Stateless Geocasting

Jordan Adamek, Mikhail Nesterenko, Sébastien Tixeuil

Stateless Geocasting

Jordan Adamek[1], Mikhail Nesterenko[1], and Sébastien Tixeuil[2⋆]

[1] Kent State University
[2] UPMC Sorbonne Universités & IUF

Abstract. We present two stateless algorithms that guarantee to deliver the message to every device in a designated geographic area: flooding and planar geocasting. Due to the algorithms' statelessness, intermediate devices do not have to keep message data between message transmissions. We formally prove the algorithms correct, estimate their message complexity and evaluate their performance through simulation.

1 Introduction

The advent of ubiquitous wireless networks, from sensor networks tracking environmental patterns to metropolitan areas offering free wireless Internet services to residents, upends classical means of routing and delivering information. The scale, volatility and dynamic nature of these networks present a formidable challenge.

One of the simplest routing algorithms for wireless networks is *controlled flooding* where each device retransmits the message to all its neighbors. The device needs to store the information about the transmission to prevent duplicate message resents. Flooding potentially involves all communication devices of the network and, therefore, does not scale well. Early routing algorithms are typically routing-table based [7, 15]. However, maintenance of these tables is resource intensive and is often infeasible. Geometric routing offers more scalable and resource frugal solutions to wireless navigation. In *geometric routing*, message forwarding decisions are based on communication device coordinates. These may be physical coordinates obtained, for example, from GPS, or virtual coordinates computed by devices themselves [6, 9–11, 14]. Routing may be unicast, where message is to be delivered to a single target device, or multicast where there are several targets.

Geometric routing allows *stateless* communication where devices do not store any information about the transmitted message between transmissions. This is a particularly attractive property: it scales well since no multihop routing information need to be maintained by the communicating devices; it is energy efficient since resources are not spent on topology updates; and configuration change- and fault- tolerant as the system trivially adjusts to them. A number of unicast stateless geometric routing algorithms is presented in the literature [1, 3, 4, 8, 10–12].

Geocasting is a particular variant of multicasting where the source device wishes to send the information to all devices located in a specific geographic region. For example, geocasting may be used to notify all households in a flood-risk area once the water level reaches some critical point; or to locate a moving vehicle whose last known coordinates are available at the source. Geometric routing may be particularly suitable for geocasting.

⋆ This work was supported in part by LINCS.

Related work. Let us cover unicast geometric routing first. The simplest form of geometric routing is greedy. In *greedy routing* each device selects the next hop neighbor with the closest Euclidean distance to the target. However, greedy routing fails if some device is the closest to the destination in its immediate neighborhood. That is, this device is a *local minimum*. Face routing guarantees delivery by navigating around faces of a planarized communication graph [1]. Face routing may be inefficient if traversed faces are large. Greedy-Face-Greedy [4] starts in greedy mode and switches to face routing only in case greedy fails. Once recovered, it switches back to greedy. Face traversal may be inefficient if its traversal direction is selected inopportunely: face traversal distance may be long in one direction and short in the other. GOAFR+[12] finds the shorter traversal direction by switching it once the message reaches a pre-determined ellipse containing source and target devices. Concurrent Face Routing [3] optimizes the speed of message delivery by sending two concurrent messages in the opposite traversal directions.

Let us now discuss existing geometric geocasting algorithms. Geographic-Forwarding-Geocast [16] starts as a geometric unicast until it reaches the geocast region. Once inside the region, the message is flooded. The flood messages that reach devices outside the geocast region are discarded. The flooding is stateful. Moreover, as noted by Casteigts et al. [2], Geographic-Forwarding-Geocast may fail to deliver the message to all devices in the geocast region, if the region is disconnected and the only connectivity is through outside devices. Virtual Surrounding Face [13] avoids this problem by pre-computing in advance all planar faces that intersect the geocast region. The algorithm unicasts to the region and, upon reaching the geocast area, floods it inside and traverses all the precomputed surrounding faces on the outside. The algorithm is stateful. Also, the pre-computation and maintenance of the virtual surrounding face is by nature stateful and may incur significant overhead in a dynamic wireless network. Bose et al. [1] propose GFG-based depth-first face exploration to implement geocasting. In principle, this solution is stateless, however Casteigts et al. [2] point out that it requires considerable message overhead and the only way to mitigate this overhead is to pre-process the network topology to give devices additional information. This preprocessing is stateful and may require significant communication resources.

Thus, existing geometric geocasting algorithms are either stateful or so inefficient that their statelessness is ineffectual.

Our contribution. We present new stateless geocasting routing algorithms. We describe a stateless controlled flooding algorithm, SF, which obviates the need for a locally stored information to prevent multiple retransmissions. This algorithm is of independent interest, as it allows to render existing work based on controlled flooding [1, 13, 16] stateless as well. Then, we present a stateless concurrent geometric routing algorithm, SPG, with better scalability and message overhead than SF. We explore combinations of these algorithms and greedy routing. We formally prove the algorithms correct, analyze their message complexity and evaluate their performance through simulation. From our analysis, it follows that the

presented algorithms are message efficient, provide guaranteed delivery to the geocast region with low latency, and do not rely on computation intensive preprocessing.

