

# Reliable Mobicast via Face-Aware Routing

Qingfeng Huang, Chenyang Lu, Gruia-Catalin Roman  
 Department of Computer Science and Engineering  
 Washington University, St. Louis, MO 63130, USA  
 Emails: {qingfeng,lu,roman}@cse.wustl.edu

**Abstract**—This paper presents a novel protocol for a spatiotemporal variant of multicast called *mobicast*, designed to support message delivery in sensor and mobile ad hoc networks. The spatiotemporal character of *mobicast* relates to the obligation to deliver a message to all the nodes that will be present at time  $t$  in some geographic zone  $Z$ , where both the location and shape of the delivery zone are a function of time over some interval  $(t_{start}, t_{end})$ . The protocol, called Face-Aware Routing (FAR), exploits ideas adapted from existing applications of face routing to achieve reliable mobicast delivery. The key features of the protocol are a routing strategy, which uses information confined solely to a node’s immediate spatial neighborhood, and a forwarding schedule, which employs only local topological information. Statistical results showing that, in uniformly distributed random disk graphs, the spatial neighborhood size is usually less than 20 suggest that FAR is likely to exhibit a low average memory cost. An estimation formula for the average size of the spatial neighborhood in a random network is another analytical result reported in this paper. This paper also includes a novel and low cost distributed algorithm for spatial neighborhood discovery.

**Index Terms**—System design, Graph theory, Simulations, Statistics, Spatiotemporal multicast, Wireless ad hoc networks, Sensor networks

## I. INTRODUCTION

Wireless sensor networks are large-scale distributed embedded systems composed of small devices that integrate sensors, actuators, wireless communication, and microprocessors. With advances in hardware, it will soon be feasible to deploy dense collections of sensors to perform distributed micro-sensing of physical environments. Sensor networks will serve as a key infrastructure for a broad range of applications including precision agriculture, intelligent highway systems, emergent disaster recovery, and surveillance [1]. Many sensor network applications have fundamental spatiotemporal constraints that do not exist in traditional applications of wireless ad hoc networks. Both sensor networks and mobile networks are increasingly heading towards supporting applications that demand spatiotemporal guarantees.

**Entity Tracking:** Many sensor networks (e.g., habitat monitoring [2] and intruder tracking [3]) need to handle physical entities that move in the environment. Only sensors close to an interesting physical entity should participate in the aggregation of data associated with that entity because activating distant sensors wastes precious energy without improving sensing fidelity. To continuously monitor a mobile entity, a sensor network must maintain an active sensor group that moves at the same velocity as the entity. This energy-efficient operation model [2] requires a communication mechanism that enables sensors to push information about a discovered entity to other sensors that the entity will approach in the future. The message must be delivered to sensors a certain time before the entity reaches their vicinity in order to wake up other sensors in time.

**Ambulance Warning:** Consider a scenario where an ambulance tries to inform vehicles down the road to yield the way. Currently, this is achieved by the ambulance using a siren which can be heard within a few blocks. We envision a more efficient warning system that alerts other vehicles of the location and velocity of the ambulance (the mobile event) through the multi-hop network formed by embedded sensors in vehicles. The spatial constraint requires that data about the ambulance only needs to be delivered to vehicles a few blocks down the road relative to the ambulance. The timing constraint requires that the delivery be done a few minutes before a potential collision can take place so the vehicles have enough time to react to the information (e.g., before they enter a narrow tunnel). As the ambulance moves, the relative geographic area of delivery changes accordingly.

As we have shown in the above examples, applications involving sensor and mobile networks require both spatial and temporal constraints to be satisfied simultaneously, i.e., data needs to be served at the right time and also at the right location. The spatiotemporal constraints motivate novel communication models tailored for sensor networks. This paper focuses on mobicast [4][5], a new class of multicast with spatiotemporal semantics tailored for sensor networks. Mobicast allows applications to specify their spatiotemporal constraints by requesting a

mobile delivery zone, which in turn enables the application to build a continuously changing group configuration, according to their spatial and temporal locality. Formally, a mobicast session is specified by a four-tuple,  $(m, Z[t], T_s, T)$ .  $m$  is the mobicast message.  $Z[t]$  is the mobile area where  $m$  should be disseminated at time  $t$ . As the delivery zone  $Z[t]$  evolves over time, the set of recipients of  $m$  changes as well.  $T_s$  and  $T$  are the sending time and duration of the mobicast session, respectively. A mobicast protocol should provide a spatiotemporal guarantee that all nodes that fall into a delivery zone within the lifetime of a mobicast session must receive the message  $m$  before they enter the delivery zone  $Z[t]$ . In this paper, we assume the delivery zone  $Z[t]$  moves at a constant velocity in space. Fig. 1 shows such an example. More complex mobility models (with changing veloci-

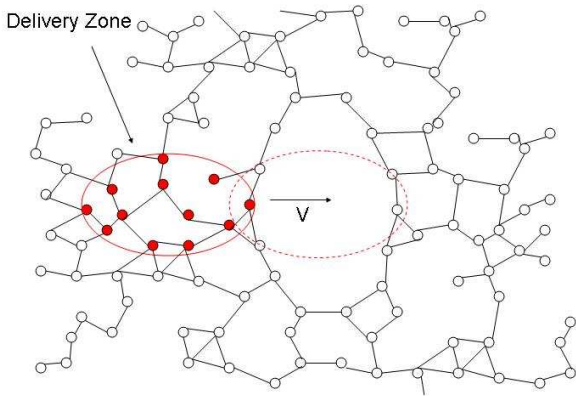


Fig. 1. A Constant-Velocity Mobicast Example

ties) can be approximated by a sequence of constant-velocity mobicast sessions. Mobicast provides a powerful communication abstraction for local coordination and data aggregation in sensor networks. For example, the group maintenance service for a mobile entity can be easily implemented on top of mobicast. When an interesting entity is discovered and a group is initiated, a group leader sends a mobicast message (including the estimated location and time of the discovery of the intruder) to a delivery zone that moves according to the estimated velocity of the intruder.

Providing spatiotemporal guarantees in mobicast introduces several key technical challenges. Since many sensor networks need to be deployed in an ad hoc fashion (i.e., dispersed from an airplane or vehicles), a mobicast protocol must achieve reliable and timely delivery to a dynamic set of nodes over random network topologies where routing voids are prevalent [6]. Fig. 1 illustrates an example in which the delivery zone is expected to move

across a hole on its path. At the same time, a mobicast protocol needs to scale to hundreds to thousands of nodes and minimize energy consumption. Naïve protocols for mobicast can either cause premature termination of a mobicast session due to network voids, or introduce excessive flooding overhead.

