2017 IEEE 36th Symposium on Reliable Distributed Systems

Stateless Reliable Geocasting

Jordan Adamek*, Mikhail Nesterenko*, James Scott Robinson*, and Sébastien Tixeuil†
*Kent State University †UPMC Sorbonne Universités & IUF

***Abstract*—We present two geometric routing algorithms that reliably deliver messages to all devices in a geocast region. One algorithm is based on flooding, the other on concurrent geometric routing. They are the fist known stateless geocasting algorithms. We formally prove the algorithms correct, evaluate their performance through abstract and concrete simulation and estimate their message complexity.**

***Index Terms*—reliable geocasting, geometric routing, stateless routing, simulation**

I. Introduction

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

In such environment, statelessness is a compelling approach to routing. Routing is *stateless* if devices do not store any information about the transmitted message between transmissions. This is a particularly attractive property: it is immediately applicable to mobile and vehicular ad hoc networks; it is configuration change- and fault- tolerant as the system trivially adjusts to them; 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.

One of the simplest routing algorithms is *controlled flooding* where each device retransmits the message to all its neighbors. Classic flooding is statefull as each device needs to store the information about the transmission to prevent duplicate message resends.

Geometric routing offers a more scalable and resource frugal solution 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 [15], [17], [18], [19], [23]. Geometric routing usually requires that the message carries only a fixed number of device coordinates. That is, it allows *constant size message routing*. Geometric routing may be unicast, where message is to be delivered to a single target device, or multicast [24] where there are several targets.

Geocasting is a problem of delivering a message from a single source to devices in a particular region. Reliable geocasting guarantees delivery to every device in the region.

The work of Sébastien Tixeuil was supported in part by LINCS.

For example, geocasting may be used to locate a moving vehicle whose last known coordinates are available at the source; or notify all households in a flood-risk area once the water level reaches some critical point. Unlike unicasting or multicasting, where the source has to be aware of the coordinates of the targets and the message has to carry these coordinates; in geocasting, the target coordinates are unknown, the coverage area is specified instead.

Related work. Let us cover unicast geometric routing first. A number of unicast stateless geometric routing algorithms are presented in the literature [5], [8], [9], [16], [18], [19], [20]. 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 [5]. Face routing may be inefficient if traversed faces are large. Greedy-Face-Greedy [9] 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+[20] finds the shorter traversal direction by reversing it once the message reaches a pre-determined ellipse containing source and target devices. Concurrent Face Routing [8] optimizes the speed of message delivery by sending two concurrent messages in the opposite traversal directions.

Several multicasting geometric routing algorithms are available in the literature [1], [7], [29]

Let us now discuss existing geometric geocasting algorithms. Geographic-Forwarding-Geocast [25] 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. [6], 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. That is, this algorithm does not carry out reliable geocasting. Virtual Surrounding Face [22] 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

978-1-5386-1679-6/17 $31.00 © 2017 IEEE
DOI 10.1109/SRDS.2017.13

outside. The algorithm does reliable geocasting. However, it 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. [5] propose to combine GFG with depth-first face exploration to implement geocasting. The face exploration is based on subdivision traversal proposed by de Berg et al [10] and improved by Bose and Morin [4]. The subdivision traversal is stateless and the geocasting is reliable. However, a subdivision, unlike face, assumes that every edge is adjacent to exactly two subdivisions. That is, an edge whose both sides are adjacent to a single face is disallowed. Letting this algorithm run on an arbitrary planar graph may lead to a livelock. Eliminating such internal edges may potentially require extensive pre-processing. Thus, to the best of our knowledge, all existing geometric geocasting algorithms are either unreliable or stateful or require pre-processing.

Our contribution. We present the first reliable geocasting 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 [5], [22], [25] 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 abstract and concrete simulation. From our analysis, it follows that the presented algorithms have good reliability, low latency and high message efficiency.

II. 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 bi-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 to how to route the messages to the targets.

To help with routing, a message carries routing information. We consider *constant message size* routing algorithms where each message may carry only a fixed number of devices' coordinates and related information. This fixed number is independent of the network size. This limitation, 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.

