Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering Research

Computer Science and Engineering

Report Number: WUCSE-2004-15

2004-04-05

# FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks

Qingfeng Huang, Sangeeta Bhattacharya, Chenyang Lu, and Gruia-Catalin Roman

This paper presents FAR, a Face-Aware Routing protocol for mobicast, a spatiotemporal variant of multicast tailored for sensor networks with environmental mobility. FAR features face-routing and timed-forwarding for delivering a message to a mobile delivery zone. Both analytical and statistical results show that, FAR achieves reliable and just-in-time mes-sage delivery with only moderate communication and memory overhead. This paper also presents a novel distributed algorithm for spatial neighborhood discovery for FAR boot-strapping. The spatiotemporal performance and reliability of FAR are demonstrated via ns-2 simulations.

... Read complete abstract on page 2.

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research

Recommended Citation

Huang, Qingfeng; Bhattacharya, Sangeeta; Lu, Chenyang; and Roman, Gruia-Catalin, "FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks" Report Number: WUCSE-2004-15 (2004). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/987

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

This technical report is available at Washington University Open Scholarship: https://openscholarship.wustl.edu/cse_research/987

# FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks

Qingfeng Huang, Sangeeta Bhattacharya, Chenyang Lu, and Gruia-Catalin Roman

## Complete Abstract:

This paper presents FAR, a Face-Aware Routing protocol for mobicast, a spatiotemporal variant of multicast tailored for sensor networks with environmental mobility. FAR features face-routing and timed-forwarding for delivering a message to a mobile delivery zone. Both analytical and statistical results show that, FAR achieves reliable and just-in-time mes-sage delivery with only moderate communication and memory overhead. This paper also presents a novel distributed algorithm for spatial neighborhood discovery for FAR boot-strapping. The spatiotemporal performance and reliability of FAR are demonstrated via ns-2 simulations.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2004-15

# FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks

Authors: Huang, Qingfeng; Bhattacharya, Sangeeta; Lu, Chenyang; Roman, Gruia-Catalin

April 5, 2004

Abstract: This paper presents FAR, a Face-Aware Routing protocol for mobicast, a spatiotemporal variant of multicast tailored for sensor networks with environmental mobility. FAR features face-routing and timed-forwarding for delivering a message to a mobile delivery zone. Both analytical and statistical results show that, FAR achieves reliable and just-in-time message delivery with only moderate communication and memory overhead. This paper also presents a novel distributed algorithm for spatial neighborhood discovery for FAR bootstrapping. The spatiotemporal performance and reliability of FAR are demonstrated via ns-2 simulations.

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160

# FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks

Qingfeng Huang, Sangeeta Bhattacharya, Chenyang Lu, and
Gruia-Catalin Roman

WUCSE-2004-15

April 5, 2004

Department of Computer Science and Engineering
Campus Box 1045
Washington University
One Brookings Drive
St. Louis, MO 63130-4899

## Abstract

This paper presents FAR, a Face-Aware Routing protocol for mobicast, a spatiotemporal variant of multicast tailored for sensor networks with environmental mobility. FAR features face-routing and timed-forwarding for delivering a message to a mobile delivery zone. Both analytical and statistical results show that, FAR achieves reliable and just-in-time message delivery with only moderate communication and memory overhead. This paper also presents a novel distributed algorithm for spatial neighborhood discovery for FAR bootstrapping. The spatiotemporal performance and reliability of FAR are demonstrated via ns-2 simulations.

# FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks

Qingfeng Huang
Palo Alto Research Center (PARC) Inc.
3333 Coyote Hill Road, Palo Alto, CA 94304, USA.
Email:qingfeng@ieee.org
Sangeeta Bhattacharya, Chenyang Lu, and Gruia-Catalin Roman
Department of Computer Science and Engineering
Washington University, Saint Louis, MO 63130.
{sangbhat,lu,roman}@cse.wustl.edu

**Abstract**

This paper presents FAR, a Face-Aware Routing protocol for mobicast, a spatiotemporal variant of multicast tailored for sensor networks with environmental mobility. FAR features face-routing and timed-forwarding for delivering a message to a mobile delivery zone. Both analytical and statistical results show that, FAR achieves reliable and just-in-time message delivery with only moderate communication and memory overhead. This paper also presents a novel distributed algorithm for spatial neighborhood discovery for FAR bootstrapping. The spatiotemporal performance and reliability of FAR are demonstrated via ns-2 simulations.

## 1. Introduction

Wireless sensor networks are large-scale distributed embedded systems composed of small devices that integrate sensors, wireless communication interfaces, microprocessors, and maybe some actuators. 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 are expected serve as a key infrastructure for a broad range of applications including precision agriculture, surveillance, intelligent highway systems, emergent disaster response and recovery[4]. Many sensor network applications have fundamental spatiotemporal constraints that do not exist in traditional applications of wireless ad hoc networks. Fig. 1 shows two such examples. Fig. 1(a) is an intruder tracking application example. A set of sensors discovers an enemy tank, they send an alert message to sensors and actuators (e.g., camera control units) on the intruder's expected path under the need to wake them up, alert them, or pre-arm them for better tracking and actions. In other words, there is a need for the alert message to be disseminate to an envelop that moves at certain distance ahead of the intruder, with a speed approximating that of the intruder's, thus creating an evolving alert "cloud" just in front of it. Fig. 1(b) depicts an information scouting example. A solider is running to the southeast area. For safety and/or action efficiency, he would like to know the field information ahead on his path, so as to adjust his actions accordingly. His area of interest changes in front of him as he runs. Only the sensors that enter his "interested area" need to receive the scouting message and to pool their currently sensed information and send aggregated data back to him. As the soldier moves, the area of interest moves as well. This again demonstrate a need for establishing a moving

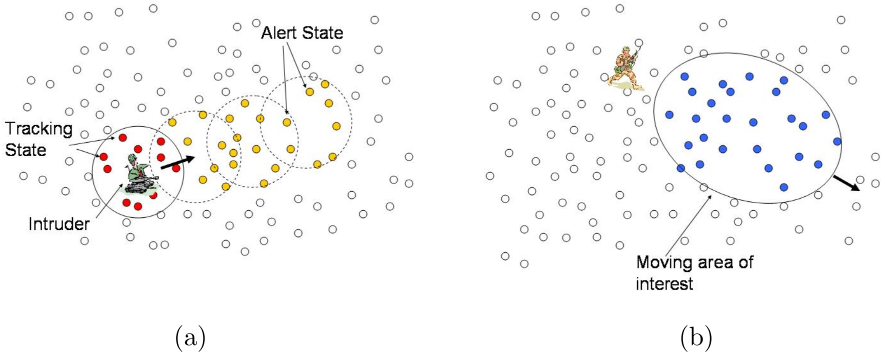


Figure 1: Tracking and Scouting Applications

area of awareness. Note that if the mobile event does not change its motion very often, the system alert or the scouting message does not need to be issued all the time. But rather, one can let the message roll on its own according to a motion plan for some period of time.