Previous work on mobicast [4][5] has explored several different approaches. The first mobicast protocol presented in [4] handles random network topologies by limiting message re-broadcasting to a mobile forwarding zone whose size depends on the compactness of the underlying geometric network. An absolute spatiotemporal guarantee can be achieved (under certain lower-level assumptions) by configuring the forwarding zone based on the global minimum compactness value which captures the notion of a worst case “hole” that might appear anywhere in the network. However, this protocol has two drawbacks due to its dependence on global knowledge about the network-wide minimum compactness. First, it cannot scale well to large and dynamic networks where the network compactness can change over time. Second, it can introduce high overhead (albeit lower than global flooding) because the forwarding zone is often unnecessarily large due to the pessimistic configuration based on minimum compactness. In [5], two other approaches were explored to address the above problems. To solve the first problem, a simple adaptive protocol was designed to dynamically change the size of the forwarding zone based on the local compactness of a node’s (multi-hop) neighborhood. To address the second problem, we found the broadcasting overhead can be reduced significantly by slightly relaxing delivery guarantees. However, the latter two approaches provide no absolute guarantees on the spatiotemporal delivery of mobicast.

This paper presents a new Face-Aware Routing protocol (FAR) for mobicast and a related spatial neighborhood discovery algorithm. FAR distinguishes itself from previous mobicast protocols by providing both reliability and scalability at the same time. Its scalability comes from the fact that it does not rely on any global topological information, and each node makes local forwarding decisions based on its *spatial neighborhood* configuration (defined in Section II), which is found to be small in average case via both theoretical analysis and simulation for random wireless ad hoc networks. We also prove in theory that FAR can reliably deliver a mobicast message to all nodes that ever enter the delivery zone.

The remainder of the paper is organized as follows. Section II describes the FAR mobicast protocol. Section III analyzes its delivery property. Section IV investigates geometric properties of planar graphs related to the

performance of FAR, especially those of spatial neighborhood. A spatial neighborhood discovery protocol is presented in Section V. Discussion, related work and conclusions are included in Sections VI and VII.

## II. FACE-AWARE ROUTING FOR MOBICAST

In this section we introduce the Face-Aware Routing (FAR) protocol for mobicast. A key contribution of this algorithm is that it does not rely on any global topology information for achieving theoretically reliable mobicast delivery. The idea of face routing is inspired by previous geometric routing algorithms such as GPSR [6] and GOAFR<sup>+</sup> [7]. They all have a face routing component to help their greedy forwarding component to get out of local minima in their unicast message forwarding path. However, these unicast protocols can not be applied directly to mobicast. There are two key problems. In unicast, the destination node is known, and so is its location in geometric routing scheme. The location of the destination node is key in determining the forwarding path and in detecting whether the greedy algorithm has gone into a local minimum. In mobicast, however, there is no single destination location, only the delivery zone is known, the exact locations of nodes in future delivery zones are not known. Simple approaches such as selecting some arbitrary location in the delivery zone path as a destination and use unicast protocols to reach the destination and dispatch the message to nodes close by does not work, since without a global node-location look up service, one can't even tell if a particular location has a node there or has any node close by. Moreover, the mobicast delivery zone is not fixed. A mobicast protocol must consider the temporal domain of information dissemination, which none of the previous geometric unicast protocols address. The FAR protocol addresses the first issue via some knowledge about its *spatial neighborhood* (to be defined later), and addresses the second problem by a novel timed face routing strategy.

For clarity, we first assume that the network is a planar graph. In general, a random wireless network may not be planar. Later we also discuss graph planarization methods and how the FAR algorithm can be modified to deal with a non-planar graph. We also assume that each node knows all of its spatial neighbors and their locations. We will provide an algorithm for obtaining this information and discuss the cost for storing such information in later sections. Next we first define the concept of a spatial neighborhood.

### A. The Planar Spatial Neighborhood

On a planar graph, each node has one or more adjacent faces. A face is the subdivision of maximal connected subset of the plane that does not contain a point on an edge or a vertex [8]. For instance, in the planar graph as shown in Fig. 2, node *A* has six adjacent faces, and node *B* has four adjacent faces. Note that the “boundary node” *M* has two adjacent faces. One of them is the “inner” face formed by nodes *M*, *L*, *G* and *H*, the other is the “outer face” formed by nodes *M*, *H*, *I*, *J*, *K*, *F*, *E*, *D*, *C*, *N*, *O* and *L*. Note also that even though the “boundary”, “inner” and “outer” faces of a planar graph seem visually easy to identify, topologically it is hard to define and distinguish them. This has important consequences on face-based geometric routing mechanisms. We will discuss this issue later.

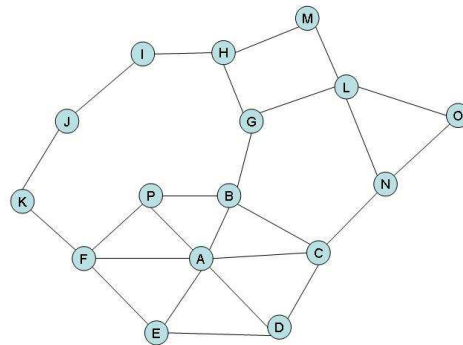


Fig. 2. Planar Graph and Planar (Spatial) Neighborhood

We define the “spatial neighborhood” of a node in a planar graph to be the set of nodes in all faces adjacent to that node except the node itself. So in Fig. 2, node *A* has six spatial neighbors (*B*, *C*, *D*, *E*, *F* and *P*) which are the same as its immediate graph neighbors. Yet node *G* has 10 spatial neighbors (*L*, *H*, *B*, *I*, *J*, *K*, *F*, *P*, *C* and *N*) while it only has three immediate graph neighbors (*L*, *H*, *B*). Note that the spatial neighborhood of a node *X* as we define it represents the set of nodes that can be reached from *X* without crossing an edge or other nodes, and in general is equal to or greater than the immediate graph neighborhood.

The spatial neighborhood information plays an important role in our face-based geometric forwarding strategies, just like immediate network neighborhood information is very useful for many routing algorithms.

### B. Face-Aware Routing

We now describe the face-aware routing algorithm. The essence of the algorithm is very simple: every

node that has at least one spatial neighbor that is a delivery-zone node will forward (locally broadcast) the mobicast packet once<sup>1</sup>. We will prove that this simple rule can guarantee all delivery zone nodes to receive the corresponding packet. Yet using this simple rule alone leads to an “as-soon-as-possible” style mobicast protocol that exhibits a high average slack-time which is not desirable [5]. We need certain temporal controls to achieve a just-in-time style mobicast protocol. As a result, the face-aware algorithm consists of two methods for forwarding packets: *greedy forwarding* and *timed forwarding*. Before discussing these two methods in detail, we first present the format of a FAR mobicast packet.