Steps, computations, fairness. 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, transferring it to the receiver and processing it according to the routing algorithm. In this paper, we assume that this transferal and processing is done in a single atomic *step*. The *atomicity* of the step means that it may not overlap with steps on this or other devices. In practice, only the neighbor device steps may not overlap.

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. For short, we call this planar embedding of a graph, a *planar graph*. 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 Euclidean 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 [5], [14], [16], [26]. Moreover, a local computation on a unit-disk graph may yield a fixed maximum degree connected dominating set subgraph [28]. In our message complexity estimations and in our simulation, we consider the original graph to be unit-disk. This is a common assumption for geometric routing algorithms.

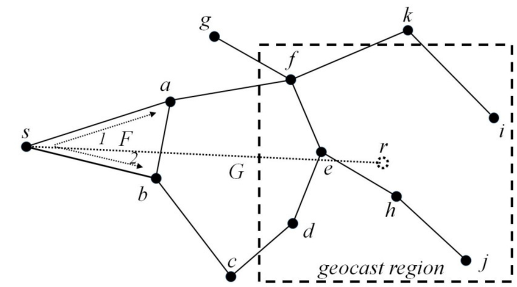
Face is a region on the plane such that for any two points in the region, there is a continuous curve that connects them without intersecting graph edges. Note, for example, faces F and G in Figure 1. A planar embedding of a finite graph divides the plane into a finite set of faces. The areas of all faces but one are finite. They are *internal* faces. One face is an infinite *external* face.

Consider device a and its neighbors b and c. Device c is *next-right* after b, if it is next neighbor of a after b clockwise; it is *next-left* after b, if it is next to it counter-clockwise. Note that if c is next left to b, then b is next-right to c. For example, in Figure 1, a is a next-right neighbor of b after s. Device a and its two next neighbors b and c and the two incident edges form *angle* $\angle bac$. Angle $\angle bac$ *intersects* a line if at least one of the edges of the angle intersects this line. Note that the angle also intersects a line if at least one of the incident devices a b or c itself lies on this line. Angle $\angle bac$ intersects an area if at least one of the incident devices lies inside or on the border of the area. Note that we limit angle intersection to the fixed-size graph edges, not to the infinite half-line rays of a classic geometric angle. For example, in Figure 1, $\angle asb$ intersects sr-line and $\angle bcd$ intersects the geocast region area.

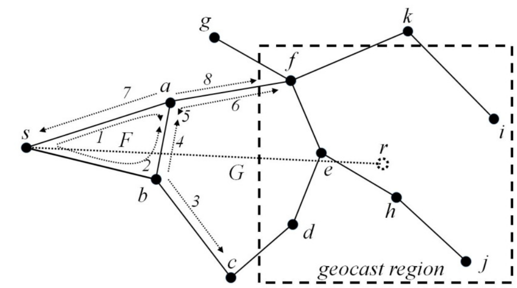
In a planar graph, messages are routed by traversing planar faces using right- or left-hand-rule. In the *right-hand-rule*, if device receives a message, it forwards the message to the next-right neighbor after the sender. In the *left-hand-rule*, the message is forwarded to the next-left neighbor. For example, in Figure 1, if b receives a right-hand-rule traversal message from s, b forwards it to a.

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. That is, in traversal algorithms, the mates are traversing the same face in the opposite directions.

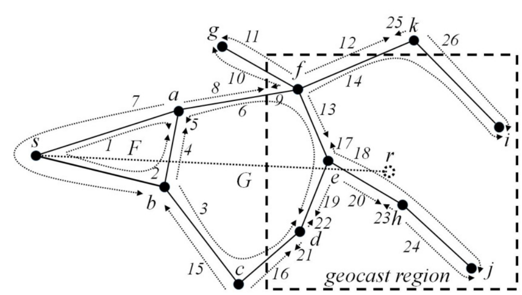
Geocasting. The problem of *geocasting* is communicating a message from a source device to all devices located in a designated *geocast region*. In other words, every device in the geocast region is a target. The geocast region is often a circle or rectangle. Note that the source itself may be inside the geocast region. Unicasting or multicasting algorithms may not be immediately applicable to geocasting since the coordinates of the devices in the geocast region are not known to the source. The problem of reliably delivering the message to all devices is complicated by the fact that devices in the geocast region may only be connected through outside devices. For example, device i in Figure 1 is connected to the rest of the devices through k which is outside the geocast region. Thus, message delivery to all devices in the geocast region requires exploring these outside connecting paths.