2 Notation and Definitions

Wireless network, message transmission, routing algorithms. A *wireless network* is a set of computer communication devices capable of exchanging messages. The network is represented as a graph $G = (V, E)$, where V is a set of devices, and E is a set of edges that connect them. An edge exists between two devices if they can send messages directly. Two such devices are called *neighbors*. The graph is *fixed maximum degree* if there is constant k, independent of network parameters such that each device has at most k neighbors. The communication is two-directional and the graph is undirected. A network is *connected* if there exists a path between any two devices.

Every device has unique planar coordinates which *embeds* the graph into the geometric plane. A *dominating set* is a subset of V where every device in V is a neighbor of at least one device in this subset. A *connected dominating set* induces a subgraph that is connected.

A *routing algorithm* ensures that a message is delivered from the *source* device to a *target* device. If the source and the target are not neighbors, the routing algorithm is executed on intermediate devices to decide as how to route the messages to targets.

To help with routing, a message carries routing information. We consider routing algorithms where the amount of information the message carries is independent of the network size. That is, we are interested in *constant message size* routing algorithms. This, for example, precludes a routing algorithm from requesting the message to carry a complete traveled route. Each message carries two addresses: the (immediate) *sender*, i.e. the device transmitting the message, and the (immediate) *receiver*, i.e. the device the message is being sent to.

Computations and fairness. A *step* in a routing algorithm is the receipt of a message by the receiver device and local processing of the message according to the algorithm, which may result in further messages added to the send queue of the device. A step is *atomic* if it does not overlap with steps at this or other devices.

Every device has a *send queue* SQ that collects messages to be sent. A message is transmitted by taking it from the sender's send queue and transferring it to the receiver where it is processed according to the routing algorithm in a single atomic step.

Computation is a sequence of atomic steps that starts in an initial state of the algorithm. A computation is *fair* if every message that is in a send queue SQ of some device is eventually either transmitted or removed from this queue during this computation. That is, a message may not "get stuck" in a send queue forever. To reason about a routing algorithm, we consider its fair computations. A computation is *finite* if it has a finite number of steps. A routing algorithm is *terminating* if all its computations are finite. A terminating routing algorithm never leaves messages indefinitely circulating in the network.

Statelessness. A routing algorithm is *stateless* if it is designed such that devices store no information about messages between transmissions. It is *stateful* otherwise.

Flooding. One of the simplest routing algorithms is flooding. In *flooding*, the source device sends a message to all its neighbors. When a device receives this message, it subsequently sends the message to all its own neighbors. This simple algorithm guarantees delivery to all devices connected to the source.

If a message is flooded, it may travel over multiple paths. Thus, a single device may receive the same messages multiple times. To avoid endless retransmission of messages, flooding must have a mechanism of eliminating duplicates. In classic flooding, each device maintains a flag for each transmitted message. If the message is already transmitted, and it is received again, the duplicate is discarded. That is, classic flooding is stateful. In this paper, we present a new stateless flooding algorithm.

Planarity, face traversal, mates. Simple flooding requires all devices in the network to transmit the message. This may not be efficient. Graph planarization offers a way to design more efficient algorithms. A graph embedding is *planar* if graph edges intersect only at vertices. A *connected planar subgraph* is a subset of vertices and their induced edges such that the resultant graph is planar and connected. In general, finding a planar subgraph is a complex task. However, for certain graph classes it is relatively simple.

A graph is *unit-disk* if a pair of vertices a and b are neighbors if and only if the distance between them is no more than 1. Such graph approximates a wireless network. In such a graph, a connected planar subgraph may be found by local computation at every device using Relative Neighborhood or Gabriel Graph [1, 5, 8, 17]. Moreover, a local computation on a unit-disk graph may yield a fixed maximum degree connected dominating set subgraph [18]. In our message complexity estimations and in our simulation, we consider the original graph to be unit-disk.

Face is a region on the plane such that, for any two points in the region, there is a continuous line that connects them without intersecting graph edges. Note, for example, faces F and G in Figure 3. A planar embedding of a finite graph divides the plane into a finite set of faces. There areas of each face but one are finite. They are *internal* faces. One face is an infinite *external* face.

In a planar graph, messages are routed by traversing such faces using right- or left-hand-rule. In the *right-hand-rule*, if device a receives a message from device b, device a forwards the message to device c that is nearest to b clockwise. In the *left-hand-rule*, the message is forwarded to the nearest counter-clockwise neighbor. Two messages are *mates* if the sender of each message is the receiver of the other. For planar traversal algorithms, mates also must have the opposite traversal direction: right- or left-hand-rule.

