point to guide the message. The imaginary point is at a distance equal to the maximum transmission distance away from the originator of the message, and the direction of the imaginary point to the originator is chosen randomly.

After the message is created, it travels in the network driven by the entropy. A message is forwarded by a node to a neighbor of the node which has the maximum entropy given the end points of the established VLLs and the virtual point. An example is shown in Fig. 1, where the message is sent from node 3 (v_m) to node 6 to maximize the entropy. The message continues to travel until it has been forwarded $MaxHop$ times or it reaches a local minimum of entropy where no neighbor has a larger entropy. When it is done traveling, the message travels back to the originator from the same path that it traveled on and reports the path as a new VLL.

In this method, the end point of a newly established VLL will be as far from the established VLLs as possible. The imaginary point is different for different messages such that each node could have different VLLs pointing to different directions. Note that the establishment of the first VLL is only affected by its imaginary point while the consequent VLLs are affected by both their imaginary points and the established VLLs.

Fig. 2a shows an example of the established VLLs, where we set $MinHop = 2$ and the number of VLLs to 3. In this example, the VLLs of node v_m in the network are {3, 4, 1}, {3, 7, 13}, and {3, 6, 8} which end at v_i, v_j, and v_k, respectively. We can see in the figure that the above algorithm can generate VLLs that lead to different areas of the network.

3.4 Small World Network Maintenance

It is necessary to set an expiration time (which will be an engineering problem) for each VLL, considering the practical situation where the continuously changing topology of the network may break some of the VLLs. Despite this, as a node periodically discovers new VLLs, the required memory for the available VLLs can be larger than the limited storage in the node. Suppose the expiration time for a VLL is T and the time interval of sending consecutive VLLs discovery messages is T', the maximum number of available VLLs is $N = T/T'$. If, due to limited storage capacity in a node, the maximum number of VLLs that can be stored is M, and $M < N$, a node should discard some VLLs when a new VLL is available and the memory is full.

When a node has $M + 1$ VLLs, our replacement method first lists all the possible combinations of M among the $M + 1$ available VLLs and then calculates the entropy of the end points of each combination of VLLs. The combination of the VLLs with the largest entropy is retained, and the VLL that is not in this combination is removed. That is, we remove some VLLs such that the total pairwise distance of the end points of the remaining VLLs are maximized.

4 Routing in SWING+

In this section, we will present SWING+ in two steps. In the first step (Section 4.1), we present a simple greedy routing protocol in the virtual small-world network which uses the VLLs. In the second step (Section 4.2), we complete our SWING+ protocol which recovers from local minima using an iterative routing scheme.

4.1 Small World Greedy Routing

In the previous section, we presented the method of building a small-world network by establishing VLLs among nodes. In this section, we present a routing algorithm which uses the VLLs in this small-world network to reduce the unnecessary face routing transitions and, thus, improve routing performance.

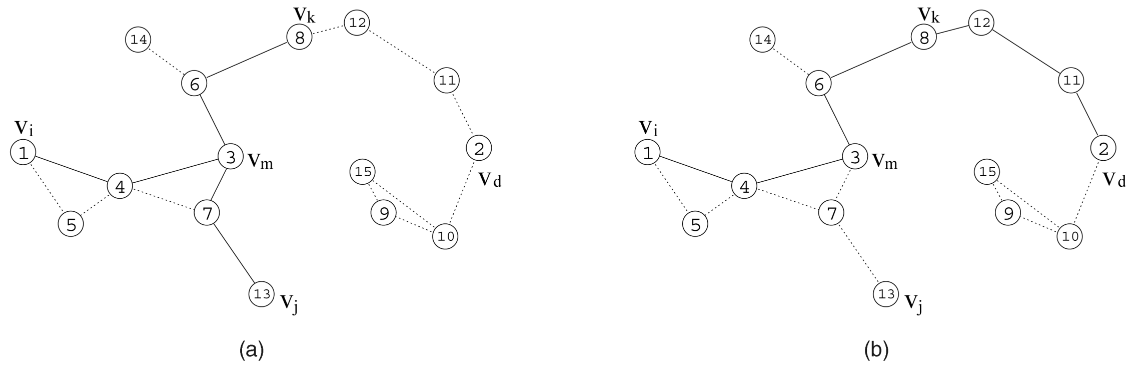


Fig. 2. (a) The VLLs of node v_m. (b) An example of the VF-based greedy protocol using VLLs.

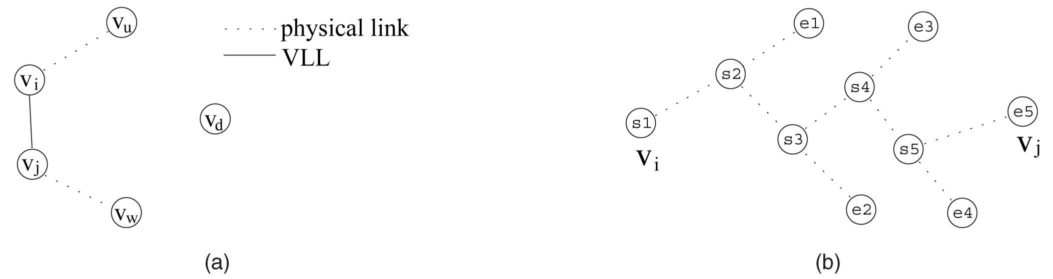


Fig. 3. (a) Loops might occur when a node does not have the information of the link that the message is traveling. (b) Proof of Lemma 1: $force_{dest}(s_1) < force_{dest}(e_1) < force_{dest}(e_2) < \cdots > force_{dest}(e_5) < \cdots$.

Here, we use the concept of VF in our routing algorithm. There is an attractive VF from the source of a message which is denoted as $force_{dest}$:

$$force_{dest}(v, v_d) = d_{max} - d(v, v_d).$$

Here, $d(v, v_d)$ is the distance between the current node v and the destination v_d, and d_{max} is a constant which can be set to arbitrary value. We assume that the maximum transmission range is normalized to 1, and the distance $d(u, v) \leq 1$ for any adjacent nodes u and v. In practice, we can multiply the force by a proportionality constant to take care of proper units. The force function ensures that the force increases as the message gets closer to its destination. We select d_{max} to be the maximum possible distance between two nodes in the network such that $force_{dest}(v, v_d)$ is positive.

Extending the traditional greedy protocol to our small-world greedy routing, which uses VLLs and VF, is straightforward. First, we define a general set of links of a node to be the set of links of the node containing both the physical links to its neighbors and its VLLs. Thereafter, we use links to refer to the links in this general set of links without causing confusion. We define the force of a VLL as the maximum force of the nodes in the VLL. Letting l be a VLL

$$force_{dest}(l, v_d) = \max_{v \in l} force_{dest}(v, v_d).$$

Our small world greedy protocol with VLLs is shown below as Algorithm 1, in which v_d is the destination of the message and v_i is the current node.