Mobicast [7][8] represents a new information dissemination paradigm with spatiotemporal semantics tailored for sensor network applications. Mobicast protocols allow 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. 2 shows such an example. More complex mobility models (with changing velocities) can be approximated by a sequence of constant-velocity mobicast sessions. Mobicast provides a powerful communication abstraction for data dissemination, aggregation, and local coordination in sensor networks. For example, the group

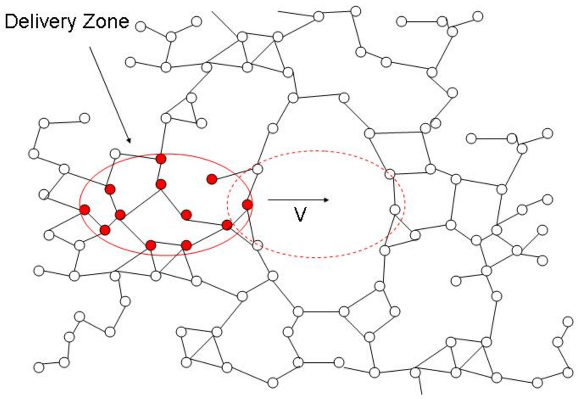


Figure 2: A Constant-Velocity Mobicast Example

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. Fig. 2 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 or 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 [7][8] has explored several different approaches. The first mobicast protocol presented in [7] 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 [8], 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 the delivery guarantees. However, the latter two approaches do not provide 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 the 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 advantages of the FAR protocol and the idea of mobicast are also demonstrated in the paper via simulation, by comparing with a geocast protocol and a greedy face routing protocol using three metrics: delivery ratio, delivery overhead, and delivery slack-time.

The remainder of the paper is organized as follows. Section 2 provides an overview of mobicast advantage and solutions. Section 3 describes the FAR mobicast protocol. Section 4 analyzes its delivery property. Section 5 investigates geometric properties of planar graphs related to the performance of FAR. A spatial neighborhood discovery protocol is presented in Section 6 . NS2 simulation results are presented in Section 7. Discussion, related work and conclusions are included in Sections 8 and 9.

# 2. Mobicast Overview

The mobicast service supports a type of application information delivery request that can be characterized by a delivery zone that changes over time. The key characteristic of the mobicast service is the explicit control over both the spatial and temporal perspectives of information delivery. This provides natural support for information dissemination tasks exhibiting "right-place and right-time" semantics, including the "just-in-time" requirement.

## 2.1. Limitation of Approaches Based on Geocast

To see the advantage of mobicast more clearly, we construct a solution for the discovery request in the information scouting problem using conventional multicast solutions such as geocast. This will illustrate a fundamental limitation of geocast-based solutions for this problem. Geocast is a reasonable match in the existing communication mechanism arsenal for supporting the application examples above. For simplicity, we use a rectangular delivery area in the example. Let the rectangle between points $A$ and $C$ be the initial geocast area for the discovery request when the soldier is at point $A$. As the soldier moves forward, his/her desired awareness area moves forward as well. So the soldier should periodically re-geocast the discovery request. Clearly, the period should be small enough such that there is at least one re-geocast between points $A$ and $C$ (before the soldier passes point $C$). Otherwise, a geocast area gap is created beyond point $C$ and some sensors close to $C$ on the right side will not get the request, and in turn the soldier might miss critical information. Let's assume that the soldier needs to re-issue the request at point $B$ at distance $W$ away from $C$. The choice of $W > 0$ is to reserve enough time for query processing, e.g., for making sure that once the soldier reaches $C$, all nodes between $C$ and $D$ have finished processing and answered the query. In this way one ensures that the soldier is aware of critical information in the area between $C$ and $D$ when he/she reaches $C$. If sensors have a sleeping schedule to conserve energy, the soldier should take that into account and make $W$ larger. Furthermore, a higher travelling speed for the soldier should result in a larger $W$. Note that re-geocasting the request at point $B$ for the area between $C$ and $D$ means that some of the nodes between point $B$ and point $C$ receive the information at least twice and they also need to act as routers for the request for nodes beyond point $C$. Clearly, the smallest number of such nodes is roughly[1] proportional to $W$, so is the number

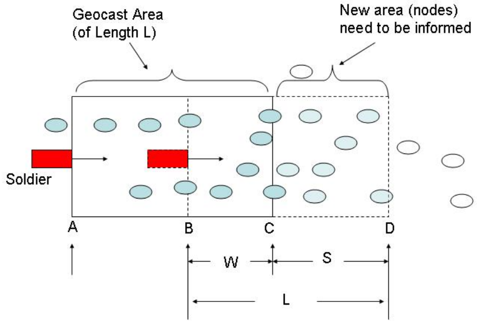


Figure 3: An Example of Geocast Based Solutions

[1] "Roughly" in the sense that we omit the discreteness effect which occurs when the size of the area is close to the radio range. Note also that we assume a constant height of the geocast area in the analysis.

of radio transmissions involved, regardless of the actual forwarding scheme used. This means that the extra routing overhead of this solution, defined as the number of extra radio transmissions per delivery, is about

$$M_W \sim \frac{W}{L-W} \tag{1}$$

In this scheme, the soldier re-geocasts at $B$ for the goal of having the sensors close to $C$ on the right receive the message $t_a = W/v_a$ time in advance, where $v_a$ is the speed of the soldier. This leads to the following consequence: most of the nodes between $C$ and $D$ receive the message at more than $t_a$ time in advance since geocast delivers messages in an as soon as possible fashion. Let $v_p$ be the maximum message (spatial) propagation speed in the network. Then the following quantity, called "average slack time," measures the average earliness of the nodes between $C$ and $D$ on receiving the message:

$$\overline{t_s} = \frac{1}{2}(\frac{S}{v_a} - \frac{S}{v_p}) = \frac{1}{2}(L-W)(\frac{1}{v_a} - \frac{1}{v_p}) \tag{2}$$

Usually a smaller average slack time is more desirable, e.g., in many real-time systems. Smaller average slack time in our example entails more nodes receiving the message just in time (to have just the right amount time for processing the request and replying), resulting in a system that is more flexible and robust against uncertainties and changes over time.