Geocasting. The problem of *geocasting* is communicating a message from a source device to all devices located in a designated *geocast region*. In other words, every device in the geocast

region is a target. The geocast region is often a circle or rectangle. Note that the source itself may be in the geocast region. The problem is complicated by the fact that devices in the geocast region may only be connected through the outside device. See, for example, devices f and i in Figure 3. Thus, message delivery to all devices in the geocast region requires exploring these outside connecting paths.

In this paper, we present a stateless geocasting algorithm and its combination with stateless flooding.

3 Algorithm Descriptions

```
device s
foreach n ∈ N do
    add M(s, n) to SQ

device n
if receive M(a, n) then
    if M(n, a) ∈ SQ then
        /* found mate */
        discard M(n, a) from SQ
    else
        foreach m ∈ N : m ≠ a do
            add M(n, m) to SQ
```

Fig. 1. SF pseudocode.

```
device s
/* let F be a face bordering s
    and intersecting the sr-line */
add L(s, d, F) to SQ
add R(s, d, F) to SQ

device n
if receive L(s, d, F) then
    if R(s, d, F) ∈ SQ then
        /* found mate */
        discard R(s, d, F) from SQ
    else
        if F is a juncture then
            foreach F' ≠ F that is
                    a juncture do
                add L(s, d, F') to SQ
                add R(s, d, F') to SQ
        add L(s, d, F) to SQ
if receive R(s, d, F) then
    /* handle similar to L(s, d, F) */
```

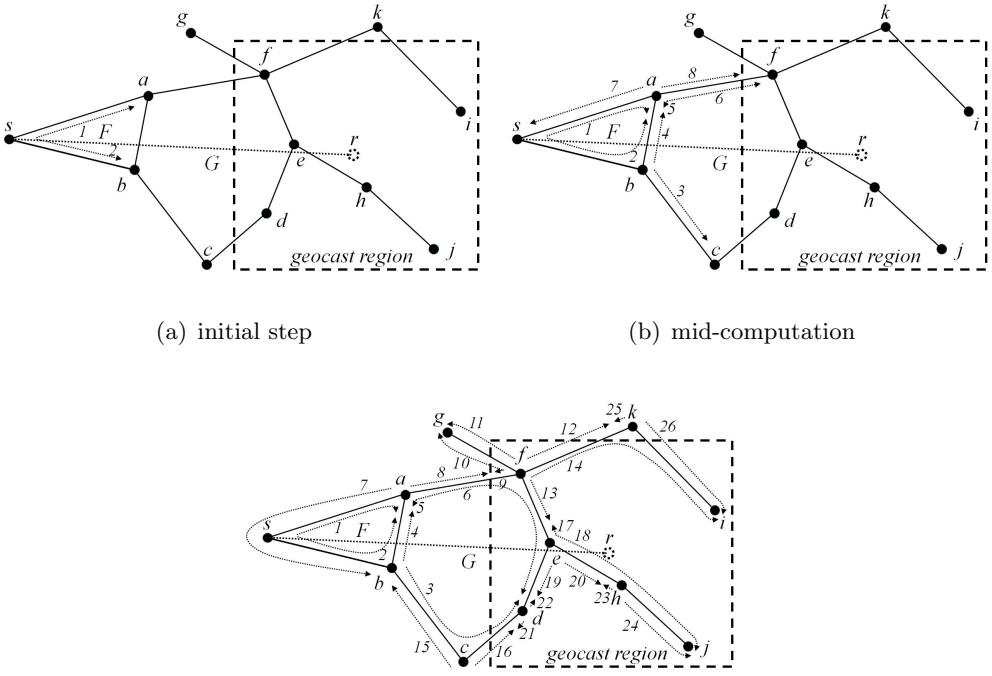
Fig. 2. SPG pseudocode.

SF. The pseudcode for *stateless flooding* (SF) routing algorithm is shown in Figure 1. The algorithm is as follows. The source device adds a message $M(sender, receiver)$ to its send queue SQ to be sent to all devices in its neighbor set N. When a device receives a message from neighbor a, it first checks its send queue for a mate. If a mate exists, both messages are discarded. Otherwise, the device sends the message to all neighbors except a.

SPG. *Stateless planar geocasting* (SPG) algorithm uses face traversal to limit the number of messages sent during mobile geocasting. Let us start the algorithm description with a couple of definitions. Every message carries the coordinates of the source and the parameters of the geocast region so it can compute the *sr-line* — the source-region line that connects the source device with the center of the geocast region. A device is a *juncture* if it is incident to an edge which intersects the sr-line or the geocast region. Note that all devices inside the

geocast region are junctures. A juncture, as any device, is adjacent to a number of faces. The number of these faces is equal to the number of the neighbors. For example, in Figure 3, device f is adjacent to faces afe, efk, kfg and gfa. Note that some of these local faces may globally be the same face. For example, afg, gfk and kfe are, in fact, the external face. When receiving a message traversing a face, a juncture device may determine which face the message is traversing if the message carries its sender and its traversal direction: right- or left-hand-rule. For example, if f receives a right-hand-rule message from a, then f is able to determine that the message traverses face afe. To simplify the presentation of SPG, we just refer to a particular face that the message is traversing. In SPG, messages carry the source coordinates s, region coordinates r and the message traversal direction R or L.

