Cost-Conscious Geographic Multicast on MANET

Akira Mizumoto, Hirozumi Yamaguchi and Kenichi Taniguchi
Graduate School of Information Science and Technology, Osaka University
1-3 Machikaneyamacho, Toyonaka, Osaka 560-8531 Japan
{mizumoto, h-yamagu, taniguchi}@ist.osaka-u.ac.jp

***Abstract*— In this paper, we propose a location-aware multicast protocol on MANET called *MgCast (multiCAST for Multiple Geographical regions)*. Given a source node and a set of geographical destination regions, MgCast constructs and maintains a routing tree from the source to nodes which reside in the regions, in a decentralized manner. Our aim is to pursue trade-off between the route discovery ratio, the number of route request messages and the number of links of the tree. The experimental results have shown that MgCast could achieve a good balance between these metrics.**

I. Introduction

Recent innovation of wireless technologies have brought us new wireless network architectures called mobile ad-hoc networks (MANETs) where mobile terminals (called nodes hereafter) directly and adaptively build networks without fixed stations. Building a route on MANETs should often be carried out on demand since the network topology frequently changes due to mobility of nodes or instability of wireless connections. However, on-demand routing, which usually requires to flood route request (RREQ) messages to all the nodes, is very expensive [1] especially for power-aware devices.

Here, MANET applications are sometimes location-aware applications such as field games and orienteering. In such an application, each person has his/her own mobile terminal, obtains its geographical location via GPS, and communicates with the other persons in specific geographic areas. Considering this fact, several research efforts have been dedicated to position-based routing protocols on MANETs where the number of RREQ messages is reduced by assistance of position information of mobile nodes (see Refs. [2]–[4] for surveys). The common idea is that if a destination is designated by its geographical location, some directionality can be specified to RREQ messages, *e.g.* RREQ messages towards opposite directions may not be able to find a route to the destination and hence, can be discarded.

In this paper, we propose a location-aware multicast protocol on MANETs called *MgCast (multiCAST for Multiple Geographical regions)*. MgCast finds and maintains a routing tree to given multiple geographical regions in a cost-conscious and distributed way. The main idea lies in the following two principles. (i) Assuming that each node knows the neighboring nodes' location information by HELLO messages, a node in MgCast forwards an RREQ message only to a specific number of the neighboring nodes which are closer to the destination area and which are not within each other's radio range. This simple idea can avoid redundant RREQ messages while keeping reasonable route discovery ratios. (ii) For given multiple destination regions, a shared tree is constructed to reduce the redundant duplication of data packets. Since the construction is done in a decentralized manner, no explicit tree computation at a certain node is required.

The experimental results have shown that MgCast could achieve a good balance between a route discovery ratio, the number of RREQ messages, and the number of links of tree.

II. Related Work and Motivation

Assuming that each node can obtain its position (*e.g.* through GPS), many literatures have presented position-based routing methods (see Refs. [2]–[4] for surveys).

Several researches have presented position-based unicast, multicast and anycast routing protocols assuming that the location information of a specified destination node can be obtained by some location service. DREAM [5] proposes a unicast protocol where an efficient update method of position information of destination nodes using their distances and mobility rates is presented. GPSR [6] also presents a unicast routing protocol but can detour obstacles by using perimeter (or face) routing. The protocol in [7] uses a Voronoi diagram in combination with the mobility prediction of the destination node to determine neighboring nodes to which packets are forwarded. The protocol in [8] considers position-based multipath routing for robustness, quality of service and other reasons. The above protocols localize the decision procedures to decide neighbors to which packets are forwarded in order to make them scale to the growth of the number of mobile nodes. Meanwhile, for small group multicast on MANET, DSM [9] gives an efficient and centralized technique for constructing multicast trees assuming that each node knows the geographic locations of all the other nodes.