Algorithm 1: VF-based greedy protocol using VLLs

```
while the current node $v_i$ is not $v_d$ and $v_i$ is not a local minimum do
    List the links which contain the physical links to all neighbors and all VLLs.
    Calculate the virtual force (VF) on these links.
    Send the message to the next node along the link with the largest VF.
end while
```

Note that Algorithm 1 is a hop-by-hop routing; a node forwards a message to a neighbor that has the maximum force or to a neighbor that is the next hop on a VLL with the maximum force. That is, a message does not need to be forwarded along a selected VLL from the current node to the node on the other end of the VLL, but a new forwarding decision will be made in each node that received a message.

In Algorithm 1, a message may visit a node multiple times. An example of this routing protocol is shown in Fig. 2b, where a message is sent from the source v_i to the destination v_d successfully. While a traditional greedy algorithm will fail on the local minimum v_m, our small-world greedy algorithm succeeds since there is a VLL $\{3, 6, 8\}$ by which a message in v_m knows that v_k is closer (v_k has a larger VF) to v_d than v_m. That is, the local minimum v_m is effectively removed by the VLL.

Since each node makes independent forwarding decisions based on their local VLLs (the views of the links are inconsistent among the nodes), the above protocol might cause loops. For example, in Fig. 3a, suppose there is a VLL, $\{v_i, v_j, \ldots, v_w\}$, in v_i that is the link which has the maximum force, while in v_j, another VLL, $\{v_j, v_i, \ldots, v_u\}$, is the link which has the maximum force, and v_i and v_j have no idea of the maximum force link of each other. When a message travels to v_i or v_j, a loop begins.

A simple solution to this problem, which causes only several bytes of additional transmission overhead, is: a message carries the information of the link (VLL) if it is traveling on one. This information is used by the next node on the link, which receives the message, to make its forwarding decision. We show how this method works using the last example. In Fig. 3a, suppose v_w is closer to v_d than v_u. When a message is forwarded from v_i to v_j, it carries link $\{v_i, v_j, \ldots, v_w\}$ on which it is traveling. Then, the decision of v_j will be that the message continues to be forwarded along link $\{v_j, \ldots, v_w\}$. This is because the carried information indicates that there is a $\{v_j, \ldots, v_w\}$ which is better than the "local" choice $\{v_j, v_i, \ldots, v_u\}$. Thus, the loop is prevented. The following theorem provides a proof of this.

Lemma 1. *If a message heading for destination v_d always carries the information of its current traveling link,*[1] *and the message travels a series of links, $l_1, l_2, \ldots, l_m$, with $s_1, s_2, \ldots, s_m$ and $e_1, e_2, \ldots, e_m$ being their starting nodes and end nodes, respectively, then*

$$force_{dest}(s_1, v_d) < force_{dest}(e_1, v_d) < force_{dest}(e_2, v_d) < \cdots < force_{dest}(e_m, v_d).$$

Proof. As shown in Fig. 3b, obviously, $force_{dest}(s_i, v_d) < force_{dest}(e_i, v_d)$. Note that e_i might not be equal to s_{i+1} (a message might not travel a whole VLL) as mentioned before. Since in each s_i (a node between s_{i-1} and e_{i-1} on l_{i-1}), for all $2 \leq i \leq n$, the message chooses e_i instead of

[1] We refer to a link as either a physical link or a VLL.

e_{i-1}, we have $force_{dest}(e_{i-1}, v_d) < force_{dest}force(e_i, v_d)$. Therefore,

$$force_{dest}(s_1, v_d) < force_{dest}(e_1, v_d) < force_{dest}(e_2, v_d)$$
$$< \cdots < force_{dest}(e_m, v_d).$$

□

Theorem 1. *If each message carries the information of its current traveling link, the VF-based greedy protocol with VLLs is loop free.*[2]

Proof. To prove Theorem 1, we need to prove that a message will not travel to any node v infinite times. Suppose l'_i is the link that has the minimum force when the message travels v for the ith time, and e'_i is the end node of l'_i. According to Lemma 1, we have $force_{dest}(e'_1) < force_{dest}(e'_2) < \cdots$. Since the number of nodes in the network is finite, the number of e'_i is finite. Therefore, the message will not travel to v infinite times. □

4.2 Small World Iterative Navigation Greedy (SWING+)

It is inherited from the family of greedy algorithms that Algorithm 1 may go to a local minimum and fail. The best part of SWING+ is the iterative method that allows the message to continue to travel to the other parts of the network after greedy failures in local minima. In order to recover from local minima, a list V_R is maintained in each message which contains the positions of the encountered local minima that will exert repulsive (negative) VFs to the message. The repulsive force $force_{lm}$ exerted by each local minimum v_{lm} on a current node v is defined by the following equation, where $d(v_{lm}, v) \geq 0$ is the distance between the current node v and the local minimum v_{lm}, and λ and δ ($\lambda \geq \frac{1}{\delta} > 0$) are two parameters in this algorithm:

$$force_{lm}(v_{lm}, v) = \begin{cases} \lambda & \text{if } d(v_{lm}, v) = 0, \\ \frac{1}{\delta} & \text{if } 0 < d(v_{lm}, v) \leq \delta, \\ \frac{1}{d(v_{lm}, v)} & \text{if } d(v_{lm}, v) > \delta. \end{cases} \tag{4}$$

In (4), δ is a small positive constant which prevents the value of the force from increasing to infinity as the node gets closer to the local minimum. If the node is not the local minimum, the maximum repulsive force is $\frac{1}{\delta}$. If the node is the local minimum, the force is λ, and we require that $\lambda > \frac{1}{\delta}$.

Given the set of local minima V_R and the destination v_d, the composition of VFs $force_{comp}$ exerted on a message in a node v and the $force_{comp}$ exerted on a link l are defined in (5) and (6), respectively:

$$force_{comp}(v, v_d, V_R) = force_{dest}(v, v_d) - \sum_{v_{lm} \in V_R} force_{lm}(v_{lm}, v) \tag{5}$$

$$force_{comp}(l, v_d, V_R) = \max_{v \in l} force_{comp}(v, v_d, V_R). \tag{6}$$

Whenever a message fails to be forwarded under the $force_{comp}$ at a local minimum (where the message has a larger $force_{comp}$ than any node neighboring the local minimum), the position of the local minimum is added to V_R. After adding the local minimum to V_R, the $force_{comp}$ in the local minimum is decreased. If after this decrease, the $force_{comp}$ in this local minimum is no longer smaller than all its neighbors, the local minimum is removed. If the local minimum is not removed, the position of the local minimum is allowed to be added to V_R multiple times, until the local minimum is removed. Our SWING+ protocol is listed in Algorithm 2.