The pseudocode for SPG is shown in Figure 2. The SPG algorithm operates as follows. The source device sends two messages in the opposite traversal directions along the face that intersects the sr-line. When a device receives a message, it checks its send queue for a mate. If a mate is present, both messages are discarded. Otherwise, the device forwards the message along its current face. If the device is a juncture and the face intersects the sr-line or geocast region, then the device *splits* the message by sending a pair of messages in every other face that intersects the sr-line/geocast region.



(a) initial step

(b) mid-computation

(c) completed computation.

Fig. 3. Example computation of SPG.

We illustrate the operation of SPG with an example shown in Figure 3. Device s geocasts a message by sending a left-hand-rule message 1 to device a and right-hand-rule message 2 to device b. See Figure 3(a). This initiates the traversal of face F. Both a and b are junctures. Device a has adjacent edges that intersect sr-line and the geocast region. Device b has an adjacent edge intersecting sr-line. Once 2 reaches b, it forwards it to a and splits it by sending messages 3 and 4 in face G. See Figure 3(b). Note that face sbc does not intersect sr-line or the geocast region so no messages are sent there. When 1 reaches a, it forwards it to b by adding it to its send queue. Device a also splits 1 by sending 5, 6, 7 and 8. Once 2 is received by a, it meets its mate in SQ and both messages are discarded completing the traversal of face F. This computation continues until all messages are delivered to targets. The result of the complete computation is shown in Figure 3(c).

SF+SPG. For routing, pure SPG uses the planar subgraph. However, this eliminates the non-planar edges that might be effective in message transmission. This elimination is unavoidable outside the geocast region to guarantee delivery to all the targets. However, inside the geocast region, SPG may be supplemented by stateless flooding.

Combined algorithm, SF+SPG, uses SPG to route toward and around the geocast region, and SF to flood inside the region. Each message carries a mode: flood or planar, and is routed using SF or SPG, respectively.

Devices outside the geocast region receive and send messages only in planar mode. When a device inside the geocast region receives a message from neighbor b, it sends a single flood message to all neighbors inside the region, and a pair of planar messages with opposite traversal directions to all neighbors outside the region, except b. If the received message was in planar mode, it is sent back to b, and discarded otherwise.

SF+SPG+G. Algorithm SF+SPG may be further combined with greedy routing to decrease the number of required message transmissions. Rather than start SPG at the source device, the message may be initially transmitted using greedy routing by transmitting a single message to the neighbor that is closest to the center of the geocast region. The algorithm switches to SPG only when the greedy routing encounters local minimum: a device with no neighbors closer to the geocast region; or when the greedy message actually reaches the geocast region.

4 Correctness Proofs and Efficiency Bounds

Correctness proofs. We focus on SF first. Let us introduce notation we use for the proofs. A device is *visited* if it receives the message at least once. An edge is *used* if the message was sent over it at least once. It is *unused* otherwise. A visited device is a *border* if it has an adjacent unused link. A visited device that is not a border is *internal*.

Lemma 1. *In SF, every border device holds a message in SQ to be sent over every unused link and it never holds a message to be send over a used link.*

Proof: We prove the lemma by induction. The source sends messages over the links to its neighbors. Therefore, right before the transmission, the source is a border device with every link unused and a message to transmit over this link. Therefore, the conditions of the lemma hold. Assume the conditions hold at some step of a computation. Let us consider the next step: a transmission of the message from device a to device b. Device b may be visited or not visited. If b is not visited, then all its links, except for link to a, are unused. When b receives a message from a, it becomes a border device and it sends messages to all neighbors except a. This satisfies the conditions of the lemma. If b is already visited, then it has a message to be sent to a. This massage is a mate of the message received by b from a. By the algorithm, these two messages are discarded. That is, once the message is transmitted to a visited device and uses the channel, there are no messages to be sent over this used channel. Again, the conditions of the lemma hold. □

Theorem 1. *SF guarantees termination and delivery from the source to all target devices connected to the source.*

Proof: Once the source device has a message to send, it sends to all its neighbors. That is, it becomes a border device. According to Lemma 1, every border device has a message to transmit over unused channels. Since we consider fair computations of routing algorithms, this message is eventually going to be transmitted. If the receiver device is not visited, it becomes visited and sends messages to all its neighbors. Eventually, all devices connected to the source will be visited, and all channels used. That is, SF delivers the message to all devices connected to the source. Note that according to Lemma 1, once the channel is used, there are no messages to be sent across it. That is, SF terminates. □

We now prove correctness of SPG. Let us introduce additional terminology. A device is *segment-visited* if it was visited during the traversal of this face. A *visited segment* of a face is a sequence of neighbor devices that have been segment-visited. A *segment-border* of a visited segment is a segment-visited device that has an edge adjacent to this face that has not been used. Note that an edge is adjacent to two faces. Thus for SPG, an edge may be used in one face and not used in the other. A visited device that is not a border is *segment-internal*. Two faces are *adjacent* if they share a common juncture device, and are *juncture connected* if there exists a sequence of adjacent faces from one to the other.