Geocasting is also a form of position-based routing where destination nodes are those in a specified region (*i.e.* a sender specifies a region as a destination). Ref. [10] presents Location Based Multicast (LBM), a geocast protocol enhanced from Location Aided Routing (LAR) [11]. In LBM, each node forwards a message only to nodes within a certain area (the smallest rectangle which contains both the sender and the destination region) called a forwarding zone. The forwarding zone can be extended or shrunk by a parameter δ, depending on obstacles, mobile node density and so on. GeoGRID [12] partitions a geographic space into squares (grids). In each grid, a node called a gateway is selected and is responsible for forwarding packets to neighboring grids. There are two different types of message forwarding policies, flood-based and ticket-based. The former one is similar with LBM, while the latter one can control the number of messages by tickets, which are issued by a source and mean the permission to send messages. OFSGP/OFMGP [13] also partitions a geographic space into cells (hexagons). Direction-based depth-first search is performed to detour obstacles. Geomulticast [14] also presents a cell-based method and adopts a distance-based greedy forwarding policy. GAMER [15] is a multi-path routing based on MGRP [16]. This adopts a similar forwarding policy as LBM but differs in maintaining and adapting the created mesh topology. From the viewpoint of forwarding policies, Refs. [5], [7], [8], [10], [12], [14], [15] adopt greedy forwarding which may be a simpler form, while Ref. [6], [13], [17] adopt a hybrid form of greedy forwarding and face routing (messages traverse along obstacles [18]) which will achieve higher route discovery ratios under existence of obstacles but usually require larger number of messages.

Our target applications are future mobile applications on MANETs such as event navigation systems. Let us suppose that people in an outdoor event such as a school festival have small devices equipped with GPS and those people are navigated by the devices. We may want to send an electronic festival guide to people at several school entrances (usually more than one) or special event notification to several crowded regions. For such a purpose, we may build a MANET composed by those walking people's devices and may require a protocol to efficiently deliver messages to the destination regions over the MANET.

Our observations in designing a geocast protocol for such an application are followings. First, we need a simple and localized (*i.e.* distributed) geocast routing. Guaranteed delivery (including simple flooding) can discover a route to a destination if it exists, but requires a number of RREQ messages. We pursue trade-off between route discovery ratio and the number of RREQ messages. Therefore we adopt a kind of a greedy forwarding policy which is adopted in many geocast routing. However, different from those approaches, each node in our approach forwards an RREQ message only to a constant number (usually two or three) of neighboring nodes selected based on distance to a destination region and each other's radio range. This simple idea can achieve high route discovery ratio as well as saving the number of RREQ messages. Note that clustering like Refs. [12]–[14] is very effective for large-scale MANET but there is a protocol overhead due to extra operations such as leader election. Considering our target application that deals with mid-size regions such as campus-wide regions, a simpler form is better. Second, we want to reduce the cost of packet delivery to multiple regions after route discovery. For this purpose, we construct a shared tree which covers the multiple destinations. To our best knowledge, only OFMGP [13] explicitly considers multiple regions along with detouring obstacles, however, they do not consider the tree cost. Finally, we maintain routing states at nodes. Many existing protocols present pure on-demand (stateless) approaches, assuming high mobility of nodes. MgCast constructs a tree and maintains it as long as possible, since our aim is to reduce the total cost in deliver packets on MANETs composed by slowly moving devices such as walking people and bicycle (we have assumed bicycle in setting the maximum velocity of nodes in our experiments). Note that under high mobility, the route discovery process of MgCast can work as a pure on-demand routing protocol if necessary.

III. Route Discovery

A. Preliminaries

We assume that each node has its unique ID and can obtain its geographic coordinate by GPS or some other ways. Recently, compared with the radio range (usually a few hundred meters) of mobile nodes, errors in GPS are much smaller (at most a few tens of meters). Therefore, we assume that obtained coordinates of mobile nodes are accurate for simplicity. However, in practice, errors from the accurate values should be considered because frequent update of coordinates through GPS may not be possible due to battery limitation or some nodes may not be equipped with GPS. We will give discussion about this issue in Section V-B. Each node knows the IDs of its neighboring nodes (*i.e.* nodes in its radio range) and their coordinates. These information is periodically exchanged by probe messages called *HELLO messages*. Note that the above information can also be exchanged via any other control messages defined later (*e.g.* RREQ, RREP and so on) to keep it up to date. For simplicity of discussion, we assume that each node has the same radio range.