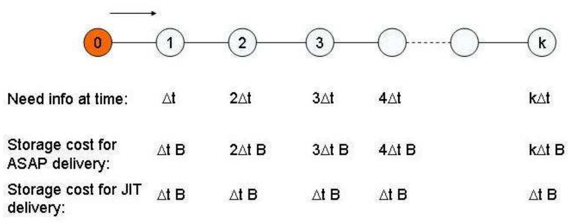
The average slack time decreases when $S$ increases, while the message overhead increases with $S$. As such, we observe a fundamental conflict in the geocast based approach: one cannot reduce the message overhead and the average slack time simultaneously. One reason behind this is that the geocast protocol is not explicitly concerned with the temporal domain of message delivery. That is, geocast only addresses an application's need for the "right place" perspective of information delivery, but does not address the "right time" perspective. Inevitably, exact steps and methods are necessary when using geocast protocols for constructing "just-in-time" type solutions, which in turn leads to extra overhead as we saw earlier.

We also observe that a more economic approach to this problem might be to let some nodes in the front of the geocast area forward the request[7] at an appropriate time rather than requiring the soldier to re-issue the geocast periodically. This approach entails a multicast paradigm with a mobile delivery area rather than a static one. This approach is exactly what our mobicast specification proposes to do [6].

## 2.2. Advantages of Just-in-Time Delivery

The just-in-time (JIT) delivery semantics inherent in mobicast has many advantages over conventional as-soon-as-possible (ASAP) delivery. One advantage of JIT delivery is that it has smaller network-wide storage-time footprint for the information being delivered than the ASAP deliver has, as shown in Fig. 4 in a simple 1-dimensional network. Storage-time of a piece of information on a node is defined as the amount of memory it takes times the amount of time it exist on the node. Assume node 0 has a piece of data ($B$ bytes) that nodes $1, 2, ..., k$ need at times $\Delta t$, $2\Delta t$,..., and $k\Delta t$, respectively. That is, the data is used only at the respective times at each node. In an ASAP delivery scheme, node 0 sends the data to the rest of the nodes and the data is received almost instantly (for simplicity, assuming the delivery latency is negligible). So the total storage-time for the data before it is being consumed is:

$$M_{ASAP} = \sum_{i=1}^{k} iB\Delta t = \frac{k(k-1)B\Delta t}{2} \tag{3}$$



| | | | | | |
|---|---|---|---|---|---|
| Need info at time: | Δt | 2Δt | 3Δt | 4Δt | kΔt |
| Storage cost for ASAP delivery: | Δt B | 2Δt B | 3Δt B | 4Δt B | kΔt B |
| Storage cost for JIT delivery: | Δt B | Δt B | Δt B | Δt B | Δt B |

Figure 4: A Just-in-Time Advantage

On the other hand, in a JIT delivery scheme, one can let each node hold the data for $\Delta t$ time before forwarding. This leads to a total storage-time of

$$M_{JIT} = \sum_{i=1}^{k} B\Delta t = kB\Delta t \tag{4}$$

One can see that the advantage of JIT over ASAP is dramatic in this example: a linear storage-time over a quadratic one.

## 2.3. Challenges

While mobicast is an interesting and useful abstraction for information dissemination in sensor network applications, implementation challenges are significant, especially when one desires high delivery guarantees. In this work, we focus on two major challenges that are specific to randomly distributed sensor networks.

First, in many scenarios, sensors are likely to be deployed in an ad hoc fashion, e.g., by dispersing them from airplanes. The topology of this type of sensor network would thus be rather random. More specifically, this type of network can contain "holes". Two nodes close in physical space can be relatively far away in logical network space (in terms in network hops). Fig. 5 shows an example of a random (yet connected) sensor network generated via uniform distribution of the $x$ and $y$ coordinates of the sensors. One can see there are many holes of varying sizes. The potential existence of holes in the network poses a challenge for mobicast. A mobicast session might be stopped prematurely because of a hole too big on its path. Furthermore, connectionless protocols are preferred for multicast in sensor networks due to their relatively low overhead. For a connectionless protocol, there is no way to reliably discover and inform the sender that the session has been stopped prematurely. In turn, the unannounced premature termination of the information propagation may adversely affect application semantics. Second, the mobicast delivery zone moves through the physical space.

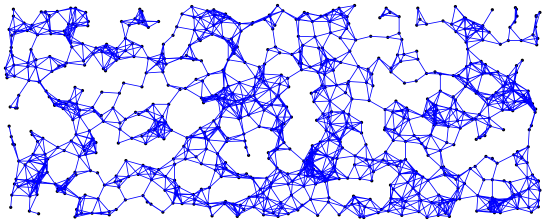


Figure 5: Random Disk Graph

As we pointed out earlier, two nodes close in physical space can be relatively far away in terms of network hops due to the presence of holes in the network. This presents a challenge to timely delivery of mobicast messages, i.e., a mobicast protocol needs to consider potential latency due to long underlying network paths, in order to achieve timeliness.

## 3. 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 Compass routing[13], FACE-2[2], GPSR [11] and GOAFR$^+$ [14]. 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 the geometric routing schemes. The location of the destination node is key in determining the forwarding path and in detecting whether the greedy algorithm entered a local minimum. In mobicast, however, there is no single destination location; only the delivery zone is known, and the exact location of nodes in future delivery zones are not known. Simple approaches such as first selecting some arbitrary location in the delivery zone path as a destination and then use geometric 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, it is impossible to know if a node exists at a particular location or if any node is 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 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.

### 3.1. 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 [3]. For instance, in the planar graph as shown in Fig. 6, 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 nodes", "inner face" and "outer face" components of a planar graph seem visually easy to identify, topologically it is hard to distinguish them. This has important consequences on face-based geometric routing mechanisms. We will discuss this in section V.