1) *Packet Format*: Each FAR mobicast packet contains the following information in its header: sender location, packet sending time, initial delivery zone coordinates, delivery zone velocity, message lifetime, message type, sender packet sequence number, and the last forwarder location. Similar to previous mobicast protocols[4][5], we do not assume each node has a unique ID. The sender location, the packet sending time stamp and the sender packet sequence number are jointly used to identify each packet on the network. The initial delivery zone field contains an ordered sequence of locations corresponding to the initial vertices of the delivery zone. For a circular delivery zone, the radius and the initial center are recorded instead. The message type field is used for indicating the type of delivery zone, e.g., rectangle, pentagon, circle, ellipse, etc. The initial delivery zone coordinates combined with the delivery zone velocity and packet sending time can be used to determine the location of the delivery zone at any point time in the future. The message lifetime is used for terminating each mobicast session. The last forwarder information is used for determining if further forwarding of a packet is needed. We will discuss these in more detail.

For simplicity, henceforth we assume each mobicast message fits in one packet, and we use the words packet and message interchangeably.

2) *Greedy Forwarding*: Greedy forwarding applies to all nodes that are currently (or previously)<sup>2</sup> covered by the mobicast delivery zone, or have at least one spatial neighbor that is currently (or previously) covered by the

<sup>1</sup>An optimization will change this to “forward the mobicast packet once, if necessary”. We try to keep it simple here and leave the optimization issue aside for the moment.

<sup>2</sup>The “previously” condition is unlikely to happen in most cases. Yet it is necessary here as our specification of mobicast requires all delivery zone nodes to receive the packet. If a new specification only admits on-time reception, then this “previously” is unnecessary.

mobicast delivery zone. In such cases, a node forwards a new packet in an “as-soon-as-possible” fashion.

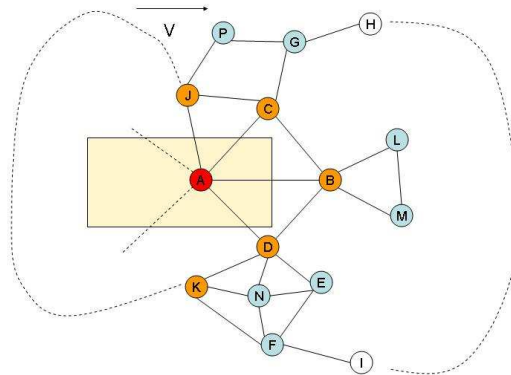


Fig. 3. Greedy and Timed Face-Aware Forwarding

Fig. 3 depicts an FAR mobicast example featuring a rectangular delivery zone moving to the right at speed  $v$  at a certain time instance. In this example, the greedy forwarding rule applies to nodes  $A, K, D, B, C, J$ , as they are either in the delivery zone or have a spatial neighbor that is in the current delivery zone. Note that the condition specifies a spatial neighbor rather than a direct neighbor, which causes  $K$  to be included.

Note that after  $K, D, B, C, J$  perform local broadcasts, nodes  $P, G, L, M, N, E$  and  $F$  all hear the mobicast message since they are each connected to at least one of the previous broadcasting nodes. But because  $P, G, L, M, N, E$  and  $F$  do not have spatial neighbors in the current delivery zone, they do not perform the greedy forwarding, and use timed forwarding instead if they have spatial neighbors to be in the delivery zone.

3) *Timed Forwarding*: Timed forwarding applies to a node that has no spatial neighbor in the current delivery zone but either itself will soon be in the delivery zone or have at least one spatial neighbor that will be in the delivery zone. Nodes  $H, G, L, M, E, F$  and  $I$  in Fig. 3 belong to this category. Nodes  $L$  and  $M$  are to be in the delivery zone themselves as to delivery zone move to the right. Nodes  $G, E$  and  $F$  find three of their spatial neighbors,  $B, L$  and  $M$  are to be in the delivery zone. Nodes  $H$  and  $I$  will discover the same after hearing the mobicast packet from  $G$  and  $F$ .

The timed forwarding method works as follows. If a node  $X$  receives a new mobicast packet at time  $t$  and finds itself in the timed forwarding category, it makes a forwarding decision based on the relative times that the delivery zone reaches its delivery zone neighbors and the expected communication latency between itself and those neighbors.

Let  $Y_1, Y_2, \dots, Y_k$  be the ordered list of all spatial



neighbors of  $X$  that will be in the delivery zone and  $\Delta t_1, \Delta t_2, \dots, \Delta t_k$  be the corresponding times for the delivery zone to reach them. Let  $h_1, h_2, \dots, h_k$  be distance from  $X$  in number. Let  $\tau_1$  be the expected 1-hop network latency. We have  $h_i \tau_1$  the expected communication latency between  $X$  and  $Y_i$ . Let  $T_a$  be the minimum time difference between the time for the delivery zone to reach  $Y_i$  and the expected latency  $h_i \tau_1$  for a message sent from  $X$  to reach  $Y_i$ . i.e.,

$$T_a = \min\{h_i \tau_1 - \Delta t_i | i = 1, 2, \dots, k\} \quad (1)$$

The forwarding decision of  $X$  is as follows:

- 1) If  $T_a \leq 0$  forward the packet as soon as possible;
- 2) If  $T_a > 0$  schedule a forwarding at time  $T_a$  from now.

In Fig. 3, nodes  $H, G, C, B, L, D, M, E, F$  and  $I$  share one face which extends to the east. Among them, nodes  $C, B$  and  $D$  have already greedily forwarded the packet. Nodes  $G, L, M, E$  and  $F$  have heard the packet and will schedule the forwarding according to the timed forwarding rule. From this example, one can also see this face forwarding algorithm can be improved. For instance, nodes  $L$  and  $M$  do not need to do the forwarding at all since their local broadcast effort does not help the mobicast packet reach any new node, and they have the local topology knowledge to learn that. Node  $G$  knows node  $B$  has received the message as it heard it from  $C$ , and  $B, C$  are connected. Note that  $G$  does not know if  $B$  has re-broadcast the packet but does know  $B$  will take care of  $L$  and  $M$ . So  $G$  may take  $B, L, M$  off its “care list”, i.e., the list of nodes used for computing the forwarding time. A similar argument is true for  $E$ . We will discuss optimization methods later.

Note that node  $F$  is a different case than  $G$  or  $E$ . It has heard the packet from node  $K$ , and it does not know if  $E$  has heard the message, or  $D, B, M, L, C, G$ . So its care list has to include  $B, L$ , and  $M$ . Note also that even though  $B$  is the earliest among its spatial neighbors to enter the delivery zone,  $F$  can not simply compute its forwarding time based on  $B$ , since  $h_{FL} \tau_1 - \Delta t_L$  may be smaller than  $h_{FB} \tau_1 - \Delta t_B$ .

In the previous discussion, we choose  $\tau_1$  to be the expected 1-hop latency. If one choose  $\tau_1$  to be the maximum 1-hop latency, the protocol will result in higher average slack time but less potentially late receptions.