Lemma 2. *In SPG, every segment-border device always holds a message to be sent over unused adjacent edge. An internal device never holds such message.*

Proof: The proof is by induction on the devices of the face having the visited segment. A visited segment is created when a juncture device is visited. This juncture may be the source device s or another juncture splitting the message when it is visited in an adjacent face. Once the visited segment is created, it contains a single border device with two messages sent in the opposite traversal directions. This is our base case. Let us consider a computation of SPG where every segment of every face is as stated in the conditions of the lemma. Let us focus on a particular face F and a message transmission affecting its segments.

First, let us consider a message transmission by a device adjacent to F. It may only be a border device. The message recipient may be a non-visited device or a border of another visited segment. If the recipient is a non-visited device, once the message is received, the recipient forwards the message to its neighbor. That is, the recipient becomes a new border device with the sent message while the sender becomes an internal device without a message. Thus, the conditions of the lemma are satisfied. If the recipient is a border device of an adjacent segment, by the induction hypothesis, the recipient holds a mate to be transmitted to the original sender. The two messages are discarded and the two adjacent segments merge preserving the conditions of the lemma.

Let us now contemplate a message transmission by the device that is not adjacent to F. The only way it may affect F is if the transmission is to a juncture of F in an adjacent face. However, by the design of the algorithm, the juncture is instantly visited in every adjacent face. Hence, the message transmission should encounter a border device with a mate and be eliminated.

That is, regardless of the kind of message transmissions we consider, the conditions of the lemma are preserved. □

Lemma 3. *In SPG, if a face has a visited segment, every device adjacent to this face is eventually visited.*

Proof: If a face contains a non-visited device, then at least one non-visited device is adjacent to a border device of a visited segment. According to Lemma 2, this border device has a message to be sent to the non-visited adjacent device. Since we only consider fair computations of routing algorithms, this message is eventually transmitted. Once the message is transmitted, the adjacent device becomes visited. This process continues until all devices are visited. □

Lemma 4. *In SPG, every device in a face connected to the source device face is eventually visited.*

Proof: We start with the face that contains the source device s. The source device is in a visited segment. According to Lemma 3, every device in this face is eventually visited. By the design of the algorithm, a juncture device is instantaneously visited in all its adjacent faces. This means that visiting every device in the face that contains s creates visited segments in every face that is adjacent to it. Repeated application of Lemma 3 proves this lemma. □

Proposition 1. *In a planar graph, if a target device is connected to the source device, then this target device lies on a face connected to the source device face.*

The below theorem follows from Proposition 1 and Lemma 4.

Theorem 2. *SPG guarantees termination and delivery of a message from the source to all target devices connected to the source.*

Message costs. In the case of stateless flooding (SF), each device sends exactly one message. That is, the total number of messages is E. In case of stateless planar geocasting (SPG), a

message may be sent along each face. An edge may be adjacent to two faces. Hence, SPG may send $2E$ messages. This estimate is tight. See, for example, Figure 4 where SF sends E messages while SPG sends $2E$. That is, in the worst case, SPG may be twice as costly as SF.

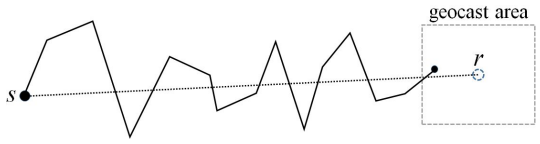


Fig. 4. Example graph for $2E$ message cost in SPG.

However, for most graphs, SPG is significantly more efficient. To give a more realistic message cost estimate for SPG, we make several assumptions about the network graphs. First, we assume that the geocast region is square.

The graph is *face smooth* if there are two constants c_1 and c_2 that are independent of network parameters such that (i) for each face $\rho^2 < c_1 a$ where ρ is the perimeter of the face, and a its area, and (ii) for any two points in the graph, $a_s < c_2 \frac{\pi d^2}{4}$ where a_s is the area of all internal faces that intersect the line between these two points and d is the Euclidean distance between them. For an internal face, the area computation is straightforward; for the external face, an area of an arbitrary figure enclosing the graph, for example convex hull, is considered. The first assumption places limits on how "ragged" the perimeter of the face may be, while the second limits how "uneven" the faces may be in size by assuming that the area of all intersecting faces is included in a certain disk whose diameter is related to the distance between two devices.