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. 6, 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 11 spatial neighbors ($L$, $H$, $B$ $I$, $J$, $K$, $F$, $P$, $C$, $M$ and $N$) while it only has three immediate graph

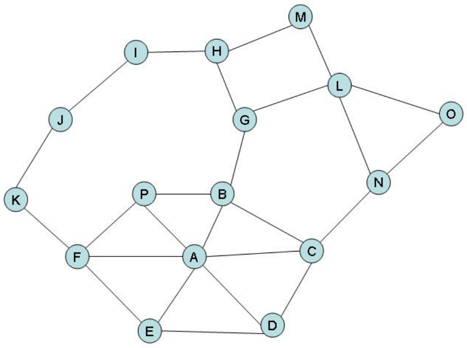


Figure 6: Planar Graph and Planar (Spatial) Neighborhood

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.

## 3.2. 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.[2]. We will prove that this simple rule can guarantee that all delivery zone nodes will 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 [8]. 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.

**3.2.1. 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[7][8], 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 time in the future. The message lifetime is

[2] 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.

used for terminating the mobicast session. The last forwarder information is used for determining if further forwarding of a packet is needed.

For simplicity, we assume each mobicast message fits in one packet, and we use the words packet and message interchangeably. We know explain the forwarding mechanisms in FAR.

**3.2.2. Greedy Forwarding.** Greedy forwarding applies to all nodes that are currently (or previously)covered by the mobicast delivery zone, or have at least one spatial neighbor that is currently (or previously) covered by the mobicast delivery zone. In such cases, a node forwards a new packet in an "as-soon-as-possible" fashion.

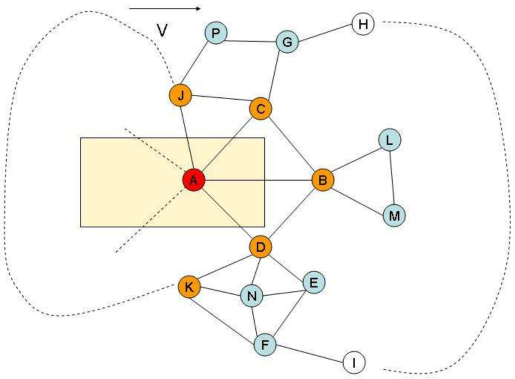


Figure 7: Greedy and Timed Face-Aware Forwarding

Fig. 7 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 in the future.

**3.2.3. 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 has at least one spatial neighbor that will be in the delivery zone. Nodes $H$, $G$, $L$, $M$, $E$, $F$ and $I$ in Fig. 7 belong to this category. Nodes $L$ and $M$ will be in the delivery zone themselves as the delivery zone moves to the right. Nodes $G$, $E$ and $F$ will find three of their spatial neighbors, $B$, $L$ and $M$ will 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 $Y_1, Y_2, ..., Y_k$ be the ordered list of all spatial neighbors of $X$ that will be in the delivery zone and $\Delta t_1, \Delta t_2, ..., \Delta t_k$ be the corresponding times for the delivery zone to reach them. Let $h_i$ be the hop distance from $X$ to $Y_i$. 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.,

$$T_a = min\{\Delta t_i - h_i\tau_1 | i = 1, 2, ..., k\} \tag{5}$$

The forwarding decision of $X$ is as follows:

1. If $T_a \leq 0$ forward the packet as soon as possible;
2. If $T_a > 0$ delay the forwarding for time length $T_a$.

In Fig. 7, 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 that 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 discover that fact. Node $G$ knows that 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 $E$.

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, $F$ cannot simply compute its forwarding time based on $B$, since $\Delta t_L - h_{FL}\tau_1$ may be smaller than $\Delta t_B - h_{FB}\tau_1$.

In the previous discussion, we choose $\tau_1$ to be the expected 1-hop latency. If one chose $\tau_1$ 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. 7, 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 $P$.

**3.2.4. Protocol Termination.** In addition to greedy forwarding and timed forwarding, the algorithm also has a mobicast termination method based on the packet lifetime 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.

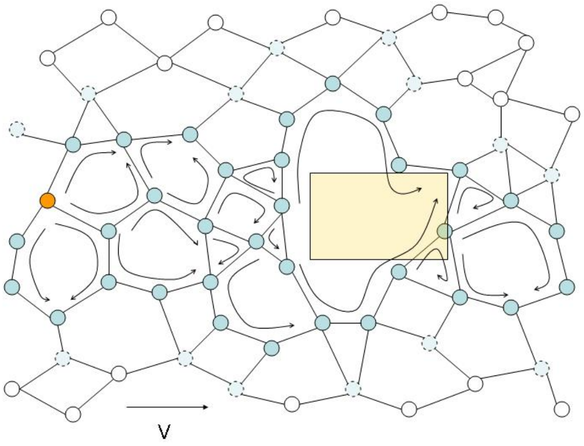


Figure 8: Bird's Eye View on the FAR Protocol Behavior and Result

To help see a bigger picture of the behavior and results of the FAR algorithm, Fig. 8 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 heard 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 along the delivery zone path and helps the delivery zone cross holes in the network.

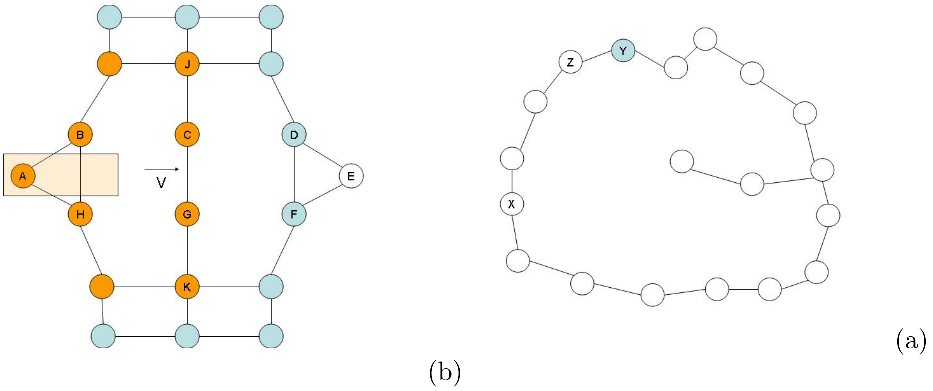
Next we prove that our forwarding strategy indeed delivers mobicast packets to all its expected recipients under one reasonable assumption.