A node sends a message to its neighbors in a broadcast manner. That is, all the neighbors receive a message by a single broadcast (*i.e.* promiscuous broadcast). After receiving a message, each neighbor in MgCast takes some action or just discards the message depending on the content of the message.

A sender node (say s) specifies a destination region group $D = \{d_1, ...d_n\}$. Each destination region d_i is a rectangle, and is expressed by the coordinate of the center point c_i, width and height.

B. Overview of Route Discovery

When the sender s wants to send packets to destination regions, MgCast activates a route discovery process and tries to find a tree which covers those destination regions. Note that as stated in Section II, MgCast does not provide a guaranteed delivery service, *i.e.* a route may not be found even if there logically exists. We will show in the experimental section that MgCast could archive reasonable trade-off between route discovery ratios and the numbers of RREQ messages. Once a tree is found, MgCast maintains the tree.

We propose a distributed heuristic route discovery algorithm. In the proposed algorithm, an RREQ message contains a destination region group and a corresponding set of neighboring nodes which should be responsible for forwarding RREQ messages for these destination regions. If such a node, say node v, receives an RREQ message, then it (i) divides the given destination region group in the RREQ message into

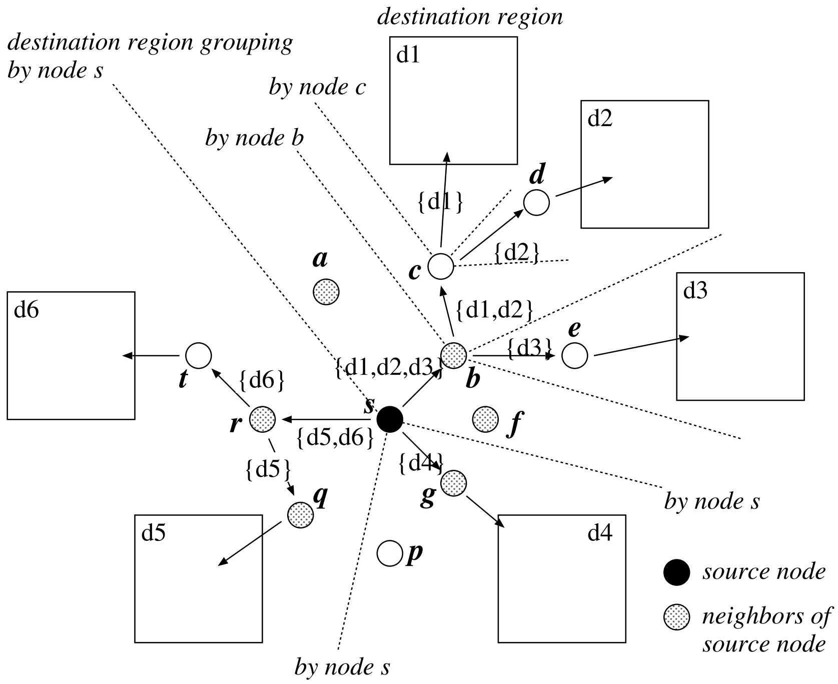


Fig. 1. Route Discovery Process in MgCast (forwarding degree=1)

a set of smaller sub-groups, (ii) assigns some neighboring nodes of v to each sub-group, and (iii) broadcasts an RREQ message which contains the set of sub-groups and corresponding neighbors. By continuing this process, RREQ messages approach the destination regions. Consequently, the RREQ messages form tree-like routes from the data source to the destination regions. Then *RREP (Route Reply)* messages are propagated from the destination regions to the data source using the routes, and finally a tree is formed. The tree state is kept at intermediate nodes. We do not construct a tree within destination regions. Instead, we use flooding limited inside the regions.