(a) initial step



(b) mid-computation



(c) completed computation

Fig. 1: Example computation of SPG.

```
device s
    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. 2: SF pseudocode.

```
device s
foreach intersection angle ∠asb do
    add L(s,r,G,a) to SQ
    add R(s,r,G,b) to SQ

device n
if receive L(s,r,G,a) then
    if R(s,r,G,a) ∈ SQ then
        /* found mate */
        discard R(s,r,G,a) from SQ
    else
        add L(s,r,G,b) to SQ
        /* let b be the next left after a */
        if ∠anb is an intersection angle then
            /* split message */
            foreach ∠cnd ≠ ∠anb such that
                ∠cnd is intersection do
                    /* let d be the next left after c */
                    add R(s,r,G,c) to SQ
                    add L(s,r,G,d) to SQ
if receive R(s,r,G,a) then
    /* handle similar to L(s,r,G,a) */
```

Fig. 3: SPG pseudocode.

III. Algorithm Descriptions

For all the algorithms, we assume that as soon as a device, including the source device, obtains the message, it checks if the device is in the geocast area. If it is, then the message is delivered first and then the processing of the message proceeds.

SF. The pseudcode for *stateless flooding* (SF) routing algorithm is shown in Figure 2. 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 geocasting. We start the algorithm description with a few definitions. Select a point r inside the geocast area, for example, the point nearest to the source. The location of r does not affect the correctness of the algorithm but may affect its efficiency. Every message carries the coordinates of the source device s and this point r so that every device can compute the *sr-line*: the line segment that connects the source with this point r. Also, every message carries the encoding of the geocast area G so that every device can determine this area. For example, if the geocast area is a square, carrying the center of the square r together with the length of the square's side is sufficient.

Intersection is an angle of a device such that this angle intersects either the sr-line or the geocast region. A device is *juncture* if at least one of its angles is an intersection. Note that all devices inside the geocast region are junctures. Note also that the source device s is always a juncture.

The pseudocode for SPG is shown in Figure 3. When receiving a message traversing a face with a particular traversal direction: either right- R or left-hand-rule L, the device checks to see if a mate is present in its send queue. If a mate is found, both messages are discarded. Otherwise, the device forwards the message to the respective next right or next left neighbor of the original message sender. If the recipient device also determines it is a juncture, it *splits* the message by sending R and L messages along the two edges of each intersection. The algorithm starts when s, which is juncture, initially splits the message. This splitting ensures that every face adjacent to a juncture is traversed.

We illustrate the operation of SPG with an example shown in Figure 1. Since $\angle asb$ intersects the sr-line, device s initiates the geocasting by sending a left-hand-rule message 1 to device a and right-hand-rule message 2 to device b. See Figure 1a. This starts the traversal of face F. Both a and b are junctures. Indeed, a has an angle that intersects sr-line and the geocast region; while b has an angle intersecting sr-line. Once 2 reaches b, b forwards it to a and splits it by sending messages 3 and 4 in face G. See Figure 1b. Note that $\angle sbc$ 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 1c.