Algorithm 2: SWING+

```
while The current node $v_i$ is not $v_d$ do
    while The current node $v_i$ is a local minimum under the $force_{comp}$. do
        Add $v_i$ to list $V_R$.
    end while
    List the links which contain the physical links to all neighbors and all VLLs.
    Calculate the virtual force (VF) on these links.
    Send the message to the next node along the link with the largest VF.
end while
```

4.3 Analysis of SWING+

In the following, we first prove that SWING+ guarantees delivery. Then we derive a bound for the route delivery time for SWING+ which is the first bound for geometric routing protocols in 3D spaces.

We define *iteration* in SWING+ as the process where the message travels from the source or a local minimum to another local minimum. That is, during an iteration, no local minimum is added to the local minimum list V_R of the message. We have the following lemma, which is similar to Theorem 1.

Lemma 2. *If the network size is N, each iteration can be finished in $O(N^2)$ steps.*

Proof. As the message travels in the network, new VLLs are chosen. Each new VLL must contain a node in which the message receives a higher $force_{comp}$ than in any of the nodes in the previous VLLs. Thus, the number of VLLs traveled by a message is $O(N)$. Since the length of each VLL is $O(N)$, the number of hops in each iteration is $O(N^2)$. □

The following lemma shows the relationship between the repulsive forces received by two adjacent nodes in the network.

Lemma 3. *Let f_i and f_j be the repulsive forces exerted from a local minimum v_m to two neighbor nodes v_i and v_j, respectively ($v_i \neq v_j \neq v_m$), and the distance between v_i and v_j is d. We have*

$$\begin{array}{ll} \frac{f_i}{f_j} \geq \frac{\delta}{d+\delta} & \textit{if } d \geq \delta, \\ \frac{f_i}{f_j} \geq \frac{1}{2} & \textit{if } d < \delta. \end{array}$$

[2] Note that temporary loop (v being traveled a finite times) is possible in this algorithm.

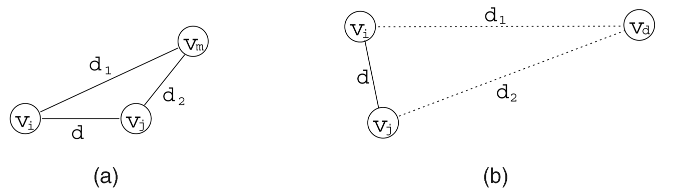


Fig. 4. (a) Proof of Lemma 3. (b) Proof of Lemma 4.

Proof. Without lost of generality, we assume that $f_i \leq f_j$. As shown in Fig. 4a, the distance between v_i and v_m is d_1, and that between v_j and v_m is d_2. We have $d_1 < d + d_2$.

According to (4), if $d \geq \delta$, $f_i = \frac{1}{\max(d_1,\delta)} \geq \frac{1}{\max(d+d_2,\delta)} = \frac{1}{d+d_2}$, and $f_j = \frac{1}{\max(d_2,\delta)}$. Thus, $\frac{f_i}{f_j} \geq \frac{\max(d_2,\delta)}{d+d_2}$, which is minimized when $d_2 = \delta$. Therefore, when $d \geq \delta$, $\frac{f_i}{f_j} \geq \frac{\delta}{d+\delta}$.

If $d < \delta$, $f_i = \frac{1}{\max(d_1,\delta)} \geq \frac{1}{\max(d+d_2,\delta)}$, and $f_j = \frac{1}{\max(d_2,\delta)}$. If $d_2 \geq \delta$, $\frac{f_i}{f_j} = \frac{d_2}{d+d_2} > \frac{1}{2}$. If $d_2 < \delta$, $\frac{f_i}{f_j} = \frac{\delta}{d+\delta} \geq \frac{1}{2}$. That is, if $d < \delta$, $\frac{f_i}{f_j} \geq \frac{1}{2}$. □

A local minimum is added to V_R after each iteration (except the last iteration), and each node can be added to V_R as local minima multiple times. If a message is in a k-times local minimum, it receives a repulsive force which equals $k\lambda$ from this local minimum. The following lemma shows that, when λ in (4) is properly chosen, the times of two adjacent nodes being local minima can differ at most by 1.

Lemma 4. *Let K be the number of local minima the message has encountered, and K_i and K_j be the times that two neighboring nodes v_i and v_j were local minima of the message, respectively. If $\lambda > \frac{K+1}{\delta}$, $|K_i - K_j| \leq 1$.*

Proof. As shown in Fig. 4b, let the distance between v_i and the destination v_d be d_1, that between v_j and v_d be d_2, and that between v_i and v_j be d ($d < 1$ in UDG). The composition VFs for v_i and v_j are given in the following equations (see (6)), where $\bar{f}_i$ and $\bar{f}_j$ are the repulsive forces to v_i and v_j exerted from nodes other than v_i and v_j:

$$force_{comp}(v_i) = (d_{max} - d_1) - K_i\lambda - K_j\frac{1}{d} - (K - K_i - K_j)\bar{f}_i,$$
$$force_{comp}(v_j) = (d_{max} - d_2) - K_j\lambda - K_i\frac{1}{d} - (K - K_i - K_j)\bar{f}_j.$$

First, we assume that $d \geq \delta$. From Lemma 3, we have $\frac{\bar{f}_i}{\bar{f}_j} \geq \frac{\delta}{d+\delta}$, which follows that $\bar{f}_i \geq \frac{\delta}{d+\delta}\bar{f}_j$. We have

$$\begin{aligned} force_{comp}(v_i) \geq\ & (d_{max} - d_1) - K_i\lambda - K_j\frac{1}{d} \\ & - (K - K_i - K_j)\frac{\delta}{d+\delta}\bar{f}_j. \end{aligned}$$

If the message will add v_i as a local minimum one more time to its V_R, i.e., K_i will be increased by one, then it should be satisfied that $force_{comp}(v_i) > force_{comp}(v_j)$. It follows that

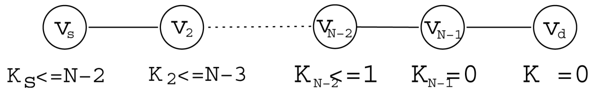


Fig. 5. Proof of Theorem 2.

$$\begin{aligned} &(d_2 - d_1) + (K_j - K_i)\left(\lambda - \frac{1}{d}\right) + (K - K_i - K_j)\left(\bar{f}_j - \frac{\delta}{d+\delta}\bar{f}_j\right) > 0 \\ &\Rightarrow (K_i - K_j)\left(\lambda - \frac{1}{d}\right) < (d_2 - d_1) + (K - K_i - K_j)\left(\bar{f}_j - \frac{\delta}{d+\delta}\bar{f}_j\right) \\ &\quad < d + (K - K_i - K_j)\frac{d}{d+\delta}\bar{f}_j \\ &\Rightarrow (K_i - K_j)\left(\lambda - \frac{1}{\delta}\right) < 1 + (K - K_i - K_j)\frac{d}{d+\delta}\frac{1}{\delta} \\ &\quad < 1 + (K - K_i - K_j)\frac{1}{\delta} \\ &\Rightarrow (K_i - K_j)\lambda < 1 + K\frac{1}{\delta} - (K_i + K_j)\frac{1}{\delta} + (K_i - K_j)\frac{1}{\delta} \\ &\quad = 1 + K\frac{1}{\delta} - K_i\frac{2}{\delta} \\ &\Rightarrow (K_i - K_j) < \frac{\delta + K - 2K_i}{\delta\lambda} < \frac{1+K}{\delta\lambda}. \end{aligned}$$

Since $\lambda > \frac{K+1}{\delta}$, $K_i - K_j < 1$. That is, K_i can only increase when $K_i \leq K_j$. The same rule also applies to K_j. It follows that $|K_i - K_j| \leq 1$.