Lemma 5. *For face smooth graphs, the message cost of SPG+SF is less than*

$$d\frac{k}{2}\sqrt{\pi c_1 c_2} + k\sqrt{c_1 A} + 2kG + 2k\sqrt{\pi c_1 c_2 G},$$

where d the length of sr-line, A and G are the respective areas of the whole graph and the geocast region, k is the maximum degree and c_1 and c_2 are constants independent of the graph parameters.

Proof: We arrive at the estimate by bounding the number of messages it takes to carry out individual parts the algorithm: to traverse the faces (both internal and external) intersecting sr-line, to flood the geocast region, and to traverse the faces that intersect it. Note that due to the design of the algorithm, the external face is traversed only once. Hence, we account for the messages it takes to traverse it only once as well.

For the sr-line, combining the two assumptions for the face smooth graphs, we get:

$$\rho_s^2 < c_1 c_2 \frac{\pi d^2}{4},$$

where ρ_s is the sum of all perimeters of internal faces intersecting the sr-line. If k is the maximum degree of the graph, the maximum number of devices lying along the perimeter ρ_s is $k\rho_s$. Hence, the number m_{sr} of messages sent traversing these faces is:

$$\left(\frac{m_{sr}}{k}\right)^2 < c_1 c_2 \frac{\pi d^2}{4}$$

That is,

$$m_{sr} < d\frac{k}{2}\sqrt{\pi c_1 c_2}$$

Similarly, if SPG+SF intersects the external face, the number of messages m_e it takes to traverse it can be bounded as follows:

$$m_e < k\sqrt{c_1 A}$$

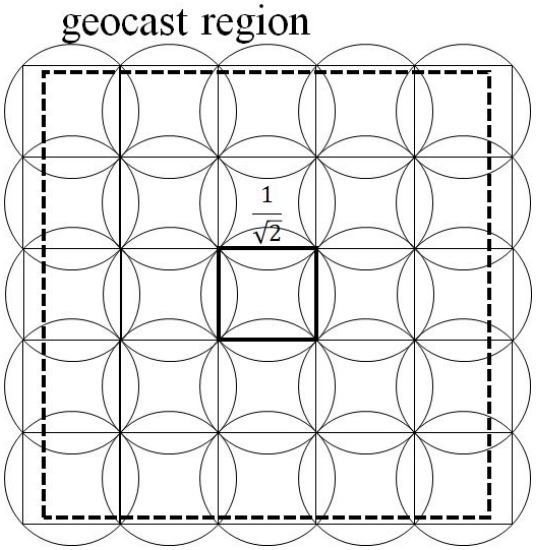


Fig. 5. Estimating the number of devices in the geocast region.

Let us estimate the number of messages expended to deliver to all the devices in the geocast region G. We first bound the number of messages sent by the flooding part of the algorithm. We completely cover the whole geocast region with unit-disks as shown in Figure 5. In this arrangement, each unit disk covers a square with side length of $\frac{1}{\sqrt{2}}$. With k being the maximum device degree, the number of devices in the geocast region, and therefore, the number of messages m_f it takes to flood it, is:

$$m_f < k\frac{G}{\left(\frac{1}{\sqrt{2}}\right)^2} = 2kG$$

Let us now estimate the number of messages sent across faces intersecting the geocast region. Since the area is square, the length of its side is $\sqrt{G}$. By the second assumption of the face smooth graph, the area of all internal faces that intersect this side is less than $c_2\frac{\pi G}{4}$. By the first assumption, the perimeters of these faces is less than $\sqrt{c_1 c_2 \frac{\pi G}{4}}$. Taking into account that the geocast region has four sides and that the maximum device degree is k, for the bound of the number of messages m_f on the faces that intersect the geocast region is as follows:

$$m_g < 4k\sqrt{c_1 c_2 \frac{\pi G}{4}} = 2k\sqrt{\pi c_1 c_2 G}$$

Adding the bounds for individual parts of the message estimate m_{sr}, m_e, m_f and m_g, we obtain the bound of the lemma. □

The following theorem follows from Lemma 5.

Theorem 3. *For face smooth graphs of bounded degree, if the geocast region size is constant, the message cost for SPG+SF is in $O(d + \sqrt{A})$, where d is the length of the sr-line and A is the area covered by the graph.*

Let us compare this bound with the message complexity of ordinary flooding. If the number of devices in the graph is proportional to this area, the message cost of flooding is in $\Omega(A)$. In other words, the message cost of SPG+SF is proportional to the linear dimensions of the geocast region while the cost of flooding is quadratic.