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 sending 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.

IV. Correctness Proofs

Correctness proofs. We focus on SF first. Let us introduce notation that 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 sent 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 holds a message to every neighbor except a. This satisfies the conditions of the lemma. If b is already visited, then, by assumption, it has a message to be sent to a in its SQ. This message 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* with respect to a particular face 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. Similarly, a device may be adjacent to multiple faces and visited separately in each face. However, by the design of the algorithm, if a juncture is visited, it splits the message in every adjacent face that intersects the sr-line or the geocast region. That it, a juncture is visited in every such face simultaneously. 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, for every face F with a visited segment, this segment-border device has a message to send across the unused edge that is adjacent to this device. A segment-internal device never holds such a message.

Proof: The proof is by induction on the devices of a particular face F. A visited segment is created in F 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.

First, let us consider a message transmission by a device adjacent to F. By hypothesis, 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 in F, 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 a that is not adjacent to F. The only way it may affect F is if a is a juncture of F in an adjacent face. However, by the design of the algorithm, the juncture is instantly visited in every adjacent face. When a receives the transmission, it may or may not be visited in F. If it is visited, then this transmission encounters a mate, both messages are eliminated and the transmission does not affect F. If a is not visited in F, then, when the transmission occurs, it creates a new visited segment in F with a single border node a and the proper outgoing messages.

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. After all devices are visited, none of them hold a message.

Proof: If a face with a visited segment 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. Once all the devices are visited, they become internal. According to Lemma 2, internal devices do not old messages. Hence the lemma □

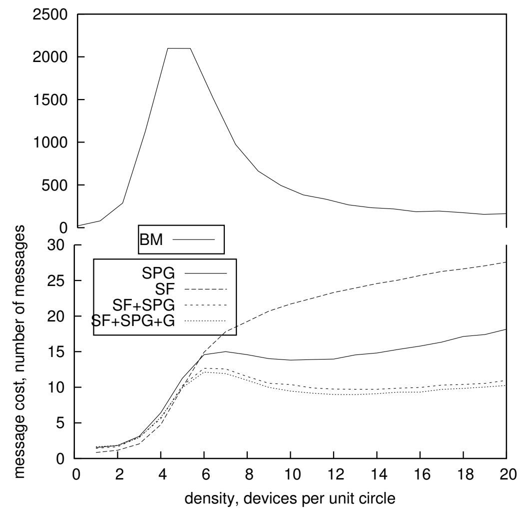
Lemma 4: In SPG, every device in a face connected to the source device face is eventually visited and no devices adjacent to it holds a message to send.

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. □

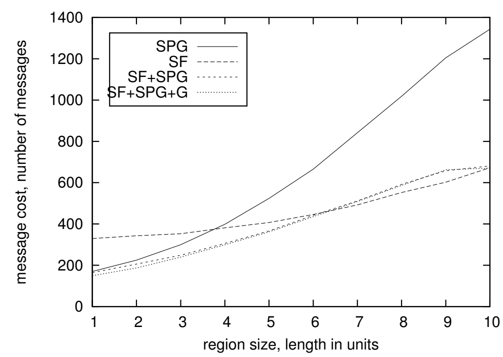
Proposition 1: In a planar graph, if a target device is connected to the source device, then this target device lies on a face connected to the source device face.

The below theorem follows from Proposition 1 and Lemma 4.

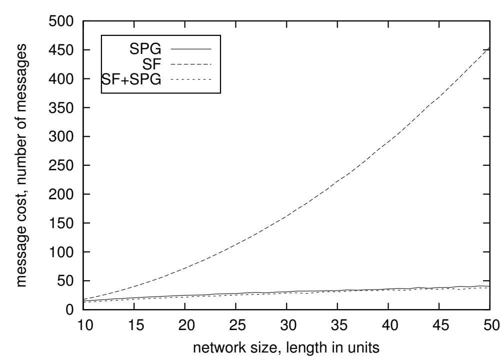
Theorem 2: SPG guarantees termination and delivery of a message from the source to all target devices connected to the source.



(a) by network density; note that y-axis is "broken" and shows BM at a scale different from the rest of the algorithms



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



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

Fig. 4: Abstract Simulation: Message cost normalized to geocast region size.

V. Abstract Simulation

Setup. In their classic study of unicast geometric routing algorithms, Kuhn et al [20] use a particular simulation setup to thoroughly evaluate the performance of their algorithms. We extend their setup 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 $n = d\frac{100}{\pi}$.