When $d > \delta$, from Lemma 3, we have $\frac{\bar{f}_i}{\bar{f}_j} \geq \frac{1}{2}$, which follows that $\bar{f}_i \geq \frac{1}{2}\bar{f}_j$, and the result can be proved similar to the situation when $d \geq \delta$. □

Since the number of steps in any iteration is finite (Lemma 2), in order to prove that SWING+ guarantees delivery, we need to prove that the number of local minima for a message is finite.

Theorem 2. *In a network of N nodes, the number of total local minima K visited by a message upon delivery is bounded by $K = O(N^2)$.*

Proof. Let us configure the worst situation where K is maximized. From Lemma 4, the numbers K_i and K_j of any pair of adjacent nodes can differ by at most 1; thus, K is maximized when, for all pairs of adjacent nodes, the times of their being local minima differ by 1, which requires a linear network topology as shown in Fig. 5. In the figure, the destination v_d is at one end of the line. Since the node v_{N-1} next to the destination cannot be a local minimum, $K_{N-1} = 0$. For the node v_{N-2} to the left of v_{N-1}, $K_{N-1} \leq 1$. In the same way, we have $K_2 \leq N - 3$, and $K_s \leq N - 2$. Thus, $K \leq \frac{(N-2+1)(N-2)}{2} < \frac{N^2}{2}$. Therefore, $K = O(N^2)$. □

Theorem 3. *SWING+ guarantees delivery in finite networks.*

Proof. It follows directly from Lemma 2 and Theorem 2. □

Theorem 4. *The route discovery cost of SWING+ is $O(N^4)$ in a network of N nodes.*

Proof. Theorem 2 shows that the number of local minima is $O(N^2)$, which follows that the number of iterations is $O(N^2)$. Lemma 2 shows that the hop-count in each iteration is $O(N^2)$. Therefore, the number of total hop-count is $O(N^4)$. □

Though we cannot derive a tight bound for SWING+, even with a large $\lambda = \frac{K+1}{\delta}$ (as required in the proof of Lemma 4), simulation results show that 1) by using only a small $\lambda = \frac{2}{\delta}$, SWING+ guarantees delivery in 1,000 randomly generated connected networks between totally 1,300,000 randomly selected pairs of nodes, and 2) the average performance of SWING+ is comparable to the state-of-the-art, average-case-optimal, greedy-face combination, GOAFR+ [12].

4.4 Extension in Dense Networks

In our presented VLL establishment scheme where each node maintains a fixed number of VLLs, per node traffic for VLL discovery messages increases as the node density increases. This might consume a large portion of the network bandwidth in dense network areas. Clusterhead selection, such as CDS [5], can be used to reduce the number of nodes that establish loose VLLs, each of which consists of multiple hops of interclusterhead connections. Then, VLL information is shared by the clusterheads within their clusters. This scheme can effectively reduce the excessive VLLs in overpopulated network areas and make the traffic for VLL discovery messages constant in various network densities.

In dense networks, SWING+ routing can be run only on the clusterheads. The source sends the message to its clusterhead. The message is then forwarded using SWING+ among clusterheads until the message reaches a clusterhead neighboring the destination. This method can lower the bound of the algorithm to $O(C^4)$, where C is the size of the clusterheads in the network. In this protocol, none of the nodes, except for the clusterheads, need to maintain VLLs, and all VLLs contain only clusterheads.

4.5 Adaptation to Mobility

VLLs would become obsolete due to mobility, and obsolete VLLs might cause routing loops. For example, as shown in Fig. 3a, node v_i forwards a message to v_j along the best VLL: $\{v_i, v_j, \ldots, v_w\}$. But, this VLL is obsolete because v_j is disconnected with the next node on the VLL. v_j then forwards the message along the second best VLL: $\{v_j, v_i, \ldots, v_u\}$. This forms a routing loop between v_i and v_j. Similarly, we can construct the cases when routing loops appear among n $(n > 2)$ nodes.

To handle the loop between two nodes, we store in the message the previous encountered broken connection between two nodes that is on a VLL that the message traveled. In the previous example, the message will store the broken link between v_j and the next node from v_j to v_w on VLL: $\{v_i, v_j, \ldots, v_w\}$. Each node visited by the message can remove its obsolete VLLs based on the broken connection stored in the message. Routing loops are prevented since a message will not travel an obsolete VLL twice. Note that routing loops among n $(n > 2)$ nodes can be handled by storing the last $n - 1$ broken connections.

Considering the introduced overhead, we only store one broken connection in the message. Delivery fails when a loop appears among n $(n > 2)$. Simulation comparison between protocols that do not use VLLs (loop free) and those that use VLLs show that the delivery rates of the latter are hardly affected since $n > 2$ is very unlikely in practice.

TABLE 1
The Classification of the Simulated Routing Algorithms

Algorithm Name	G	F	CDS	VLL	BD	SB	ING
GREEDY	√						
GREEDY(VLL)	√			√			
SWING+	√						√
SWING+(CDS)	√		√				√
SWING+(VLL)	√			√			√
GFG	√	√				√	
GFG(VLL)	√	√		√		√	
GFG(CDS)	√	√	√			√	
GFG(VLL+CDS)	√	√	√	√		√	
GOAFR	√	√			√	√	
GOAFR(VLL)	√	√		√	√	√	
GOAFR(CDS)	√	√	√		√	√	
GOAFR(VLL+CDS)	√	√	√	√	√	√	
GOAFR+	√	√			√	√	
GOAFR+(VLL)	√	√		√	√	√	
GOAFR+(CDS)	√	√	√		√	√	
GOAFR+(VLL+CDS)	√	√	√	√	√	√	

5 Simulation

To evaluate SWING, we performed simulations in a variety of network settings. We implemented families of algorithms as shown in Table 1 in which each algorithm implements some of the components such that we can observe the improvement of the individual component and their combinations. We also implement a number of geometric algorithms and their extensions based on a number of existing techniques. For each simulation, results are shown and discussed.