# 4. 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 delivery zone: the delivery 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 perpendicular span is too small, the algorithm may terminate prematurely. Fig. 9(a) 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 delivery guarantee in the general case. We start from the following lemma.

Lemma 4.1. *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



(b)

(a)

Figure 9: (a) FAR Assumption (b) FAR on a Face

neighbors which include $X$. So $Y$ can compute whether $X$ was previously, is 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 neighbors is closer to $X$ in terms of hops than $Y$ is (e.g., node $Z$ in Fig. 9(b)). As a result, when the neighbors of $Y$ hear the packet, the packet has moved at least 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 4.1 we can prove the following theorem regarding the FAR protocol.

THEOREM 4.1. *In a connected network, FAR guarantees that all delivery zone nodes will receive the mobicast message (but not necessarily on time) 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. 10 in which the long dashed rectangle represents the integral delivery zone. Let $A$ be the source node. Let $X_1, X_2, ..., X_k$ be the set of intersection points between the line segment $\overline{AB}$ and the communication graph edges, in order from $A$ to $B$.

If $B$ missed the packet, none of the two end points of edge $e_k$ would have received the packet. Otherwise, by Lemma 4.1, $B$ 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 $C$. Since $C$ is a delivery zone node that missed the message, this leads to the same argument that none of the endpoints on edge $e_{k-1}$, and in turn $e_{k-2}, ..., e_2, e_1$, have received the message. Yet, $e_1$ and $A$ are around the same face, and by

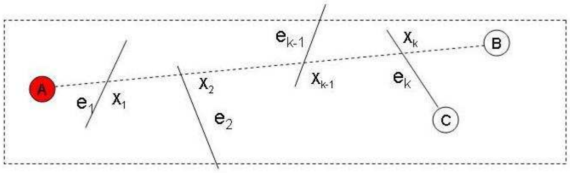


Figure 10: Delivery Accuracy of the FAR protocol

Lemma 4.1, 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.

# 5. 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.

## 5.1. Spatial Neighborhood Size

**5.1.1. 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 a planar graph.

THEOREM 5.1. *Given a planar graph $G(V, E)$, the average size of a face is*

$$\overline{S_f} \leq \frac{2n_e}{n_f} \tag{6}$$

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

**Proof:** Let $s_1, s_2, \cdots, s_k$ 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

$$s_1 + s_2 + \cdots + s_k \leq 2n_e \tag{7}$$

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

$$\overline{S_f} = \frac{s_1 + s_2 + \cdots + s_k}{k} \tag{8}$$

Combined with inequality 7, gives

$$\overline{S_f} \leq \frac{2n_e}{n_f}$$

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 how to count the number of faces.

Corollary 5.1. *Given a planar graph $G(V, E)$, the average size of a face is*

$$\overline{S_f} \leq \frac{2n_e}{n_e - n_v + 2} \tag{9}$$

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

**Proof:** From Euler's formula [3], we have the following relation between nodes, edges, and faces of any planar graph:

$$n_f + n_v - n_e = 2 \tag{10}$$

Use Theorem 5.1 and the Euler's formula, we get

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

**5.1.2. 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. The next lemma helps lead to an answer.

Lemma 5.1. *On a planar graph $G(V, E)$, the edge degree of a node is always equal to or 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*

$$de_i \geq df_i \tag{11}$$

**Proof:** 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 5.1, we can derive the following theorem

Theorem 5.2. *The average number of faces $D_f$ each node has in a planar graph $G(V, E)$ is upper bounded by the following expression*

$$\overline{D_f} \leq 2\frac{n_e}{n_v} \tag{12}$$

*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

$$\sum_{i=1}^{n_v} de_i = 2n_e \tag{13}$$

because each edge is counted once on both ends.

From Lemma 5.1, we also have the sum of face degrees to be no greater than the sum of edge degrees

$$\sum_{i=1}^{n_v} df_i \leq \sum_{i=1}^{n_v} de_i \tag{14}$$

This leads to

$$\overline{D_f} \equiv \frac{\sum_{i=1}^{n_v} df_i}{n_v} \leq \frac{2n_e}{n_v} \tag{15}$$

**5.1.3. Average Spatial Neighborhood Size.** From Theorem 5.1 and Theorem 5.2, 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[3]. 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 counted nodes, say, with respect to node $G$ in Fig. 11, include the following three kinds:

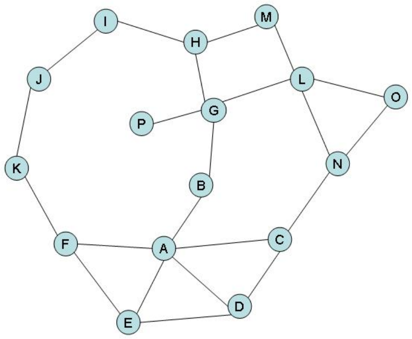


Figure 11: Planar (Spatial) Neighborhood

(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); (3) non-immediate double-faced neighbors such as node $A$. We

[3] Note that $mean(x_iy_i)$ 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 $mean(y_i)$.

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 $\overline{S_f D_f}$. This leads to

$$\begin{aligned}\overline{\Upsilon} &\sim (\overline{S_f} - 2)\overline{D_f} \\ &\sim \frac{4n_e(n_v - 2)}{n_v(n_e - n_v + 2)} \sim \frac{4n_e}{n_e - n_v + 2}\end{aligned} \tag{16}$$

Fig. 12 plots this estimation of spatial neighbor size against the relative edge to node ratio of a graph. We can see that, given a fixed number of nodes, more edges means a 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 has an upper bound

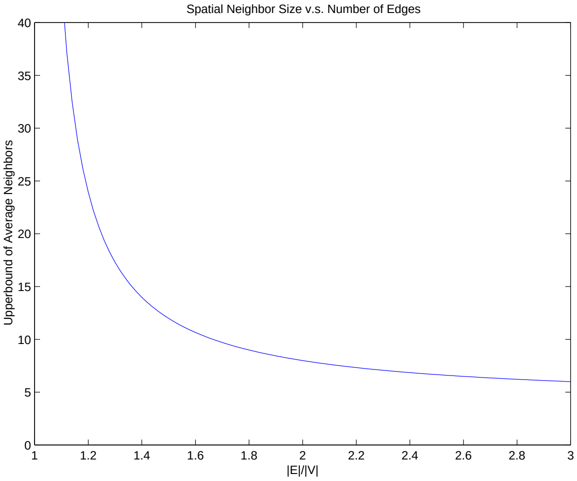


Figure 12: Average spatial neighbor Size Estimation

$$n_e \le 3n_v - 6 \tag{17}$$

Fig. 12 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 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.