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. We ensure that the geocast region 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. For each specific data point, we conduct 1000 experiments.

Results. We implement SF, SPG, SF+SPG and SF+SPG+G. We also implement the geocasting algorithm BM based on subdivision traversal [4], [5]. Since BM does not operate correctly on faces with internal edges we drop the experiments where messages livelock.

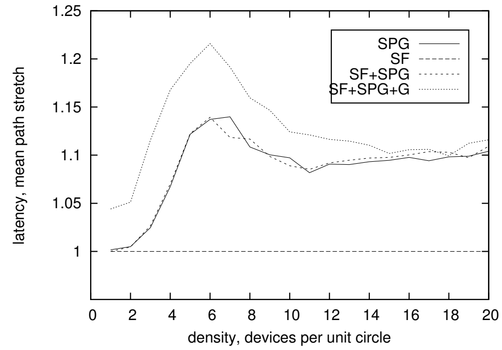
We evaluate the message cost and latency of the algorithms. *Message cost* is the number of messages it takes to deliver to all devices in the geocast region. Message cost quantifies the amount of network resources necessary to deliver the message. *Latency* of message arrival is the shortest path taken 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. Latency quantifies the time it takes to deliver the message to every device in the geocast region. *Path stretch* is latency normalized to the optimal path to this device.

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. Figures 4 and 5 show the simulation results for message cost and latency respectively.

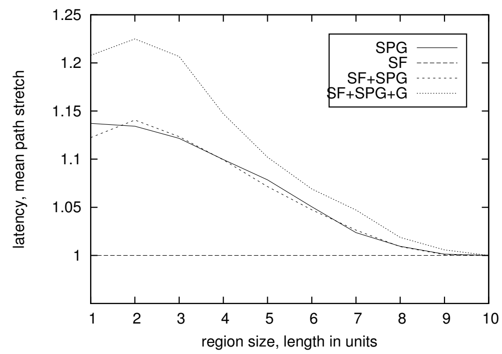
Analysis. Let us discuss message cost evaluation. BM is two orders of magnitude more expensive than the rest of the algorithms. So we only present the results in Figure 4a for comparison. SF becomes comparatively costly as the density of the network increases (see Figure 4a). Indeed, SF delivers the 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 4b). The growth of the field size adversely affects SF's performance (see Figure 4c).

Let us now discuss the latency results. 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.



(a) by network density



(b) by geocast region size

Fig. 5: Abstract Simulation: Arrival latency normalized to optimal path (path stretch).

VI. Concrete Simulation

Setup. To evaluate the performance of our algorithms in concrete wireless environment, we implement them in WS-Net [3], [11], [12], [13] wireless sensor network simulator. The simulated MAC layer is IEEE 802.15.4 with 866 MHz frequency band and BPSK modulation. The radio model is freespace propagation model with constant path loss and rayleigh fading [3]. We implement 1×1 km field. The geocast region is a 300×300 meters square. The network topology is assumed to be a unit-disk of 100 meters. The topology is calculated offline. We evaluate delivery ratio, latency and message cost. The *delivery ratio* is the number of devices in the geocast region that received the message divided by the total number of devices in the region. In case of imperfect delivery, the ratio is less than one. The *latency* is the amount of simulated time it takes to deliver the message to the device in the geocast region normalized by the amount of time it takes to deliver the message if its unicast over optimal route. The *message cost* is the number of messages the algorithm takes normalized by the number of devices in the geocast region.

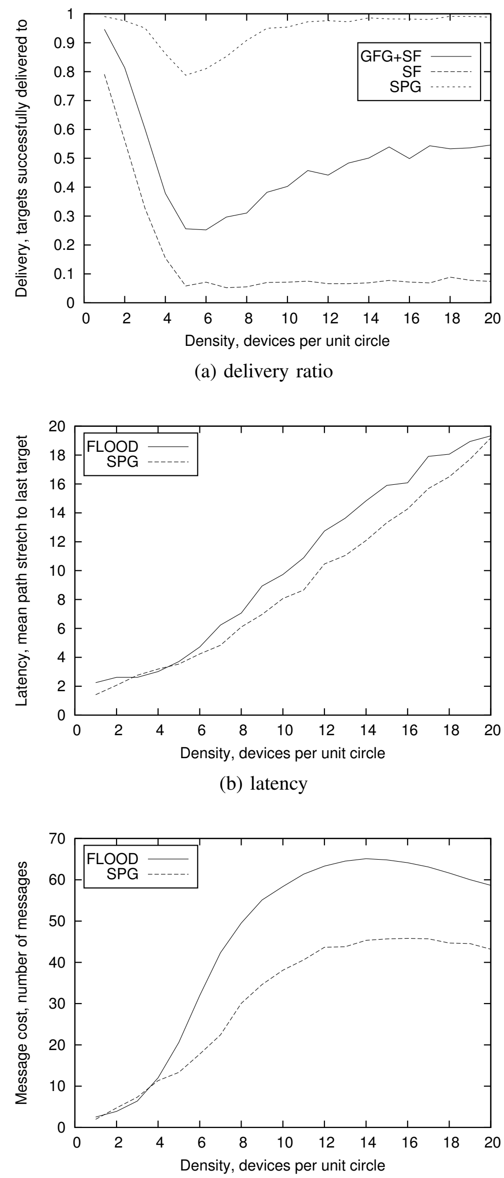
If messages are lost, SPG messages may not find mates and continue circling the faces indefinitely. To overcome this, we introduce time to live (TTL) in each message, after which the message is discarded. Signal strength affects the message reception due to signal attenuation and message transmission interference. We vary signal strength to evaluate its effect on our algorithms. Similar to the abstract simulation, we construct a new random graph for each message delivery and conducted 1000 experiments for each data point. Unlike the abstract simulation, a single broadcast to all neighbors is counted as a single message.