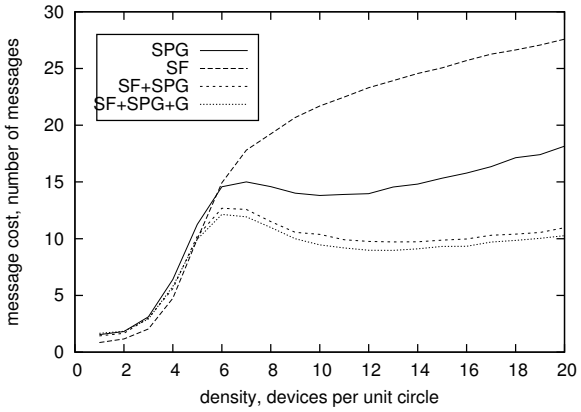
5 Simulation

Setup. In their classic study of unicast geometric routing algorithms, Kuhn et al [12] use a particular simulation setup to thoroughly evaluate the performance of their algorithms. We extend the setup similar to theirs to use in our simulation.

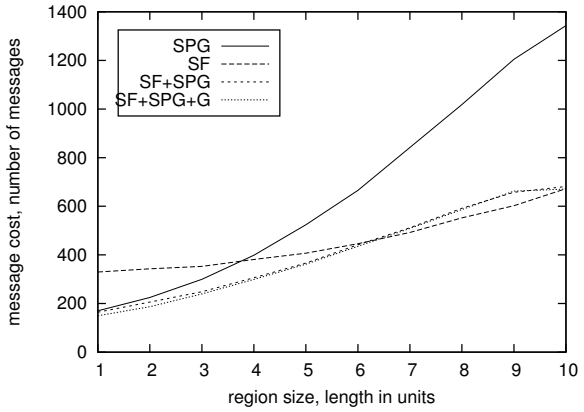
Specifically, we populate a 10×10 unit square field with devices placed uniformly at random to achieve a specific network density. The total number n of devices is equal to the area of the field divided by area of the unit circle and multiplied by the required density d. That is $n = d\frac{100}{\pi}$. We randomly pick the source device and randomly place a square geocast region so that it fits into the field completely. We then calculate each device's neighbors as follows. We first construct a unit-disk graph. For the planar geocasting algorithms, we also compute Gabriel subgraph and connected dominating set on it.

Experiment is a single delivery of a message from a particular source to a particular geocast region. In other words, it is a single complete computation of an algorithm. For each experiment, we generate a new random graph with a randomly selected source and randomly placed geocast region. For each specific data point, we conduct 1000 experiments.

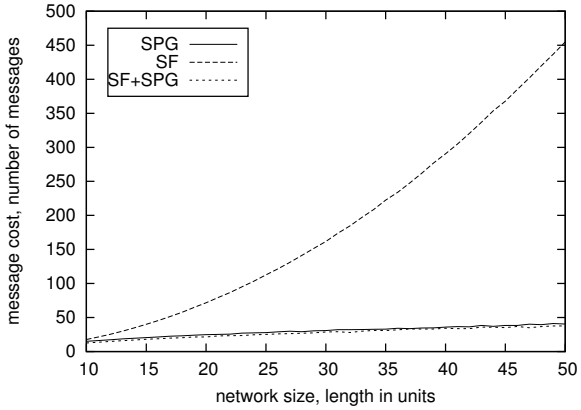
Results. We evaluate the algorithms on message cost and latency. *Message cost* is the number of messages it takes to deliver to all devices in the geocast region. Message cost



(a) by network density

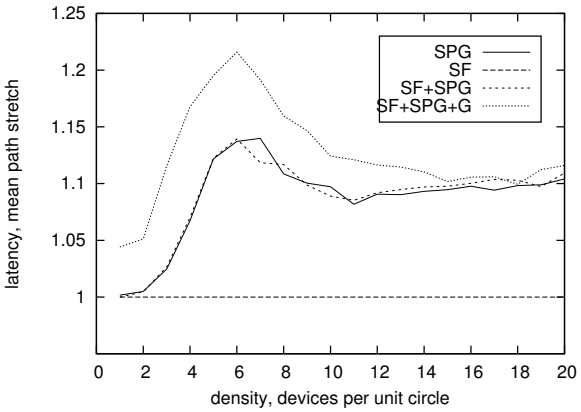


(b) by geocast region size, length of geocast square side

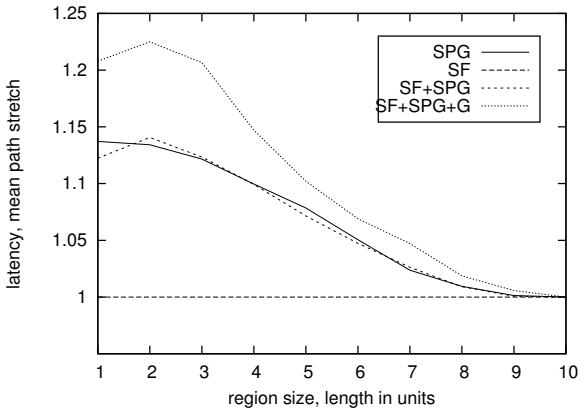


(c) by network size, length of field square side

Fig. 6. Message cost normalized to geocast region size.