## 5.2. Statistical Face Size and Spatial Neighborhood Size Distribution in Planar Graphs

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

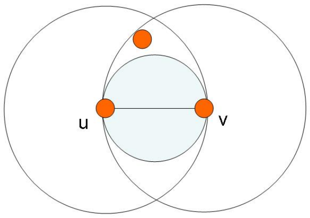
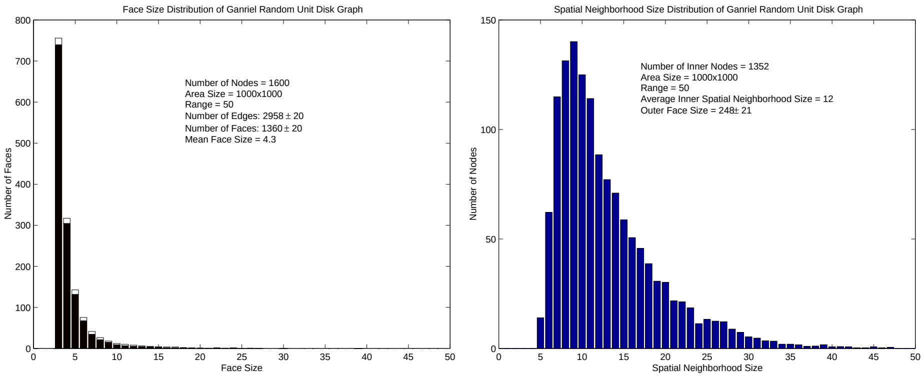


Figure 13: A Gabriel Edge

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)[5] and the Relative Neighborhood Graph (RNG) [3][10] 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. 13. A graph is a GG if it contains only Gabriel edges. Gabriel subgraphs of non-planar graphs have been used in [2][11] 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 for collecting geometric statistics. In a unit disk graph, two nodes have a common edge if and only if their Euclidean distance is less than a constant.

**5.2.1. 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 in the order of 10. Fig. 14(a) shows the face size distribution and Fig. 14(b) illustrates the spatial neighborhood size distribution obtained in our simulation[4].



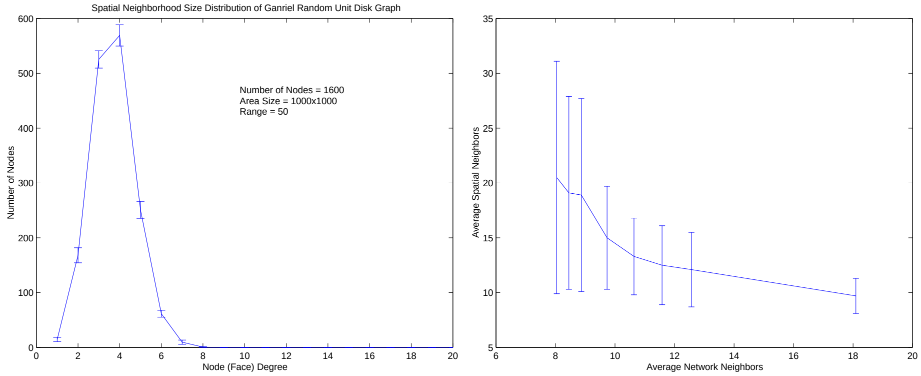
(a) (b)

Figure 14: (a) Faces Size Distribution in Gabriel Spanner of Random Unit Disk Graphs (b) Spatial Neighborhood Size Distribution in Gabriel Spanner of Random Unit Disk Graphs

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

[4] 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 is given in later sections.

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 the 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 adjacent faces to a node is around 4 and does not vary much across the network. Fig. 15(a) shows the distribution of the number of adjacent faces to a node in the graph. These results also suggest that our earlier observation about the spatial neighborhood size is valid.



(a) (b)

Figure 15: (a) Node Degree Distribution in Gabriel Spanner of Random Unit Disk Graphs (b) Spatial Neighborhood Size and Network Neighbor Size

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. 15(b). 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[5]. 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.

## 6. 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 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 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.

[5] Note that direct neighbors are not necessarily spatial neighbors, because some edges are eliminated during the planarization of the graph.

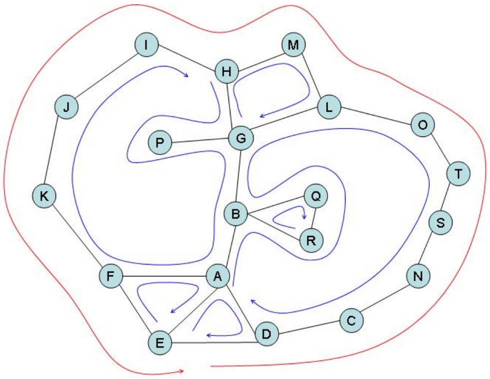


Figure 16: Right-hand Neighborhood Discovery Protocol

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.

## 6.1. Face Identification

We solve the face identification 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 clock-wise. 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.

## 6.2. Face Traversal Termination

A node determines if an incoming a discovery message $dm$ 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 $\cdots$-$H$-$G$-$P$-$G$-$B$-$\cdots$ face, and $B$ is also traversed twice on the $\cdots$-$A$-$B$-$R$-$Q$-$B$-$G$-$\cdots$ 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.

### 6.3. Cost Minimization

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. 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 itself; 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.

### 6.4. The Outer Face

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. A practical way to identify an "outer" face is from 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.

## 7. NS2 Simulation Results

We study and compare the performance of FAR with both Geocast and a scaled down version of FAR that incorporates only greedy forwarding, which we will henceforth refer to as the Greedy protocol. Our implementation of the Geocast protocol is as follows. A node that receives a message that isn't previously-seen, forwards the message only if it is in the delivery zone. Otherwise it drops the message. Since the originator of the message is inside the delivery zone, there is no requirement of first geocasting a message to the delivery zone from another point in space. This implementation of geocast confirms to the geocast protocol presented in [12]. Further, we use the topology discovery protocol presented in the previous section, to obtain spatial neighborhood information.