Since every node makes the forwarding decision locally, it is possible for a node to receive a packet it has forwarded earlier. In this case a node simply ignores the packet. For a node to be able to determine which packets are new and which are old, every node maintains a local cache to log received packets. This cache is periodically checked, and packets that have expired are removed.

Note that in Fig. 3, although node  $N$  has heard the packet, it will never forward the packet since it has no spatial neighbor that is a delivery zone node. This is also true for node  $P$ .

4) *Protocol Termination:* In addition to greedy forwarding and timed forwarding, the algorithm also has a mobicast termination method based on the packet life time value in the packet header. A packet is not simply ignored if it has expired. An expired packet is dropped only in the timed forwarding mode, i.e., when the recipient node finds that no node in its care list is in any previous delivery zone. If a node is in greedy forwarding mode, it will forward the packet even if the packet has expired. This choice intends to tolerate some level of timing uncertainty by admitting marginal overhead caused by potential “expired face forwarding” in the last few faces in the delivery zone path. This also simplifies our statements and proofs of the delivery properties of the protocol later.

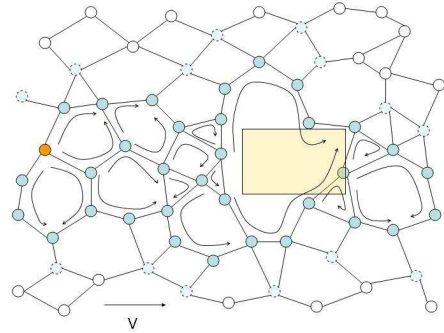


Fig. 4. Bird's Eye View on the FAR Protocol Behavior and Result

To help see a bigger picture of the behavior and results of the FAR algorithm, Fig. 4 schematically shows a rectangular mobicast history in a larger network context. The faces with arrows are those that have experienced face-forwarding. The solid circles represent the nodes that have forwarded the packet. The lightly shaded dashed circles represent those that have heard the packet but did not forward it. The empty circles never hear the packet. One can see that the face-aware forwarding algorithm creates a localized forwarding cloud (area) surrounding the mobile delivery zone, and the forwarding area adapts to the topology on the delivery zone path and makes the delivery zone cross holes in the network.

Next we prove that our forwarding strategy indeed delivers mobicast packets to all its expected recipients under one reasonable assumption.

### III. FAR DELIVERY GUARANTEE

The FAR algorithm guarantees the delivery of a mobicast packet to all its delivery zone nodes, under

the following assumption on the size of the forwarding zone: the forwarding zone span on the direction perpendicular to the mobicast velocity direction (we call it “perpendicular span” henceforth) must be no smaller than the maximum neighbor distance. (In wireless ad hoc networks, this may be interpreted as the perpendicular span to be no smaller than the maximum communication range). If the the perpendicular span is too small, the algorithm may terminate prematurely. Fig. 5 shows such an example in a partial network. Nodes  $J, C, G$  and  $K$  will not forward the packet because they have no spatial neighbor that is a delivery zone node. This results in  $E$ , a delivery zone node, never receiving the packet. Note that the constraint is only on the perpendicular span of the delivery zone. Small delivery zone size on the velocity direction is acceptable. Next we prove this

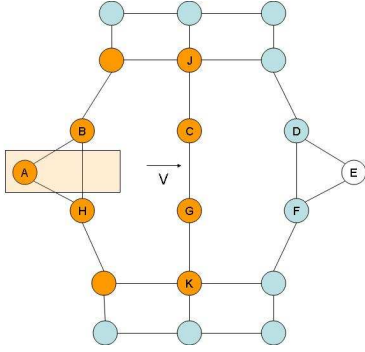


Fig. 5. FAR Assumption

delivery guarantee in the general case. We start from the following lemma.

*Lemma 1:* If  $X$  and  $Y$  are in the same face and  $X$  is a delivery zone node, the FAR protocol guarantees that if  $Y$  has received the mobicast packet,  $X$  either has received it or will receive it.

**Proof:** Assume that  $X$  has not received the packet.  $X$  will at some point in time be in the delivery zone. The fact that  $Y$  has received the packet means it has the data for computing the delivery zone trajectory over the packet lifetime.  $Y$  also has the knowledge of the locations of all its spatial neighbors which include  $X$ . So  $Y$  can compute if  $X$  is either previously, currently or will be in the delivery zone.

Without loss of generality, let  $Y$  be the closest (among the nodes that have received the packet) in terms of hops to  $X$  on the face under consideration. If  $Y$  finds  $X$  was previously in the delivery zone or is currently in the delivery zone, it will do a local broadcast as soon as possible according to the FAR protocol. Note that one of  $Y$ 's direct neighbor is closer to  $X$  in terms of hops than  $Y$  is (e.g., node  $Z$  in Fig. 6). As a result, when the neighbors of  $Y$  hear the packet, the packet has at least

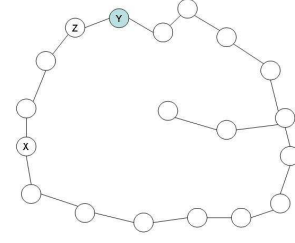


Fig. 6. FAR on a Face

moved one step closer to  $X$ . The same argument applies to the closer neighbor ( $Z$ ). The mobicast packet moves a node closer to  $X$  in each step, until the distance is zero, when  $X$  receives the packet.

If  $Y$  finds that the delivery zone will reach  $X$  some time in the future, it will schedule a forwarding at the appropriate time according to the FAR protocol. The same “one step closer” argument applies. ■

Using Lemma 1 we can prove the following theorem regarding the FAR protocol.

*Theorem 1:* In a connected network, FAR guarantees that all delivery zone nodes will receive the mobicast message if the initial delivery zone contains the source node.

**Proof:** We prove the theorem by contradiction. Let  $B$  be a delivery zone node that missed the packet. Being a delivery zone node,  $B$  must be located inside the integral delivery zone (the union of all delivery zone areas over the packet's lifetime), as shown in Fig. 7 in which the long dashed rectangle represents the integral delivery zone. Let  $A$  be the source node. Let  $X_1, X_2, \dots, X_k$  be the set intersection points between the line segment  $\overline{AB}$  and the graph edges, in order from  $A$  to  $B$ .