5.1 Evaluation Metrics

The metrics that we use to evaluate the protocols are delivery ratio, route establishment time, and route length. Delivery ratio is the ratio of the messages delivered to the destination over the total amount of messages sent. Route establishment time and average route length is counted in terms of hops. There are three reasons for choosing the hop metric for our simulations. First, hop metric is a tradition for ad hoc networks. In most communication standards (such as IEEE 802.11), radio devices transmit with a fixed power which is not adaptive. Second, the networks used as the basis for our simulations are UDGs in which each node has a fixed unit transmission range. Third, it is shown in [12] that the euclidean distance, the link distance, and the energy metrics of a path are equal up to a constant factor on the UDG.

5.2 Simulation Environment and Settings

Simulations were conducted on four protocol families: the Greedy family, the GFG family, the GOAFR family, and the GOAFR+ family. Table 1 shows all of these protocols (in rows) and the algorithms used in each protocol (in columns).

The algorithms used as configurable components in these routing protocols include Greedy algorithm (G), Face algorithm (F), CDS used in face routing, VLL used in greedy routing, bound ellipse/circle (BD) used in the face traversal of GOAFR [13] and GOAFR+ [12], the *sooner back* (SB) algorithm [5] (which makes the message routing in the face mode return sooner to the greedy mode when the current node has a neighbor whose distance to the destination is shorter than that of the last local minimum), and the VF-based *ING method*. It can be proved that a geometric routing algorithm, combining the VF-based greedy mode and the face mode, also guarantees delivery.

We run the simulation on our custom simulator. Our simulations are done in both static connected networks and networks with the random waypoint mobility. In each simulation with a static network, a connected graph with N (ranging from 150 to 450) nodes is randomly generated in a 1,000 × 1,000 square field. One hundred random networks are generated for each of the 13 network densities. We generate connected networks by randomly spraying nodes and discard the networks that are not connected until a connected one is generated. After that, we let the simulator run for a period of time which is sufficient for the nodes to grow the VLLs. We selected this period of time to be 60 seconds. Then, we select 1,000 source-destination pairs by randomly selecting nodes in the network. For each pair of nodes, we generate a message for each protocol. That is, for each routing protocol, 1,300,000 different messages are created in our experiment.

The static network density in our experiment ranges between two extremes. In the sparse extreme, the shortest path is usually much longer than the euclidean distance between the source and the destination, and the topology is more like a tree structure or a linear structure than a mesh. This density is critical for performance evaluation of routing algorithms, where finding a good path at a low cost becomes a nontrivial task and a real challenge for position-based routing. In the dense region, the traditional greedy routing has a high success probability, and therefore, all algorithms have similar performance since they all degrade to greedy routing. The important parameters in our simulation are shown in Table 2.

In the networks following the random waypoint mobility model, the number of nodes is fixed at 250. The pause times of the nodes are selected randomly between 10 and 50 seconds. The nodal mobility speeds of the nodes in different sets of simulations ranges from 5 to 100 m/s in increments of 5 m/s.

5.3 Simulation Results

We display our simulation results in the following hierarchy:

1. We first show the results of the greedy-face combinations which include GFG, GOAFR, and GOAFR+, then we show the results of the SWING+ family which include SWING+ (using VLLs), SWING+(CDS) (without using VLLs but only nodes on CDS forward messages), and SWING+(VLL) (using VLLs).

TABLE 2
Experiment Settings

Parameter	Value
Field size	$1,000 \times 1,000(\mathrm{m})$
Transmission range	100(m)
δ	20(m)
λ	10
d_{max}	$10,000$
Transmission delay	10(ms)
Number of nodes	$150 \sim 450$
Network degree	$4.71 \sim 14.13$
Number of VLLs	$0 \sim 5$
Min length of a VLL	2
Init time for VLLs	60(s)

2. We display results for route establishment time and route length separately, and both of them are based on hop-count. We let each message record the nodes they visit, whose size is the route establishment time. Then, we use Dijkstra's algorithm to get the shortest path in the subgraph of the network consisting only of the visited nodes. The size of the shortest path is the route length.
3. We display results separately for all messages (which we call "total average") and for the messages which the traditional greedy protocol fails to forward (denoted by "when greedy fails" or "NG").
4. We show the delivery rate of all of the protocols under the random waypoint mobility model.

Fig. 6a shows that the number of local minima decreases as the number of VLLs per node in the network increases. In the figure, the number of local minima decreases rapidly before the number of VLLs reaches 3. As analyzed in Section 3.4, the number of available VLLs (and the memory requirement) on each node depends on the VLL message interval and the TTL of the VLLs. In the following experiments, we assume that the establishment/maintenance overhead of three VLLs is not a burden for the network, and that three VLLs do not exceed the memory limit of the nodes.

Fig. 6b shows that the delivery ratio of the pure greedy protocol increases as the number of VLLs increase. Like the previous figure, the first three VLLs are more critical for improving performance than the fourth and the fifth VLLs.

Figs. 7a, 7b, 7c, and 7d are simulation results for GFG, GOAFR, and GOAFR+. Face routing is only run on the CDS by default. We use the best parameter setting for GOAFR, i.e., the major axis of the ellipse is 1.2 times the distance between the local minimum and the destination, and the multiple factor is $\sqrt{2}$ [13]. For GOAFR+, we used the parameters given in [12], i.e., $\sigma = \frac{1}{100}$, $\rho_0 = 1.4$, and $\rho = \sqrt{2}$. It is shown that GOAFR+ has the least route establishment time among the three. As for route length, GOAFR and GOAFR+ have similar performance, which is much better than GFG.

Figs. 8a, 8b, 8c, and 8d are the simulation results for GFG, GOAFR, and GOAFR+ when they use VLLs. These figures

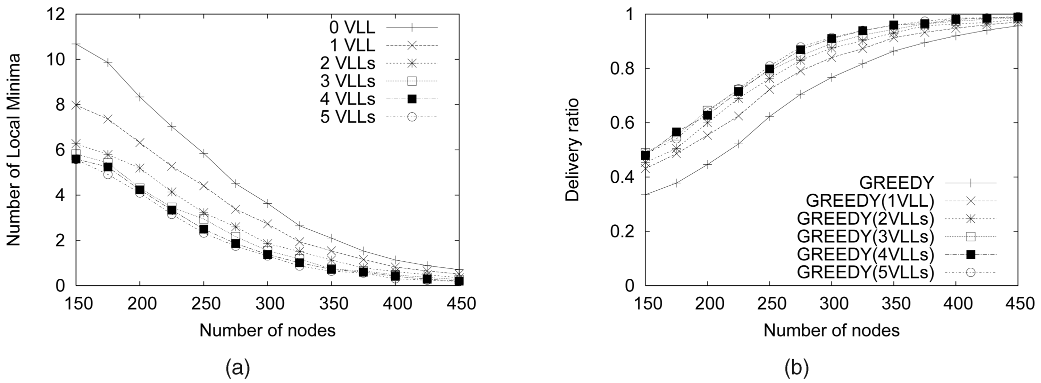


Fig. 6. (a) Number of VLLs versus average number of local minima. (b) Number of VLLs versus delivery ratio in pure greedy.

show similar results as the previous group of figures: GOAFR+ has the least route establishment time among the three; GOAFR and GOAFR+ have similar route lengths, which are much better than that of GFG. Comparing these figures to the previous group of figures, we can see that using VLLs slightly improves the performance of all of the protocols.