The simulation environment consists of 600 sensors dispersed randomly in a simulation area of 1000x400 $m^2$ under the uniform distribution. A rectangular delivery zone of size 120x100 $m^2$ is assumed to move at a constant velocity of 20 m/s during a Mobicast session of length 35 s. The three protocols are simulated for three different communication ranges of 50m, 75m and 100m. Varying the communication range allows us to test the variation in the performance of the protocols with network density.

We use two different metrics to compare the performance of the different protocols. The first metric is the *delivery ratio*, which is the fraction of sensor nodes in the delivery zone that receive the Mobicast message. The delivery ratio represents the spatial reliability of the protocol. The second metric is *slack time*. A positive slack time indicates that the deadline hasn't been missed. Hence, a positive and low slack time is desired, since lower slack time indicates lower storage-time cost, as shown in Section 2.2. FAR aims at reducing slack time and hence providing high temporal efficiency and low storage-time cost.

## 7.1. Spatial Reliability

| Protocol | FAR | Greedy | Geocast |
|---|---|---|---|
| Delivery ratio | $1 \pm 0$ | $1 \pm 0$ | $0.61 \pm 0.36$ |

Table 1: Average and standard deviation of delivery ratio of FAR, Greedy and Geocast.

Table 1 shows the delivery ratio obtained by the three protocols for a communication range of 50m. The values in the graph are an average over five runs on different network topologies. As we can see from the figure, both FAR and Greedy achieve a much higher spatial reliability than Geocast. In our simulations, FAR and Greedy achieve 100% spatial reliability while Geocast, on average, achieves only 61% spatial reliability. We also see from the figure, that Geocast displays a lot of variance. This variance and low average delivery ratio is caused by the presence of holes at different locations, in the path of the delivery zone. Unlike FAR and Greedy that route around holes with the help of spatial neighborhood information, Geocast does not support routing around holes. Hence, at low network densities, where the possibility of holes is high, FAR and Greedy greatly outperform Geocast. In simulation also we found that Geocast performs better at high network densities (communication ranges 75m and 100m), where all three protocols achieve 100% spatial reliability (graphs for these densities are not shown).

While in our limited simulation runs FAR and Greedy achieved 100% spatial delivery, it is to be noted that in certain cases FAR (and hence Greedy) may not achieve 100% spatial reliability. One reason for this is the inherent unreliability associated with wireless connectionless broadcast. Another reason is that, since our implementation uses the topology discovery protocol discussed in Section 6, there may be practical situations where the delivery zone path crosses the outer face or a large inner face (see Section 6.4). In this case, due to lack of complete spatial neighborhood information, the message may not reach certain nodes. However, the probability of this is quite low.

## 7.2. Slack time

The slack time obtained by a run of the three protocols for a communication range of 75m is shown in Fig. 17. We can see from this figure that FAR greatly reduces the slack time. This is achieved because of timed forwarding in FAR. Greedy and Geocast obtain similar slack times and the slack time increases with the increase in the distance of sensor nodes from the starting location of the delivery zone. The increase in slack time with increase in distance from the start point is due to the fact that both Greedy and Geocast implement as-soon-as-possible message forwarding.

Table 2 shows the average and standard deviation of the slack time obtained by the three protocols for the same run depicted in Fig. 17. This graph clearly shows the large variance in slack time

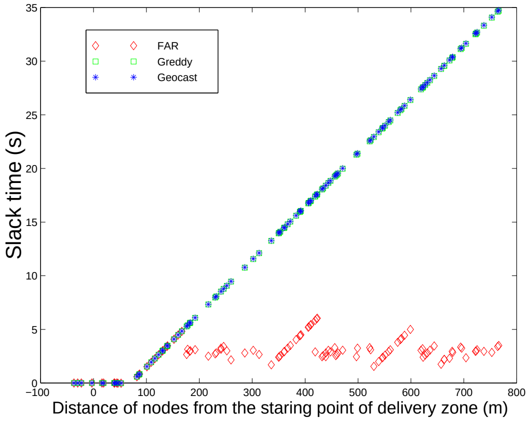


Figure 17: Slacktime obtained by FAR, Greedy and Geocast for a typical run with a communication range of 75m.

| Protocol | FAR | Greedy | Geocast |
|---|---|---|---|
| Slack time (s) | $2.8 \pm 1.4$ | $15.7 \pm 10.8$ | $15.7 \pm 10.8$ |

Table 2: Average and standard deviation of slacktime obtained by FAR, Greedy and Geocast for a typical run with a communication range of 75m.

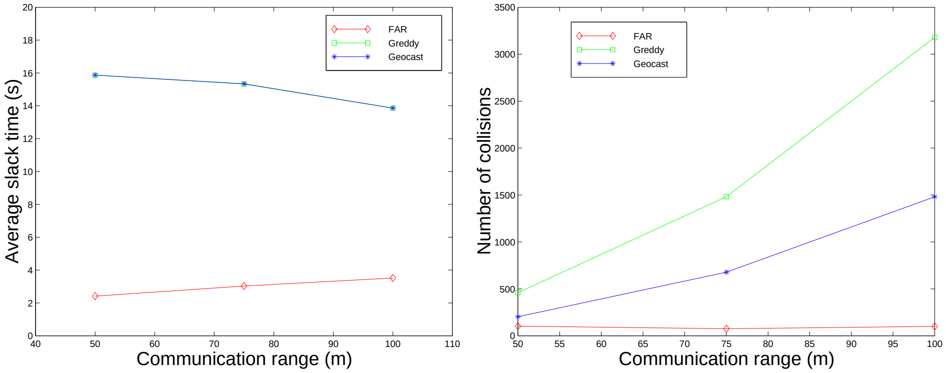
obtained by both Greedy and Geocast as against the small variance in slack time obtained by FAR. We also see the difference in the average slack time obtained by the three protocols.

Variance of slack time with density is shown in Fig. 18(a). Each point in this figure is an average over five runs. The graphs for the different runs were chosen at random with the constraint that Geocast achieves high spatial reliability in all of them. This constraint is required to carry out a fair slack time comparison of Geocast with the other protocols. From the figure, we see that the slack time obtained by Greedy and Geocast for different densities is the same (their lines overlap in the figure) and it decreases with density. This is caused due to the increase in the number of collisions with an increase in density. The number of collisions caused by the three protocols at different densities is shown in Fig. 18(b).