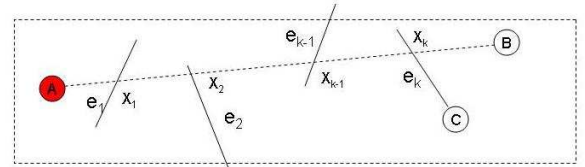


Fig. 7. Delivery Accuracy of the FAR protocol

If  $B$  missed the packet, none of the two end points of edge  $e_k$  would have received the packet. Otherwise, by Lemma 1,  $B$  should receive the packet because  $e_k$  and  $B$  are around the same face (they are around the same face because there is no edge between  $X_k$  and  $B$ ). Note also that at least one of the endpoints of the edge  $e_k$  is in the delivery zone because the height of the integral delivery zone is equal to the perpendicular span of the

delivery zone, which is assumed to be larger than the edge length. Let this end point of  $e_k$  be  $C$ .  $C$  being a delivery zone node that missed the message leads to the same argument that none of the endpoints on edge  $e_{k-1}$ , in turn  $e_{k-2}, \dots, e_2, e_1$ , has received the message. Yet,  $e_1$  and  $A$  are around the same face, and by Lemma 1, this is not possible, because, as the source node,  $A$  must have the message, so the message would have traversed  $e_1$ . ■

The FAR algorithm assumes all nodes have locally accessible information about their spatial neighbors. An important question is: how big is the spatial neighborhood in general? The answer to this question will shed light on the question of how much memory and storage the algorithm needs, which is very important in protocol and system design. Another important question is: how big is the average face size? The answer to this question relates to the forwarding overhead of the FAR protocol. We address these issues in the next section.

#### IV. FAR COST ANALYSIS

In this section we explore two cost metrics of FAR: (1) the memory space needed for the spatial neighborhood information, and (2) the communication overhead due to the traversing of face nodes that are not in the delivery zone. We start from an investigation of the average face size, average node degree on planar graphs and average spatial neighborhood size via geometric analysis, and conclude with simulation results from random networks.

##### A. Spatial Neighborhood Size

1) *Average Face Size*: The size of a face is defined by the number of vertices surrounding the face. The following theorem states a bound on the average size of faces on the planar graph.

*Theorem 2*: Given a planar graph  $G(V, E)$ , the average size of a face is

$$\overline{S}_f \leq \frac{2n_e}{n_f} \quad (2)$$

where  $n_e$  and  $n_f$  are the numbers of edges and faces of  $G$  respectively.

**Proof**: Let  $s_1, s_2, \dots, s_k$  be the sizes of all the faces of graph  $G$ . We have  $k = n_f$  and the total number of edges on all the faces is

$$s_1 + s_2 + \dots + s_k \leq 2n_e \quad (3)$$

the 2 appears in the equation because each edge is counted at most twice (once on each side). Note that dangling edges are counted only once, resulting in an inequality rather than an equality expression.

The average number of edges on each face is

$$\overline{S}_f = \frac{s_1 + s_2 + \dots + s_k}{k} \quad (4)$$

Combine it with inequality 3, we immediately have

$$\overline{S}_f \leq \frac{2n_e}{n_f}$$

■

Next we derive a bound for  $S_f$  in terms of the number of nodes and edges rather than edges and faces. This is more desirable because it is straightforward to count the number of nodes and edges in a graph and it is not very obvious to count the number of faces.

*Corollary 1*: Given a planar graph  $G(V, E)$ , the average size of a face is

$$\overline{S}_f \leq \frac{2n_e}{n_e - n_v + 2} \quad (5)$$

where  $n_v$  and  $n_e$  are the numbers of nodes and edges of  $G$  respectively.

**Proof**: From Euler's formula[8], we have the following relation between nodes, edges, and faces of any planar graph:

$$n_f + n_v - n_e = 2 \quad (6)$$

Use Theorem 2 and the Euler's formula, we get

$$\overline{S}_f \leq \frac{2n_e}{n_e - n_v + 2}$$

■

2) *Average Node (Face) Degree*: So far we have derived an upper bound for the average face size. Another question is how many faces each node has. Next lemma helps lead to an answer.

*Lemma 2*: On a planar graph  $G(V, E)$ , the edge degree of a node is always equal to a greater than its face degree. That is, let  $de_i$  be the edge degree of node  $i$ , and  $df_i$  be the face degree of node  $i$ . We have the following inequality

$$de_i \geq df_i \quad (7)$$

**Proof**: This is easy to prove. For each node  $i$ , sort its edges in clockwise or counter-clockwise order. There is at most one face between adjacent edges. Note that it is "at most" because of potential dangling edges which do not create new faces. ■

Using Lemma 2, we can derive the following theorem

*Theorem 3*: The average number of faces  $D_f$  each node has in a planar graph  $G(V, E)$  is upper bounded by the following expression

$$\overline{D}_f \leq 2 \frac{n_e}{n_v} \quad (8)$$

where  $n_v$  and  $n_e$  are the numbers of nodes and edges of  $G$  respectively.

**Proof:** Let  $de_i$  and  $df_i$  be the edge and face degrees of node  $i$  respectively. Then the sum of degrees across all nodes is

$$\sum_{i=1}^{n_v} de_i = 2n_e \quad (9)$$

because each edge is counted once on both ends.

From Lemma 2, we also have the sum of face degrees to be no greater than the sum of edge degrees

$$\sum_{i=1}^{n_v} df_i \leq \sum_{i=1}^{n_v} de_i \quad (10)$$

This leads to

$$\overline{D_f} \equiv \frac{\sum_{i=1}^{n_v} df_i}{n_v} \leq \frac{2n_e}{n_v} \quad (11)$$

■

3) *Average Spatial Neighborhood Size:* From Theorem 2 and Theorem 3, we may estimate the average spatial neighborhood size ( $\overline{\Upsilon}$ ) as follows.

Let  $\overline{D_f}$  be the average number of faces of each node, and  $\overline{S_f}$  be the average face size.  $\overline{S_f} * \overline{D_f}$  may be used for estimating the average number of nodes in all faces adjacent to each node if the variances in face sizes and node degrees are not high<sup>3</sup>. This leads to

$$\overline{\Upsilon} \sim \frac{4n_e^2}{n_v(n_e - n_v + 2)}$$

Considering the double counting of nodes in adjacent faces, this estimation can be improved. The double

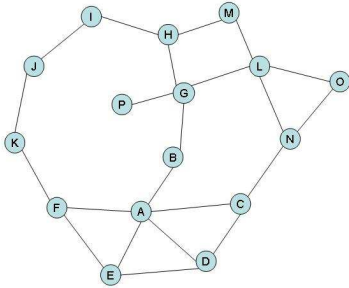


Fig. 8. Planar (Spatial) Neighborhood

counted nodes, say, with respect to node  $G$  in Fig. 8, include the following three kinds: (1) the node  $G$  itself, being counted twice (once on each adjacent face); (2) immediate double-faced neighbors of  $G$ :  $H, L, B$  (note that even though  $P$  is an immediate neighbor of  $G$ , it was not counted twice as it belongs to only one face);

<sup>3</sup>Note that  $mean(x_i y_i)$  does not equal to  $mean(x_i)mean(y_i)$  in general. But these two quantities have close values when all  $x_i$ 's are close to  $mean(x_i)$  and all  $y_i$ 's are close to  $mean(y_i)$ .