Results. First, we compare the performance of SPG with other strategies that can be used for geocasting. Specifically, we compared SPG with stateless flooding SF and with classic stateful flooding (FLOOD). In addition, we consider a strategy where a message is delivered to the geocast region using geometric unicast (GFG) and then flooded inside. For the SPG, we select the signal strength of 0 dBm and the TTL of 55. The results are shown in Figure 6. Due to its obvious inefficiency BM was not simulated. Due to poor delivery ratio of SF (see Figure 6a), the latency and message cost results for SF are inconsistent and are not shown in Figures 6b and 6c.

Then, we evaluate the performance of SPG at different power levels. The selected power levels are: 0 dBm – Bluetooth standard radio transmission power, 15 dBm – typical Wireless LAN transmission power, 33 dBm – cellphone transmission power. The results are shown in Figure 7. The plot in Figure 7a shows the influence of TTL size to the delivery ratio. In the rest of the plots of Figure 7, we used the TTL of 55 hops which appears optimal.

Analysis. Figure 6 shows that our face-traversal based algorithm SPG has significantly better delivery ratio than simple flooding-based approaches. Flooding leads to imperfect delivery due to incomplete geocast area connectivity and due to broadcast-storm-based message loss [27]. It seems that fewer messages result in better delivery. Thus, GFG+SF produces better results than simple flooding, see Figure 6a. The effect on latency is the opposite. An algorithm with greater message volume takes longer to capture the channel and transmit the message. Hence, the latency of stateful flooding is greater than that of SPG, see Figure 6b. Alternatively, the message cost of simple flooding is greater than that of SPG, see Figure 6c.

Let us now comment on the signal strength based experiments 7. The TTL affects message delivery up to about 55 hops which seems to be the saturation point. SPG provides near-perfect delivery except for the critical region (3-8 devices per unit disk) where weak signal strength adversely affects it. Delivery latency predictably grows with the network density. For the critical region, the higher signal strength appears to provide better latency due to more assured delivery, while for higher densities, the higher signal strength is a liability due to greater amount of interference. The message cost also grows with network density. However, higher signal strength requires fewer messages due lower signal attenuation and more reliable delivery.



(a) delivery ratio

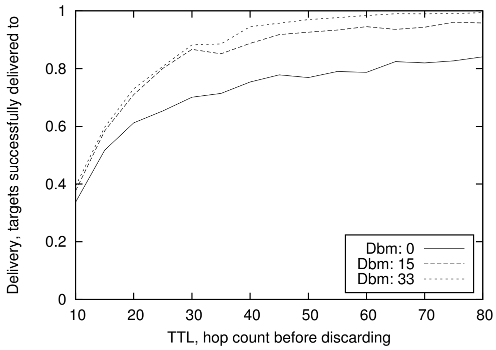
(b) latency

(c) overhead

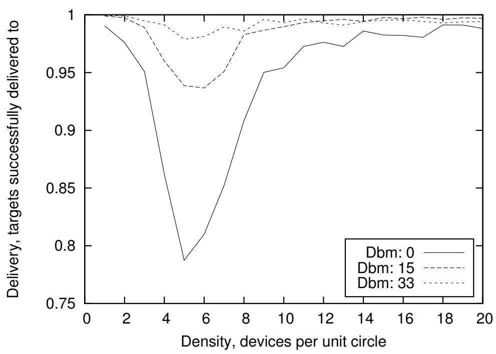
Fig. 6: Concrete simulation: Comparing SPG with stateful flooding (FLOOD) and GFG plus flooding in the geocast area (GFG+SF).

VII. Efficiency Bounds

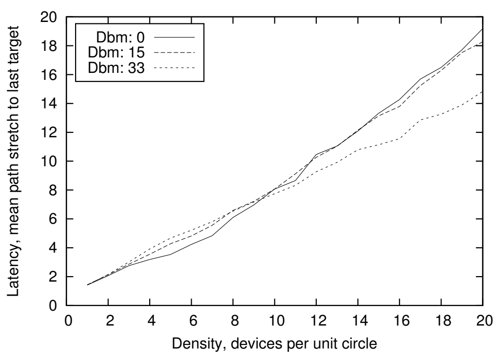
Message cost. In the case of stateless flooding (SF), each device sends exactly one message. That is, the total number of messages is $|E|$. In case of stateless planar geocasting (SPG), a message may be sent along each face. An edge may be adjacent to two faces. Hence, SPG may send $2|E|$ messages.



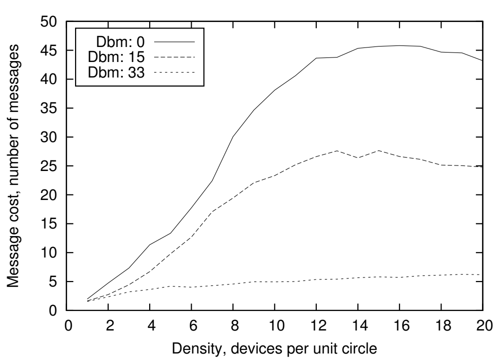
(a) Delivery ratio by TTL



(b) Delivery ratio by density



(c) Latency by density



(d) Overhead by density

Fig. 7: Concrete simulation: Comparing SF at different signal strengths

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 a 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. These assumptions hold for most realistic wireless communication graphs such as unit-disk graphs.