Figs. 9a, 9b, 9c, and 9d are the results for SWING+, SWING+(CDS), and SWING+(VLL). In SWING+(CDS), routing is performed on the CDS, and other nodes can only be the source and the destination. The comparison shows that SWING+ has the lowest route establishment time among the three. As for route length, all three protocols are similar. It is shown in Fig. 9c that SWING+ has the smallest route length when the network is dense, but the differences are not big. This is because the resulting routes are not so straight when using VLLs. But considering all cases, SWING+(VLL) is obviously the best in this group.

Figs. 10a and 10b compare the performance of the winners in the previous simulations: SWING+(VLL) and GOAFR+(VLL), where NG denotes the results averaged over the cases when greedy fails. The comparison shows that the two protocols have very close performances, and in most cases, SWING+(VLL) is observed to be slightly better.

Figs. 11a and 11b compare the delivery rate of all protocols in networks with the random waypoint mobility. As shown in the figure, all protocols (except Greedy) have very high delivery rates under mobility. The major reason for the undelivered messages of the protocols other than Greedy is that the network is partitioned due to mobility.

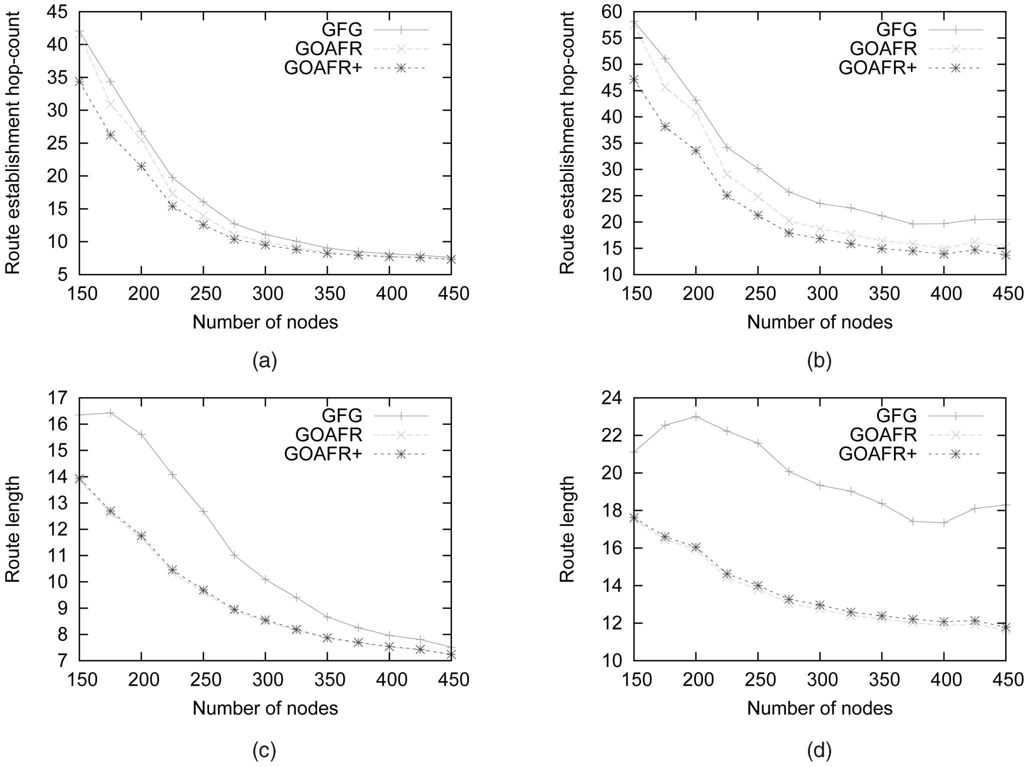
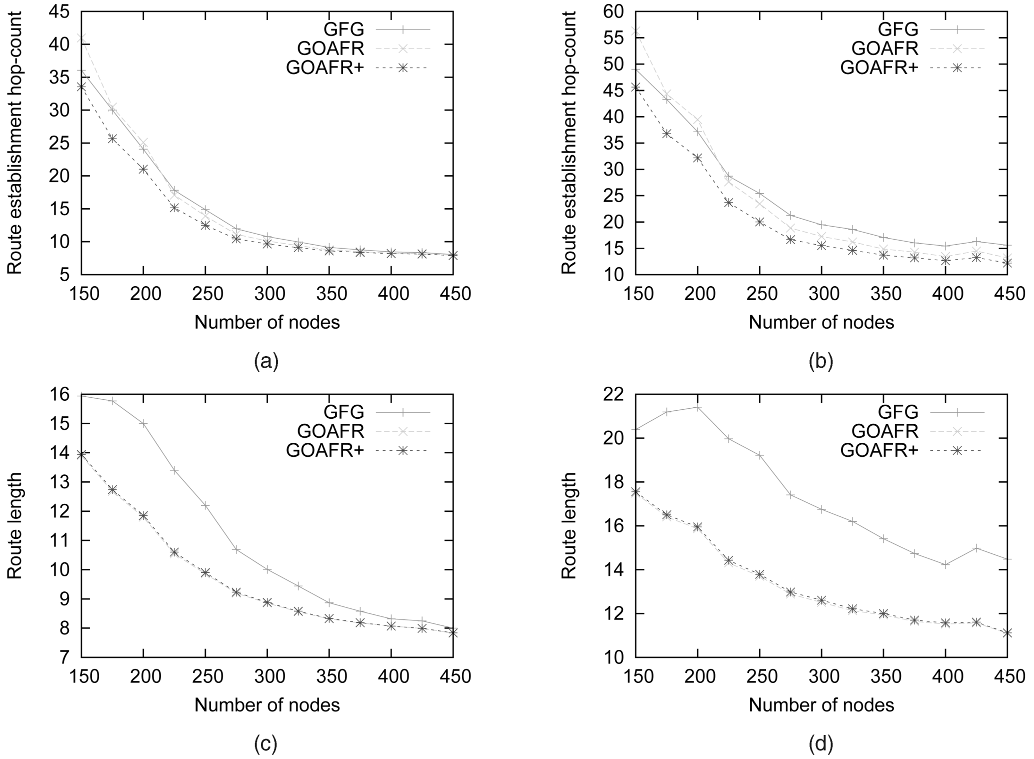
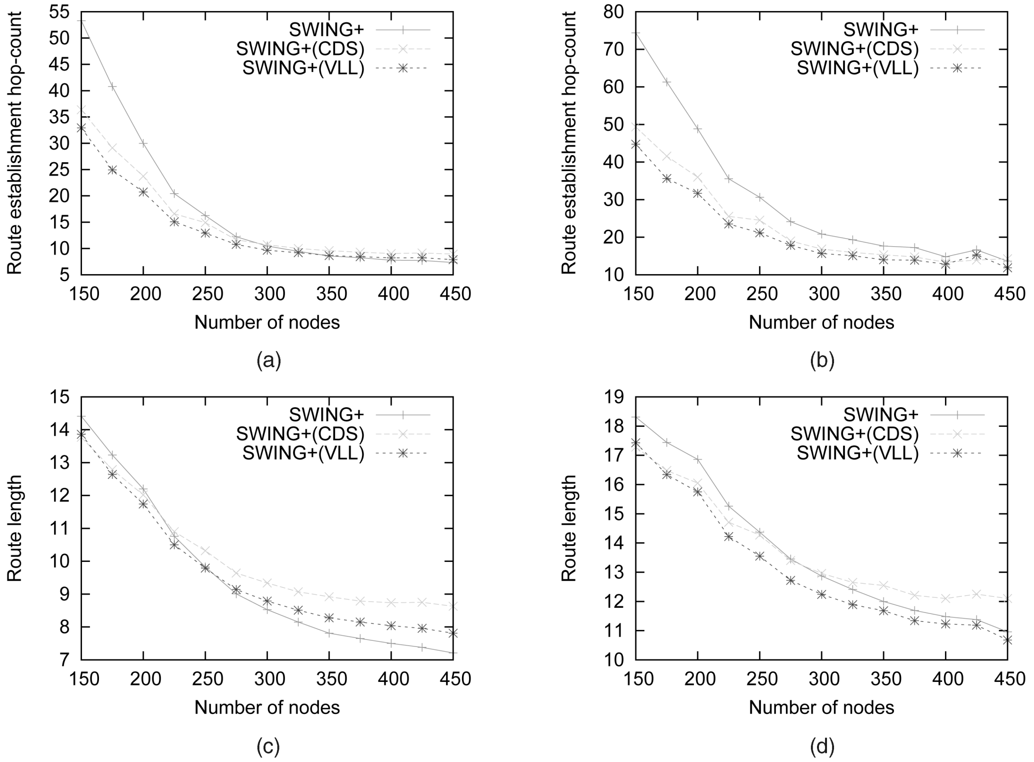