(3) non-immediate double-faced neighbors such as node  $A$ . We know that on average, the first kind of double-counting occurred  $\overline{D_f}$  times, and the second kind also occurred  $\overline{D_f}$  times. So there were at least  $2\overline{D_f}$  double counting of nodes in  $\overline{S_f}\overline{D_f}$ . This leads to

$$\begin{aligned} \overline{\Upsilon} &\sim (\overline{S_f} - 2)\overline{D_f} \\ &\sim \frac{4n_e(n_v - 2)}{n_v(n_e - n_v + 2)} \sim \frac{4n_e}{n_e - n_v + 2} \quad (12) \end{aligned}$$

Fig. 9 plots this estimation of spatial neighbor size against the relative edge to node ratio of a graph. We

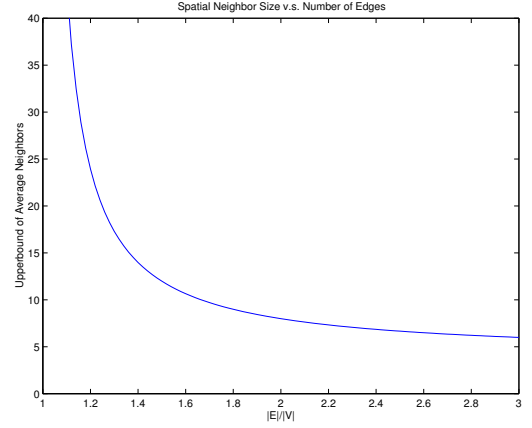


Fig. 9. Average spatial neighbor Size Estimation

can see that, given fixed number of nodes, more edges means smaller spatial neighborhood. In other words, the “denser” the graph is, the smaller its average spatial neighborhood is. Note that planar graphs have a limit on the number edges they can have. A well-known corollary of Euler’s formula states that for a planar graph, the number of possible edges is upper bounded by

$$n_e \leq 3n_v - 6 \quad (13)$$

Fig. 9 also suggests the the size is around 6 when  $n_e/n_v$  gets close to 3.

An important insight from this analysis is that for random ad hoc networks with uniform distribution, the average spatial neighborhood size is likely to be around the order of 10. As alluded to earlier, the closeness of this estimation depends on the variations on face sizes and node degrees of the planar network. This average case approximation is good only when the variances are small. These variances are likely to be relatively small in uniformly distributed networks. Next we test this observation via simulation.

### B. Statistical Face Size and Spatial Neighborhood Size Distribution of Planar Graphs

The goal of this section is to study the statistical distribution of face sizes in a planar graph. The statistical



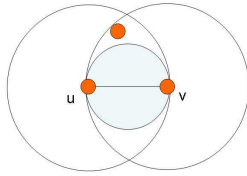


Fig. 10. A Gabriel Edge

information complements our previous average case results for estimating memory cost for our FAR mobicast protocol.

Note that ad hoc wireless networks are often not planar graphs. On the other hand, the FAR protocol uses the knowledge of spatial neighborhood defined on a planar graph. To let each node find out locally who its spatial neighbors are, we first need a method to planarize the network. It is well known that the Gabriel Graph (GG), the Relative Neighborhood Graph (RNG) [8][9] are planar graphs. In a geometric graph, an edge  $e = (u, v)$  is called a “Gabriel edge” if there is no other node inside the disk which uses  $e$  as a diameter. An example is in Fig. 10(a). A graph is a GG if it contains only Gabriel edges. Gabriel subgraphs of non-planar have been used in [10][6] for unicast geometric routing. A simple distributed algorithm can be found in both papers.

We use unit disk graph as an approximation for wireless ad hoc networks in our simulation. In an unit disk graph, two nodes have a common edge if and only if their Euclidean distance is less than a constant.

1) *Face and Spatial Neighbor Statistics:* For random unit disk graphs, we found the average face size of their Gabriel subgraph and the average spatial neighborhood size are both on the order of 10. Fig. 11 shows the face size distribution and Fig. 12 illustrates the spatial neighborhood size distribution obtained in our simulation<sup>4</sup>. The results shown in these figures were averaged over 8 random unit disk graphs. All unit disk graphs were generated in a 1000x1000 area with 1600 nodes and a communication range of 50, 25% greater than the critical range (40 in this setting) for a connected graph. In this case the average face size is about 5 and the average spatial neighborhood size of non-boundary nodes in the Gabriel subgraph stays very close to 19. These results also indicate that, on average, if we use the Gabriel subgraph of a wireless ad hoc network, the memory needed for the FAR algorithm is very low. Furthermore, we also found that the average number of

<sup>4</sup>In this figure, we eliminated the distribution related to the network “boundary” nodes, since they are not scale invariant and will be treated in different manner. More discussion on this in later sections.

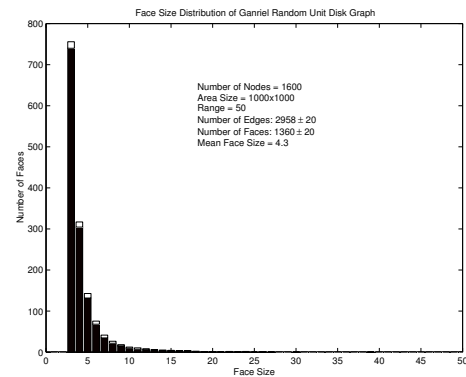


Fig. 11. Faces Size Distribution of Random Gabriel UDG

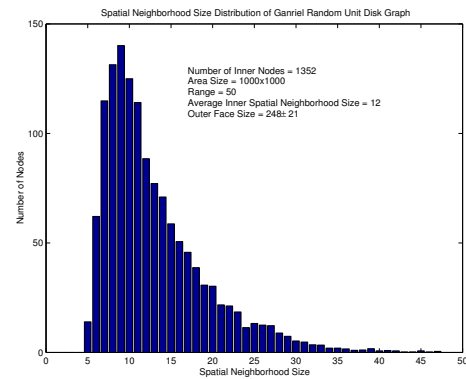


Fig. 12. Spatial Neighborhood Size Distribution of Random Gabriel UDG

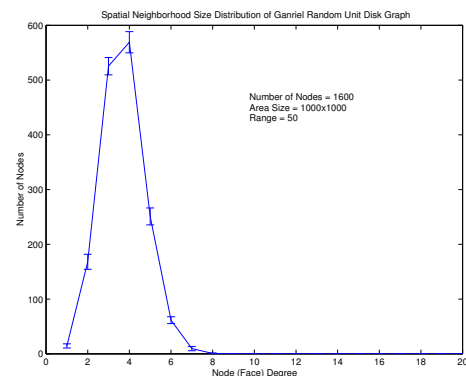


Fig. 13. Node Degree Distribution of Random Gabriel UDG

adjacent faces to a node is around 4 and does not vary much across the network. Fig. 13 shows the distribution of the number of adjacent faces to a node in the graph. These results also suggest our earlier observation about the spatial neighborhood size to be valid.