Here we show how the route discovery process works in more details. In an RREQ message from a node u, a set of "*ND-pairs*" ("ND" implies "Neighbors and Destination-regions") is designated. Each ND-pair is a pair of a *neighbor group* and a *destination region group* and has the following form "$\langle N, D\rangle$" where N is a set of some neighbors of node u and D is a destination region group. An ND-pair means that the neighboring nodes in N are responsible for forwarding RREQ messages to the destination regions in D. A node v listening to an RREQ message from u acts as follows.

1) If node v is not designated in any of the ND-pairs in the RREQ message, node v just discards it.
2) Otherwise, let $ND_i = \langle N_i, D_i\rangle$ denotes the ND-pair where node v is specified in N_i. If node v is not inside any destination region in D_i, it divides the destination region group D_i into a set of sub-groups and selects some of its neighbors for each sub-group. They form a set of ND-pairs. Node v broadcasts an RREQ message with this set of ND-pairs.

We exemplify this operation in Fig. 1. In the figure, node s is the source node and there are six destination regions d_1,...,d_6. At first, node s examines possible division patterns of the given destination region group $\{d_1, ..., d_6\}$. For example, $\{\{d_1, d_2, d_3\}, \{d_4\}, \{d_5, d_6\}\}$ is one possible pattern. For each division pattern, node s examines its distance-based cost (this cost is defined in the next section) and selects the pattern which minimizes the cost. Let us assume that the previous division pattern $\{\{d_1, d_2, d_3\}, \{d_4\}, \{d_5, d_6\}\}$ is such a pattern. Node s assigns node b to sub-group $\{d_1, d_2, d_3\}$, node g to sub-group $\{d_4\}$ and node r to sub-group $\{d_5, d_6\}$. Note that for each sub-group, only one neighbor is selected in this example, however, the number of neighbors here is a protocol parameter. The number of neighbors to be assigned to a sub-group is called a *forwarding degree*.

The sub-groups and associated neighbors $\langle\{b\}, \{d_1, d_2, d_3\}\rangle$, $\langle\{g\}, \{d_4\}\rangle$ and $\langle\{r\}, \{d_5, d_6\}\rangle$ form a set of ND-pairs. Then node s broadcasts an RREQ message including this set of ND-pairs. Nodes b, g and r receive the RREQ message and act in the same way (the other neighbors discard the message). For example, node b broadcasts an RREQ message with a set of ND-pairs $\langle\{c\}, \{d_1, d_2\}\rangle$ and $\langle\{e\}, \{d_3\}\rangle$.

When a node in a destination region receives an RREQ message from a neighbor outside the destination region, it replies an RREP message to the neighbor. The RREP message is propagated through the route to the source node in updating the routing tables of the intermediate nodes. Finally, by receiving RREP messages from all (or some) of the destination

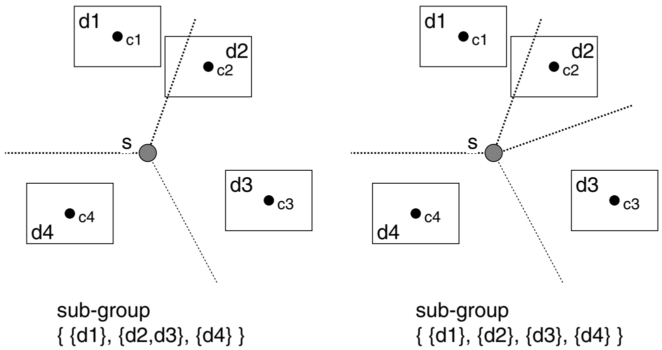


Fig. 2. Two Example Sets of Sub-groups for Given Set $D = \{d_1, d_2, d_3, d_4\}$

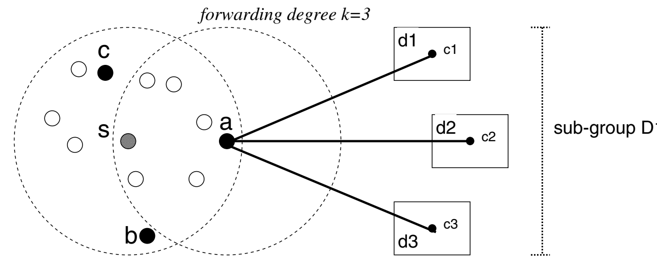


Fig. 3. Neighbors Selection for a Sub-group $D_1 = \{d_1, d_2, d_3\}$

regions, the source node is ready for sending data packets. If RREP messages from some destination regions are missing, the source nodes may choose whether it activates another route discovery process for the missing destination regions or not, depending on the types of the applications.