Fig. 7. Comparison of the route establishment time (in terms of hop-count) and route length of GFG, GOAFR, and GOAFR+ without VLLs. (a) Total average. (b) When greedy fails. (c) Total average. (d) When greedy fails.



(a) (b) (c) (d)

Fig. 8. Comparison of the route establishment time (in terms of hop-count) and route length of GFG, GOAFR, and GOAFR+ with VLLs. (a) Total average. (b) When greedy fails. (c) Total average. (d) When greedy fails.

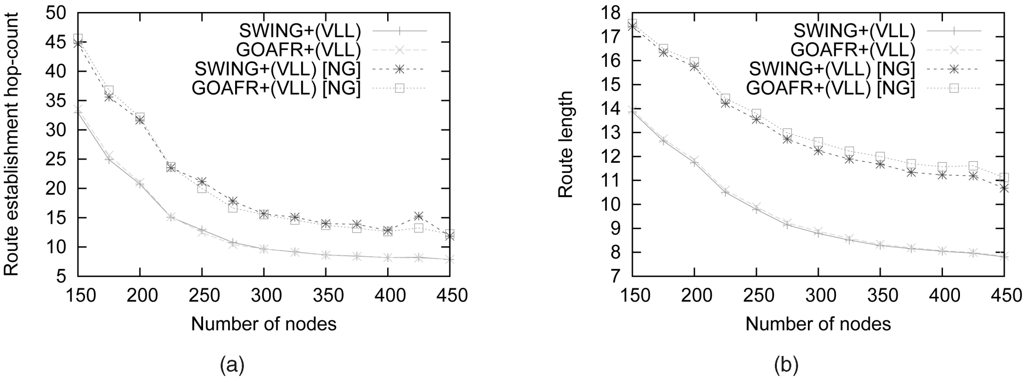


(a) (b) (c) (d)

Fig. 9. Comparison of the route establishment time (in terms of hop-count) and route length of SWING+ without VLLs, SWING+ with CDS and without VLLs, and SWING+. (a) Total average. (b) When greedy fails. (c) Total average. (d) When greedy fails.

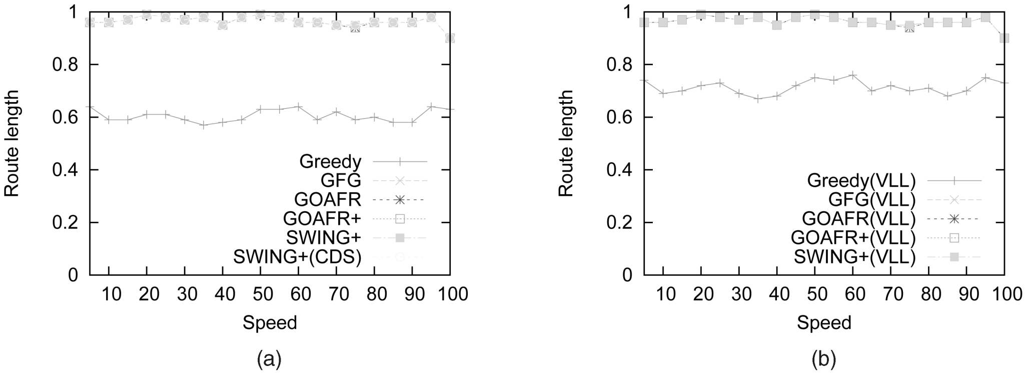
This can be implied from the almost identical delivery rates of all routing protocols (except Greedy).

Comparing Fig. 11a (where protocols do not use VLLs) and Fig. 11b (where protocols use VLLs), we found that 1) VLL improves the delivery rate of Greedy by about 10 percent and 2) the delivery rate of the protocols using VLLs is hardly affected by the level of mobility, which shows that our approach in Section 4.5 is efficient.



(a) (b)

Fig. 10. Comparison of the route establishment hop-count and route length of SWING+ and GOAFR+ in the total average case and in the case when greedy protocol fails.



(a) (b)

Fig. 11. Comparison of the delivery rate in random waypoint mobility model in different nodal mobility speed.

Looking carefully at Figs. 11a and 11b, there are some data points where the greedy-face combinations have smaller delivery rates. We believe that this is caused by mobility: a routing loop would occur if a face changes during the time when the message is traveling on it. This further shows that greedy routing is more robust than face routing.

To summarize the simulation, our new purely greedy position-based routing protocol SWING+ has a comparable (or slightly better) routing performance to the state-of-the-art geometric routing protocol GOAFR+ in terms of route establish delay and route length. The trade-off between SWING+ and GOAFR+ is that GOAFR+ relies on the assumption of using planar graph and UDG, and GOAFR+ has a little more overhead. Simulation results show that our VLL-based protocols are tolerant toward mobility.