Furthermore, we observe that when node density increases from the critical (connectivity) density (about 8 network neighbors per node in our experiments), the average face size quickly decreases, as shown in Fig. 14. When the average number of network neighbors is beyond 14, the average number of spatial neighbors

is smaller. This suggests in such cases most spatial neighbors of a node are within one hop<sup>5</sup>. Face-aware forwarding is virtually reduced to local broadcast forwarding. The advantage of face-aware forwarding are expected to disappear from this point on, since there are few holes in high density networks.

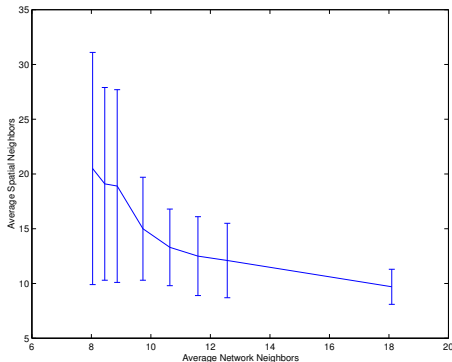


Fig. 14. Spatial Neighborhood Size and Network Neighbor Size

### C. FAR Message Overhead

The FAR protocol propagates the message on all faces that are inside or intersecting the path of the delivery zone. Its overhead can be measured by the number of non-delivery-zone nodes traversed per delivery-zone node delivery. Fig. 15 shows our preliminary simulation results of this delivery cost on uniformly distributed random networks of 1600 nodes in a 1000x1000 area. The mobicast setting is a rectangular delivery zone moving at a velocity of  $35m/sec$  for  $20sec$ . From Fig. 15 we can see that given a fixed delivery zone width (i.e., the size perpendicular to the velocity direction), FAR overhead decreases with the increase of node density (in terms of average number of network neighbors). This is reasonable since a smaller density means larger holes, and FAR adapts to it and uses more nodes for successfully routing around the holes. Note also that given a network density, the per node delivery cost decreases when the delivery zone path is wider, as a result of amortization effects.

## V. TOPOLOGY DISCOVERY

In this section we present a protocol for spatial neighborhood discovery. This protocol features a sorted ring-buffer assisted right-hand rule, a randomization strategy and a location-based tie-breaking rule. It used the following result of the Gabriel planarization as a starting

<sup>5</sup>Note that direct neighbors are not necessarily spatial neighbors, because some edges are eliminated during the planarization of the graph.

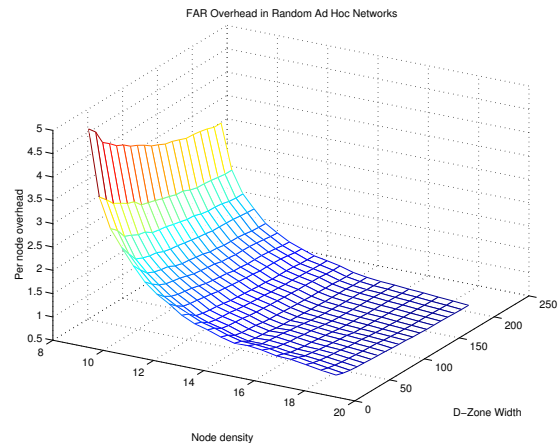


Fig. 15. FAR Protocol Pernode Delivery Overhead

point: each node  $v$  not only knows who their immediate network neighbors are, but also who among them are its immediate planar neighbors, defined as the set of nodes whose edges to the node  $v$  remain in the Gabriel subgraph of the original connectivity graph.

The protocol essentially creates a discovery message flow in each face, as shown in an example in Fig. 16. As a discovery message traverses a face, the coordinates

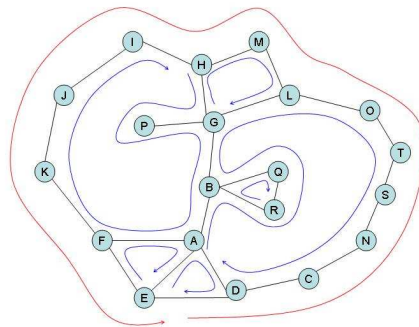


Fig. 16. Right-hand Neighborhood Discovery Protocol

of the nodes it has traversed are added to the message. After a discovery message finishes traversing a face, all nodes' locations on the face are collected and a message traverses the same face another time to inform everyone on the face of the complete discovery results.

There are four key problems that such a protocol needs to address: (1) Identification: how to make each discovery message traverse the correct face; (2) Termination: how to determine when a message has traversed the whole face; (3) Cost minimization: how to coordinate between nodes such that only one discovery message flows around each face; (4) Outer face limitation: the size of the outface is proportional to  $\sqrt{N}$ , where  $N$  is the total number of nodes in the network. When the network is very large, it is not feasible and not

reasonable to traverse this face, since a node shouldn't really concern itself with nodes on the other side of the network boundary.

We solve the first problem by using a ring-buffer on each node for storing the incident planar edges. The edges are directed (all viewed as outgoing edges from the node under consideration) and are sorted counter clockwise. When a discovery message comes from one edge, it will be sent on the next edge in the ring-buffer. Each discovery message contains the next hop location and an ordered list of visited nodes' locations, so it can be used to identify the incoming edge and designate the outgoing edge. This simple direction sorted ring-buffer enables each node to always choose the right outgoing edge for each discovery message, and in such a way make a message traverse a face correctly.

Upon receiving a discovery message  $dm$ , a node determines if  $dm$  has completed a full traversal of a face by the following criterion: the outgoing edge for  $dm$  is contained in its ordered traversal list. Note that a node can be traversed many times via a right-hand walk on a face. In turn, a simple termination rule such as "when the message come back to a node already traversed" does not work. For instance, in Fig. 16, node  $G$  is traversed twice on the  $\dots-H-G-P-G-B\dots$  face, and  $B$  is also traversed twice on the  $\dots-A-B-R-Q-B-G\dots$  face. Note also that the edges should be viewed as directed edges, e.g., edge  $G-P$  and edge  $P-G$  should be viewed as different edges. If  $P$  gets a discovery message that contains a  $G$  in the message's ordered traversal list, it should not necessarily think that the edge  $P-G$  has been traversed by the message.

The cost of the discovery protocol will be unnecessarily high if every node has its own discovery message flowing on each face. On each face, ideally one traversing discovery message will suffice. The problem is, some kind of leader election mechanism is needed for each face to determine who should initiate the discovery message. However, leader election is not possible before the members are known.