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.

See [2] for the proof.

Latency. *Latency* of a geocasting algorithm is the shortest path taken by this algorithm to the device in the longest distance from the source. Let ρ to be length of this path. A flooding algorithm concurrently explores all paths. Hence, it has the optimal latency. We, therefore, are focusing on estimating the latency of SPG.

Theorem 4: For a convex geocast area the latency of SPG is in $O(\rho^2)$ which is optimal for geographic routing algorithms.

Proof: Since we define path stretch for geocasting for a single node, we use the complexity results for a similar unicast algorithm [8, Theorem 2]. This algorithm concurrently traverses all the planar faces that intersect the source-target line. Specifically, the latency for such algorithm is in $O(|source, target|^2)$.

Let us consider the operation of SPG. When device s geocasts a message, it first reaches device r the in geocast area which is closest to s. SPG then sends the message to all devices in the area, including device t that has the longest path to s. That is, the shortest path selected by SPG between s and t contains r. SPG behaves like a concurrent unicast algorithm for the pair s, r. Now, since the geographic area is convex, line r, t lies completely inside the geocast area. Therefore, every face that intersects this line is traversed by SPG. That is, the length of the shortest path taken by the SPG is no longer as that for a unicast algorithm sending a message from r to t. That is for both segments of the path selected by SPG, we can use the unicast algorithm estimates.

Since r is the closest to s device in the geocast area, $|sr| \leq |st|$. Also, due to triangle inequality: $|rt| < |sr| + |st| \leq 2|st|$.

Therefore, $|sr| + |rt| < 3|st| = 3\rho$. Hence, the latency of the path selected by SPG is in O(ρ^2).

The optimality of this estimate follows from Kuhn et al [21, Theorem 5.1] which states that no geometric routing algorithm can select a path whose latency is less than quadratic of the optimal path length. □

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.

VIII. Future Research

The research described in this paper presents novel stateless algorithms for reliable geocasting. However, there is plenty of further work to be done. Our simulation shows optimal parameters such as TTL and signal strength for particular network topologies and configurations. However, it would be interesting to study the behavior of SPG in real network deployments. Another promising research direction is to design a geometric routing algorithm that adaptively selects its parameters based on the properties of the specific network.

References

[1] Jordan Adamek, Mikhail Nesterenko, James Scott Robinson, and Sébastien Tixeuil. Concurrent Geometric Multicasting. Research Report hal-01540744, UPMC Sorbonne Universités, 2017.

[2] Jordan Adamek, Mikhail Nesterenko, and Sébastien Tixeuil. Stateless Geocasting. Research Report hal-01168488, UPMC, 2015.

[3] Elyes Ben Hamida, Guillaume Chelius, and Jean-Marie Gorce. On the complexity of an accurate and precise performance evaluation of wireless networks using simulations. In *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 395–402. ACM, 2008.

[4] Prosenjit Bose and Pat Morin. An improved algorithm for subdivision traversal without extra storage. *International Journal of Computational Geometry and Applications*, 12(4):297–308, 2002.

[5] 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.

[6] Arnaud Casteigts, Amiya Nayak, and Ivan Stojmenovic. Multicasting, geocasting, and anycasting in sensor and actuator networks. *Wireless Sensor and Actuator Networks*, page 127, 2010.