The slack time obtained by FAR, however, increases with density. The reason for this is that the communication range increases with increasing density and hence nodes tend to overhear packets from their neighbors before the time they are scheduled to receive the packets. This results in an increase in slack time. Since FAR uses timed forwarding, which reduces the number of overlapping broadcasts, the number of collisions in FAR when compared to the other two protocols is very small. This is visible in Fig. 18(b) and depicts another advantage of FAR over simple broadcasting protocols.

## 8. Discussion and More Related Work

Mobicast has a spatial multicast component similar to geocast, a multicast paradigm proposed by Navas and Imielinski [18]. 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



(a) (b)

Figure 18: (a) Variance of average slacktime obtained by FAR, Greedy and Geocast with increasing network density; (b)Number of collisions for the different protocols at different densities.

Vaidya [12] investigated geocast in the context of mobile ad hoc networks. Other mechanisms ([19, 16, 1, 20]) have been proposed to improve geocast efficiency and delivery accuracy in wireless ad hoc networks. Zhou and Singh proposed a content-based multicast [21] in which sensor event information is delivered to nodes in some geographic area determined by the velocity and type of the detected events. While different in style and approach, all these techniques assume the delivery zone to be fixed. They also assume the same information delivery semantics along the temporal domain, i.e., information is to be delivered "as soon as possible," and the issue of guaranteed geocast delivery in the case of a partitioned network inside the geocast area has not been extensively explored [20]. Mobicast differentiates itself from geocast by a mobile delivery area rather than a fixed one, and gives application developers a powerful tool for controlling information dissemination in both spatial and the temporal domains rather than just the spatial domain. As a mobicast protocol, FAR uses face routing to achieve a high spatial delivery guarantee and timed forwarding for controlling information propagation speed. The earliest work that is face routing related appears to be the compass routing [13], proposed for getting out of local minimum in stateless location-based geometric unicast routing strategies. The idea was further explored many other unicast routing and geocast routing work such as [2] and [11]. FAR is the first attempt to applying the face routing idea for the context of mobicast, and also augment the conventional face routing idea with with temporal control in message forwarding. Note that our simulation results show that the temporal control not only dramatically reduces the slack-time in message deliveries, but also dramatically reduces potential collisions in broadcast-type message forwarding.

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 [15] 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].

Note that mobicast as a information dissemination paradigm applies both static and mobile networks even though the FAR protocol and its analysis in this paper are applicable only to relatively static ad hoc network such as sensor networks. When the network is mobile and dynamic, such as a vehicle network, the topology changes relatively fast and the cost of constant updating topology information becomes significant. As a result, FAR might not be a good choice in such scenarios and new strategies in balancing reliability and cost are in demand. This clearly points to a new topic in future research for supporting reliable mobicast in ad hoc mobile networks.

## 9. Conclusion

In this paper we presented FAR, a new face-aware mobicast routing protocol which, in theory, reliably delivers 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 and there exists no close protocol for interesting and fair quantitative comparison, we focused on analyzing the qualitative perspectives of this protocol, e.g., 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 finding was verified by a 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 consider, such as face identification, discovery termination, and duplicate elimination. The cost and reliability of the FAR protocol is also studied via NS2 network simulation, and is compared to two other approaches. The result clearly show the advantages of FAR in achieving the reliability of spatial delivery and low slack-time with relatively low overhead. Besides the novelty of the FAR and spatial neighborhood discovery protocol, we believe that this study helps to build a solid foundation for spatiotemporal protocol analysis in wireless ad hoc networks.

## Acknowledgements

This research was supported in part by the National Science Foundation under Grant No. CCR-0325529 and 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. Qingfeng Huang would also like to thank Ivan Stojmenovic for information on related work. Furthermore, the authors would also like to note that this paper is an extension of our preliminary work that appeared in INFOCOM 2004 [9].

## References

[1] 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)*, pages 184–193, April 2001.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with gurranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7:609–616, 1999,2001. An earlier version appeared in ACM DIAL M, August 1999, 48-55.

[3] M. de Berg, M. van Kerveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry.* Springer, 1998.

[4] 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.

[5] K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *System Zoology*, 18:259–278, 1969.

[6] Q. Huang. *Spatiotemporal Multicast and Group Membership Service in Ad Hoc Networks.* PhD thesis, Washington University in Saint Louis, 2003.

[7] Q. Huang, C. Lu, and G.-C. 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*, pages 442–457, Palo Alto, CA, USA, April 2003.

[8] Q. Huang, C. Lu, and G.-C. Roman. Spatiotemporal multicast in sensor networks. In *Proceedings of the First ACM Conference on Embedded Networks Sensor Systems (SenSys)*, Los Angelos, California, Nov. 2003.

[9] Q. Huang, C. Lu, and G.-C. Roman. Reliable mobicast via face-aware routing. In *Proceedings of the 23rd Conference of IEEE Communication Society (INFOCOM)*, Hong Kong, China, March 2004.

[10] J. Jaromczyk and G. Toussaint. Relative neighborhood graphs and their relatives. *Proc. of IEEE*, 80(9):1502–1517, 1992.

[11] 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)*, pages 243–254, 2000.

[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)*, pages 101–110, New Orleans, LA, February 1999.

[13] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG'99)*, 1999.

[14] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric Ad-Hoc Routing: Of Theory and Practice". In *Proc. $22^{nd}$ ACM Int. Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[15] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *Proceedings of IEEE INFOCOM*, 2003.

[16] 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*, 1(2):23–32, 2000.

[17] A. Liebers. Planarizing graphs - a survey and annotated bibliography. *Journal of Graph Algorithms and Applications*, 5(1):1–74, 2001.

[18] J. C. Navas and T. Imielinski. Geocast - geographic addressing and routing. In *Proc. of the 3rd Annual Intl. Conf. on Mobile Computing and Networking (MobiCom)*, pages 66–76, 1997.

[19] I. Stojmenovic. Voronoi diagram and convex hull based geocasting and routing in wireless networks. TR TR-99-11, University of Ottawa, December 1999.

[20] I. Stojmenovic. geocasting with guaranteed delivery in sensor networks. In *International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Fort Lauderdale, Florida, Feb 20-21 2004.

[21] H. Zhou and S. Singh. Content based multicast (cbm) for ad hoc networks. Mobihoc 2000, Boston, MA, August 2000.