C. Determining ND-pairs in RREQ Messages

Source node s divides the given destination region group $D = \{d_1, ..., d_n\}$ into a set of sub-groups $\{D_1, .., D_m\}$ where $\bigcup_i D_i = D$ and $\forall i, j \;\; D_i \cap D_j = \emptyset$. In determining a set of sub-groups $\{D_1, .., D_m\}$ ($m \leq n$) from given D, the following optimization is performed. Each possible set $\{D_1, .., D_m\}$ of sub-groups is found by separating the given destination region group D by m lines ($m \leq n$) from s based on their center points (see Fig. 2). Here, we regard that the cost of a routing tree is approximately proportional to the total Euclid distance from the source to the center points of the destination regions. Therefore, for each sub-group D_i, we find a neighbor v which minimizes the following cost

$$Cost(D_i) = \min_{v} \sum_{d_j \in D_i} d(v, c_j)$$

where $d(a, b)$ is the Euclid distance between two points a and b, and c_j is the center point of destination region d_j. Fig. 3 shows an example. Node s selects node a where the total distance from a to the center point c_1, from a to c_2 and from a to c_3 is minimum for a sub-group $D_1 = \{d_1, d_2, d_3\}$. Then we define the minimum cost of a set $\{D_1, ..., D_m\}$ of sub-groups as follows.

$$Cost(\{D_1, ..., D_m\}) = \sum_{0 \leq i \leq m} Cost(D_i)$$

Finally, once node s determines the set of sub-groups $\{D_1, ..., D_m\}$ and the corresponding m neighbors, node s further selects $k-1$ neighbors for each sub-group D_i where k is the forwarding degree. The selection policy is as follows. For each D_i and its already selected neighbor, we iterate selecting a node (i) which is not selected yet and not in the radio range of already selected neighbors and (ii) which makes $Cost(D_i)$ minimum, until $k-1$ nodes are selected or until there is no node which satisfies the above. For example in Fig. 3 (forwarding degree $k = 3$), node a has been already selected as the neighbor for $D_1 = \{d_1, d_2, d_3\}$. Then, node b is selected since it is not in the radio range of a and makes $Cost(D_i)$ minimum. Finally, node c is selected and as a result, three nodes a, b and c are selected for D_1. Compared with a simple greedy forwarding method which determines those neighbors based only on the distance to the destination, this increases possibility to reach the destination regions even if there are sparse node areas or obstacles between s and the destination regions because this enables us to find detouring routes. Also, we can avoid interference between the selected neighbors, and can prevent them from selecting the same nodes as the nodes to which RREQ messages are forwarded.

Source node s specifies m ND-pairs in an RREQ message and broadcasts it to neighbors. If a neighbor u receives the RREQ message, it checks whether node u is contained in an ND-pair $ND_i = \langle N_i, D_i \rangle$ or not. If so, the same process as the above is performed for the given destination set D_i, otherwise node u just discards the message.

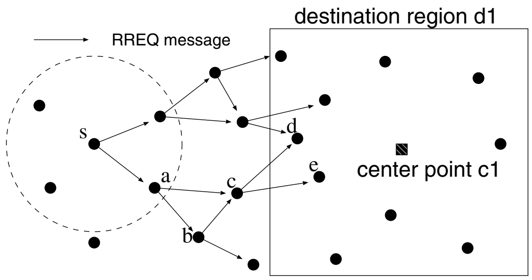
D. Route Reply (RREP) Message

If a node v in a destination region d_i receives an RREQ message from a node u outside d_i, it sends a route reply (RREP) message to node u. This message is propagated to node s using the reverse path. Note that the reverse path is kept in routing tables at intermediate nodes explained in the next section. Source node s waits for a certain period after it sends the first RREQ message. If it receives RREP messages from the all of the destination regions, it knows that a tree has successfully been formed involving all the destination regions. If RREP messages from some destination regions are missing, as stated earlier, the source node may choose whether it activates another route discovery process for the missing regions with an incremented sequence number or not. This choice may depend on the context of the applications.