[7] Kai Chen and Klara Nahrstedt. Effective location-guided tree construction algorithms for small group multicast in MANET. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM-02)*, volume 3 of *Proceedings IEEE INFOCOM 2002*, pages 1180–1189, Piscataway, NJ, USA, June 23–27 2002. IEEE Computer Society.

[8] 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.

[9] 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.

[10] Mark de Berg, Marc J. van Kreveld, René van Oostrum, and Mark H. Overmars. Simple traversal of a subdivision without extra storage. *International Journal of Geographical Information Science*, 11(4):359–373, 1997.

[11] Tony Ducrocq, Michaël Hauspie, and Nathalie Mitton. Geographic routing with partial position information. In Octavian Postolache, Marten van Sinderen, Falah H. Ali, and César Benavente-Peces, editors, *SENSORNETS 2014 - Proceedings of the 3rd International Conference on Sensor Networks, Lisbon, Portugal, 7 - 9 January, 2014*, pages 165–172. SciTePress, 2014.

[12] Tony Ducrocq, Michaël Hauspie, Nathalie Mitton, and Sara Pizzi. On the impact of network topology on wireless sensor networks performances: Illustration with geographic routing. In Leonard Barolli, Kin Fun Li, Tomoya Enokido, Fatos Xhafa, and Makoto Takizawa, editors, *28th International Conference on Advanced Information Networking and Applications Workshops, AINA 2014 Workshops, Victoria, BC, Canada, May 13-16, 2014*, pages 719–724. IEEE Computer Society, 2014.

[13] Antoine Fraboulet, Guillaume Chelius, and Eric Fleury. Worldsens: development and prototyping tools for application specific wireless sensors networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 176–185. IEEE, 2007.

[14] K Ruben Gabriel and Robert R Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969.

[15] Tomasz Imieliński and Julio C Navas. Gps-based geographic addressing, routing, and resource discovery. *Communications of the ACM*, 42(4):86–92, 1999.

[16] 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.

[17] 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.

[18] Young-Bae Ko and Nitin H Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.

[19] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *in Proc. 11 th Canadian Conference on Computational Geometry*. Citeseer, 1999.

[20] 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.

[21] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, Atlanta, Georgia, USA, September 2002.

[22] 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.

[23] Julio C Navas and Tomasz Imielinski. Geocast - geographic addressing and routing. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–76. ACM, 1997.

[24] Juan A. Sánchez, Pedro M. Ruiz, and Ivan Stojmenovic. GMR: Geographic multicast routing for wireless sensor networks. In *3d IEEE Communication Society Conference on Sensors and Ad Hoc Communications and Networks (SeCon)*, pages 20–29. IEEE, September 2006.

[25] Karim Seada and Ahmed Helmy. Efficient and robust geocasting protocols for sensor networks. *Computer Communications*, 29(2):151–161, 2006.

[26] Godfried T Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern recognition*, 12(4):261–268, 1980.

[27] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2/3):153–167, 2002.

[28] 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.

[29] Shibo Wu and K. Selcuk Candan. GMP: Distributed geographic multicast routing in wireless sensor networks. In *26th IEEE International Conference on Distributed Computing Systems (26th ICDCS'06)*, page 49, Lisboa, Portugal, July 2006. IEEE Computer Society.