6 Overhead and Scalability Analysis

In this section, we analyze the scalability of SWING+ from its communication overhead, computation overhead, and storage overhead. Admittedly, SWING+ introduces additional overhead, but we will see that this additional overhead is only moderate.

Since the length of a VLL is in power-law distribution (see (1)), the expectation of the length of the VLL is $MinHop + 1$. Therefore, the amortized communication overhead for establishing VLL overhead per VLL message interval is $O(MinHop + 1)$. There are two different ways of implementing long links: 1) quick establishment in which long links discovery messages are relayed instantly and the communication overhead is reflected in the number of transmissions and 2) slow establishment (by integrating it in "hello" messages) in which the addition of communication overhead is just a few more bytes in each "hello" message. But the slow establishment is only applicable if the network is relatively static. When the latter mechanism is used, there should be additional error handling for invalid VLLs.

The amortized additional communication overhead of a routing message is the positions of the local minima. We can set a maximum number of iterations, 10 for example, with which most delivery should be guaranteed. We believe that this overhead is acceptable in most applications.

The computation overhead for establishing VLLs is the computation overhead of the entropies as discussed in Section 3.1. Assume that the entropies are calculated whenever the number of VLLs exceeds the memory capacity M. The computation overhead is $O(M^3)$ ($O(M)$ for the number of combinations and $O(M^2)$ for the entropy of each combination).

In SWING+, besides storing the ID and the position of the destination ($O(1)$ memory overhead), a node only needs to store several VLLs. Let M be the number of VLLs that can be stored in each node. The per-node memory overhead is $O(M)$.

The computation for making a forwarding decision includes listing all of the links and calculating the VF of each link given a list V_R. The computation overhead for message forwarding is $O(D|V_R|)$, where D is the average length of the VLLs.

Location service is a common assumption made in geometric routing. The overhead of location service depends on specific algorithms for location service. The overhead of the state-of-the-art location service is $O(\sqrt{d})$, where d is the distance between the source and the destination.

7 Conclusion

This paper has presented a research in position-based routing in manets. This paper removes the constraints in face routing, i.e., planar graph and UDG. We tackle the local minimum problem from two new angles: constructing a virtual small-world network and using a VF-based greedy protocol to recover from local minima. We combined these two methods and propose the purely greedy routing protocol SWING+, which is the first geometric routing protocol applicable in 3D networks. We showed a bound of the route discovery time in SWING+. Extensive simulations were conducted to compare SWING+ with the greedy-face combinations: GFG, GOAFR, and GOAFR+. Simulation results show that SWING+ is comparable to GOAFR+ in terms of route discovery time and route length.

Acknowledgments

This work was supported in part by US National Science Foundation (NSF) Grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

References

[1] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, "Self-Organization in Mobile Ad Hoc Networks: The Approach of Terminodes," *IEEE Comm. Magazine,* pp. 166-175, June 2001.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third ACM Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIALM),* 1999.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *ACM Wireless Networks,* vol. 7, no. 6, pp. 609-616, 2001.

[4] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm.,* special issue on smart spaces and environment, vol. 7, no. 5, pp. 28-34, 2000.

[5] S. Datta, I. Stojmenovic, and J. Wu, "Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," *Cluster Computing,* special issue on mobile ad hoc networks, vol. 5, no. 2, pp. 169-178, Apr. 2002.

[6] K. Gabriel and M. Pearlman, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology,* vol. 18, pp. 259-278, 1969.

[7] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, *Geometric Spanner for Routing in Mobile Networks,* 2001.

[8] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM MobiCom,* 2000.

[9] J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective," *Proc. 32nd ACM Symp. Theory of Computing (STOC),* 2000.

[10] J. Kleinberg, "Small-World Phenomena and the Dynamics of Information," *Proc. Advances in Neural Information Processing Systems (NIPS '01),* vol. 14, 2001.

[11] R. Kleinberg, "Geographic Routing Using Hyperbolic Space," *Proc. IEEE INFOCOM,* 2007.

[12] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric Ad-Hoc Routing: Of Theory and Practice," *Proc. 22nd ACM Symp. Principles of Distributed Computing (PODC),* 2003.

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

[14] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad-Hoc Routing," *Proc. ACM MobiCom,* 2002.

[15] C. Liu and J. Wu, "Position-Based Routing Using Virtual Small World in MANETs," *Proc. 15th Int'l Conf. Computer Comm. and Networks (ICCCN),* 2006.

[16] C. Liu and J. Wu, "SWING: Small World Iterative Navigation Greedy Routing Protocol in MANETs," *Proc. 49th IEEE Globe Telecommunications (GLOBECOM),* 2006.

[17] S. Milgram, "The Small World Problem," *Psychology Today,* vol. 1, no. 61, 1967.

[18] M. Nesterenko and A. Vora, "Void Traversal for Guaranteed Delivery in Geometric Routing," *Proc. Second IEEE Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS),* 2005.

[19] A. Renyi, "On Measures of Entropy and Information," *Proc. Fourth Berkeley Symp. Math. Statistics and Probability,* 1960.

[20] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition,* vol. 12, no. 4, pp. 261-268, 1980.

[21] D. Watts and S. Strogatz, "Collective Dynamics of Small-World Networks," *Nature,* vol. 393, no. 440, 1998.

[22] F. Zhang, H. Li, A. Jiang, J. Chen, and P. Luo, "Face Tracing Based Geographic Routing in Nonplanar Wireless Networks," *Proc. IEEE INFOCOM,* 2007.

Cong Liu received the BS degree in microelectronics from South China University of Technology in 2002 and the MS degree in computer software and theory from Sun Yat-sen (Zhongshan) University, Guangzhou, China, in 2005. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, under the supervision of Dr. Jie Wu. He has served as a reviewer for many conferences and journal papers. His main research interests include routing in mobile ad hoc networks (manets), delay-tolerant networks (DTNs), and transaction processing.

Jie Wu is a distinguished professor in the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, and a program director at the National Science Foundation. His research interests include the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 400 papers in various journals and conference proceedings. He was on the editorial board of the *IEEE Transactions on Parallel and Distributed Systems* and was a coguest editor of the *IEEE Computer* and *Journal of Parallel and Distributed Computing*. He served as the program cochair for MASS '04 and the program vice chair for ICDCS '08 and ICPP '00. He was also the general chair for MASS '06 and is the general chair for IPDPS '08. He is the author of the text "Distributed System Design" published by the CRC Press. He was also the recipient of the 1996-1997, 2001-2002, and 2006-2007 Researcher of the Year Award at Florida Atlantic University. He has served as an IEEE Computer Society distinguished visitor and is the chairman of the IEEE Technical Committee on Distributed Processing (TCDP). He is a fellow of the IEEE.