We use two strategies for reducing the number of discovery messages. First, we use a random starting time to reduce the number of messages initiated on each face. On each node, an initial discovery message  $dm_i$  is scheduled at a random time for each of its faces  $f_i$ . The initial discovery message contains the next hop location and a list containing only the sender location. The initial scheduled discovery message  $dm_i$  will not be sent if the node receives a discovery message  $dm$  from its neighbor regarding the same face before  $dm_i$ 's scheduled sending time. When this happens, the node simply appends itself to the ordered list in  $dm$ , resets the

next hop destination in the message, and forwards it. This randomization method can eliminate some but not all unnecessary discovery message initiations. For instance, in Fig. 16,  $A$ ,  $L$  and  $N$  may have all sent their discovery message for the same face (before receiving any from their neighbors). A tie-breaking strategy is needed to reliably reduce the messages to one. We use a starting location based tie-breaking rule: east is preferred, if there is still a tie, north is preferred. That is, if a node receives a discovery message initiated by others on the same face on which it has sent one, it will forward the message only if the initiator of this message is located east of him; if they are on the same east location (i.e., have the same x-coordinate), then only if the initiator is located north from it. When no two nodes have the same coordinates, this rule can uniquely identify one legitimate initiator and make each face have only a single discovery message remaining.

The outer face problem is hard since there is no way to determine which face is the "outer" one without a global bird's eye view. The outer face and the inner faces are topologically indistinguishable. To see this more clearly, imagine the graph is on the surface of a sphere instead of a plane. The only way to identify the "outer" face is its size. This leads to our solution: a discovery message has a max hop count. If it reaches its hop limit, a flag is set and it will traverse back to the originator. By doing this, every "boundary node" learns a limited amount of spatial neighborhood information on the outer face. Obviously, this strategy also leads to a potentially incomplete traversal in any "inner" face that is large. The existence of a better strategy is an open question.

## VI. DISCUSSION AND MORE RELATED WORK

Mobicast has a spatial multicast component similar to geocast, a multicast paradigm proposed by Navas and Imielinski [11]. In a geocast protocol, the multicast group members are determined by their physical locations. The initiator of a geocast specifies a fixed area for a message to be delivered, and the geocast protocol tries to deliver the message only to the nodes in that area. Ko and Vaidya [12] investigated geocast in the context of mobile ad hoc networks. Other mechanisms ([13], [14], [15]) have been proposed to improve geocast efficiency and delivery accuracy in wireless ad hoc networks. Mobicast differentiates itself from geocast by a mobile delivery area rather than a fixed one, and gives application developer a powerful tool for controlling information dissemination in the spatiotemporal domain rather than just the spatial domain. As a mobicast protocol, FAR uses face routing to achieve high space delivery guarantee

and uses timed forwarding for controlling information propagation speed.

The FAR protocol relies on the notion of spatial neighborhoods, and a smaller spatial neighborhood means that less memory is needed. This suggests that our protocol desires a planar graph with as many edges as possible. Given a non-planar graph, how to find its maximal planar subgraph is an active research subject. Recently Li *et al.* [16] proposed a localized Delaunay graph *LDel* which is denser compared to the Gabriel graph. Some other pointers to related research on maximal planarization can be found in [17]. Space limitations forced us to omit a discussion about optimization issues relating to the FAR protocol as they are not essential to understanding our contributions.

## VII. CONCLUSION

In this paper we presented FAR, a new face-aware mobicast routing protocol which, in theory, reliably delivers mobicast message spatially and has good mobicast temporal characteristics. This protocol relies on the notion of spatial neighborhoods and features a novel timed face-aware forwarding method. Since mobicast belongs to a new spatiotemporal multicast paradigm we proposed and there exists no close protocol for interesting and fair quantitative comparison, we focus on analyzing the qualitative perspectives of this protocol such as theoretical delivery accuracy, protocol cost and optimization opportunities. Besides proving that the FAR protocol achieves reliable spatial delivery, we estimated the size of its routing table in random wireless ad hoc networks via geometric analysis, and found that it is on the order of 10 entries. The latter is verified by statistical study of spatial neighborhood sizes on planar graphs. Furthermore, we also presented a novel spatial neighborhood discovery protocol and addressed key issues a spatial neighborhood discovery protocol must face, such as face identification, discovery termination, and duplicate elimination. Besides the novel merits of the FAR protocol and the spatial neighborhood discovery protocol, we believe this study, especially the proven theorems and the insight gained from statistical study about spatial neighborhood properties, helps to build a solid foundation for spatiotemporal protocol analysis in wireless ad hoc networks.

## ACKNOWLEDGEMENTS

This research was supported in part by the Office of Naval Research under MURI research contract N00014-02-1-0715. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of

the authors and do not necessarily reflect the views of the research sponsors.

## REFERENCES

- [1] D. Estrin et al., "Embedded everywhere: A research agenda for networked systems of embedded computers," National Academy Press, 2001, Computer Science and Telecommunications Board (CSTB) Report.
- [2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, Costa Rica, April 2001*, 2001.
- [3] D. Li, K. Wong, Y.H. Hu, and A. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, March 2002.
- [4] Qingfeng Huang, Chenyang Lu, and Gruia-Catalin Roman, "Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints," in *Proc. of the 2nd International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, USA, April 2003, pp. 442–457.
- [5] Qingfeng Huang, Chenyang Lu, and Gruia-Catalin Roman, "Spatiotemporal multicast in sensor networks," to appear in *ACM SenSys '03*, November 2003.
- [6] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, 2000, pp. 243–254.
- [7] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger, "Geometric Ad-Hoc Routing: Of Theory and Practice," in *Proc. 22<sup>nd</sup> ACM Int. Symposium on the Principles of Distributed Computing (PODC)*, 2003.
- [8] Mark de Berg, Marc van Kerveld, Mark Overmars, and Otfried Schwarzkopf, *Computational Geometry*, Springer, 1998.
- [9] J.W. Jaromczyk and G.T. Toussaint, "Relative neighborhood graphs and their relatives," *Proc. of IEEE*, vol. 80, no. 9, pp. 1502–1517, 1992.
- [10] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, pp. 609–616, 2001.
- [11] Julio C. Navas and Tomasz Imielinski, "Geocast - geographic addressing and routing," in *Proc. of the 3rd Annual Intl. Conf. on Mobile Computing and Networking (MobiCom)*, 1997, pp. 66–76.
- [12] Y. Ko and N. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, LA, February 1999, pp. 101–110.
- [13] I Stojmenovic, "Voronoi diagram and convex hull based geocasting and routing in wireless networks," TR TR-99-11, University of Ottawa, December 1999.
- [14] W.-H. Liao, Y.-C. Tseng, K.-L. Lo, and J.-P. Sheu, "Geogrid: A geocasting protocol for mobile ad hoc networks based on grid," *Journal of Internet Technology*, vol. 1, no. 2, pp. 23–32, 2000.
- [15] J. Boleng, T. Camp, and V. Tolety, "Mesh-based geocast routing protocols in an ad hoc network," in *Proc. of the IEEE Intl. Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (IPDPS)*, April 2001, pp. 184–193.
- [16] Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *Proceedings of IEEE INFOCOM*, 2003.
- [17] Annegret Liebers, "Planarizing graphs - a survey and annotated bibliography," *Journal of Graph Algorithms and Applications*, vol. 5, no. 1, pp. 1–74, 2001.