(a) by network density



(b) by geocast region size

Fig. 7. Arrival latency normalized to optimal path.

quantifies the amount of network resources necessary to deliver the message. *Latency* of message arrival is the shortest path taken by the algorithm to reach the device in the geocast region that is furthest away from the source. Devices not connected to the source are not counted. *Path stretch* is latency normalized to the optimal path to this device. Latency quantifies the time it takes to deliver the message to every device in the geocast region.

We estimate message cost by varying three parameters: network density, geocast region size and complete communication field size. When we vary one of the three parameters, the other two are held constant at: 7 devices per unit square for density, 3×3 units for geocast region size and 10×10 units for field size. Figure 6 presents the simulation results.

We study latency by varying network density and geocast region size. The results for latency are shown in Figure 7.

Analysis. The results of the message cost study are intuitive. SF becomes comparatively costly as the density of the network increases (see Figure 6(a)). Indeed, SF delivers the message to every device in the whole network, regardless of whether they are inside or outside the geocast region. The delivery to the outside devices is overhead. As the density grows, the ratio of outside devices to inside devices also grows. The overhead grows with the increase of this ratio. SF+SPG performs better than either of the two individual algorithms. The combined algorithm achieves message savings compared to pure SPG since it floods the geocast region. When flooding, the algorithm sends only one message per edge, while SPG may potentially send two messages. Adding greedy to the combined algorithm helps further reduce message cost.

Let us consider geocast region variation. Again, since SF sends one message per edge, while SPG may potentially send two messages, SF outperforms SPG as the geocast region size approaches field size (see Figure 6(b)). The growth of the field size adversely affects SF's performance (see Figure 6(c)).

The results of the latency study are also intuitive for the most part. SF is always latency-optimal as all possible paths are traveled. The other algorithms achieve similar mean path stretch, whether under varied density or geocast region size. However, adding a greedy component dramatically worsens the algorithm's performance: greedy routing does not have the advantage of concurrently exploring multiple paths to find the faster one to deliver the message.

6 Future Research

In conclusion, we would like to point out several ways the algorithms presented in this paper may be extended. We assumed fault-free delivery for our geocasting algorithms. It would be interesting to incorporate fault-tolerance and message-loss resilience into them.

Other practical considerations enhance the applicability of our algorithms. For example, both SF and SPG, upon arrival of a message to a device, require examination of the send queue at this device. This examination has to be implemented at all levels of the network stack of each device. Such cross-stack queue examination presents an interesting implementation challenge.

References

1. Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless networks*, 7(6):609–616, 2001.
2. Arnaud Casteigts, Amiya Nayak, and Ivan Stojmenovic. Multicasting, geocasting, and anycasting in sensor and actuator networks. *Wireless Sensor and Actuator Networks*, page 127, 2010.
3. Thomas Clouser, Mark Miyashita, and Mikhail Nesterenko. Concurrent face traversal for efficient geometric routing. *Journal of Parallel and Distributed Computing*, 72(5):627–636, 2012.
4. Susanta Datta, Ivan Stojmenovic, and Jie Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks. *Cluster computing*, 5(2):169–178, 2002.
5. K Ruben Gabriel and Robert R Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969.
6. Tomasz Imieliński and Julio C Navas. Gps-based geographic addressing, routing, and resource discovery. *Communications of the ACM*, 42(4):86–92, 1999.
7. David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.
8. Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.
9. Young-Bae Ko and Nitin H Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pages 101–110. IEEE, 1999.
10. Young-Bae Ko and Nitin H Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.
11. Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *in Proc. 11 th Canadian Conference on Computational Geometry*. Citeseer, 1999.
12. Kuhn, Wattenhofer, Zhang, and Zollinger. Geometric ad-hoc routing: Of theory and practice. In *PODC: 22th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 2003.
13. Jie Lian, Kshirasagar Naik, Yunhao Liu, and Lei Chen. Virtual surrounding face geocasting with guaranteed message delivery for ad hoc and sensor networks. In *Network Protocols, 2006. ICNP'06. Proceedings of the 2006 14th IEEE International Conference on*, pages 198–207. IEEE, 2006.
14. Julio C Navas and Tomasz Imielinski. Geocastgeographic addressing and routing. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–76. ACM, 1997.
15. Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.
16. Karim Seada and Ahmed Helmy. Efficient and robust geocasting protocols for sensor networks. *Computer Communications*, 29(2):151–161, 2006.
17. Godfried T Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern recognition*, 12(4):261–268, 1980.
18. Peng-Jun Wan, Khaled M Alzoubi, and Ophir Friede. Distributed construction of connected dominating set in wireless ad hoc networks. In *INFOCOM 2002. Twenty-First annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, volume 3, pages 1597–1604. IEEE, 2002.