E. Data Delivery inside Destination Regions

If a node v in a destination region d_i receives a data packet, it broadcasts the data packet to all the neighbors inside the region d_i. This is called *Limited Flooding*.

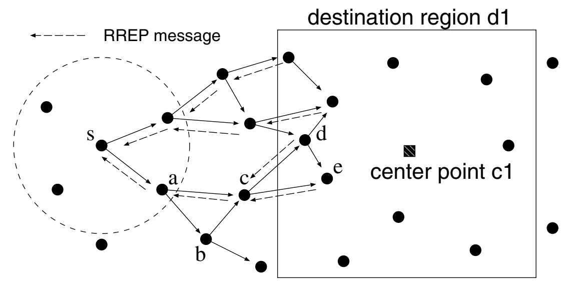
We do not manage routing tables inside regions because there may exist a lot of nodes (remember that our target applications may send packets to crowded regions), and the limited flooding is a simple form to deliver packets to them.



routing table of node c

src	dst	from	to	metric	seq	flag	lifetime
s	d_1	a	d	2	1	*False*	LT-1
s	d_1	a	e	2	1	*False*	LT-1
s	d_1	b	d	3	1	*False*	LT
s	d_1	b	e	3	1	*False*	LT

(a) RREQ



routing table of node c

src	dst	from	to	metric	seq	flag	lifetime
s	d_1	a	d	2	1	*True*	LT-3
s	d_1	a	e	2	1	*True*	LT-3
s	d_1	b	d	3	1	*False*	LT-2
s	d_1	b	e	3	1	*False*	LT-2

(b) RREP

Fig. 4. Message Propagation and Table Update

Note that of course there may exist nodes inside destination regions which cannot receive data packets in MgCast.

IV. Route Maintenance

A. Routing Table Update

Each node in MgCast may keep a routing table. Each entry of the table has the following form.

$$\langle src, dst, from, to, metric, seq, flag, lifetime\rangle$$

src is the ID of a source node, *dst* is the ID of a destination region, *from* is the ID of a neighbor from which data packets are forwarded, *to* is the ID of a neighbor to which data packets are forwarded, *metric* is the metric of the route (hop count in this paper) from the source node to the node, *seq* is the sequence number specified in the corresponding RREQ message, *flag* is a boolean variable to represent whether this entry is active or not and *lifetime* is the residual time to the expiration of the entry. In forwarding RREQ messages and RREP messages, routing tables at intermediate nodes are updated. Note that at the time of creating an entry, the node sets lifetime to the entry which decreases as time passes and is updated whenever the entry is looked up. The entry with zero lifetime will be removed from the table.

Let us assume that node v receives an RREQ message from u. Also assume that node v is contained in an ND-pair $ND = \langle N, D\rangle$ in the message and sends an RREQ message with a set of ND-pairs $\{ND_1, ..., ND_m\}$. Then node v creates a new entry $\langle s, d, u, w, hc, seq, False, LT\rangle$ for each $ND_i = \langle N_i, D_i\rangle$ where $w \in N_i$, $d \in D_i$, hc and seq are the hop count and the sequence number specified in the received RREQ message respectively, and LT is a constant (a protocol parameter) which represents the initial lifetime of this entry.

Note that node v may receive more than one RREQ message with the same destination regions and the same sequence number. In such a case, node u sends an RREQ message only when it receives the first RREQ message. For the following RREQ messages, it just updates the routing table as described above and never sends RREQ messages.

Fig. 4(a) shows a snapshot of the routing table of node c after node c receives two RREQ messages with a same sequence number 1 from nodes a and b. We assume that the RREQ message from node a arrived one unit of time prior to the one from node b. Upon the arrival of the RREQ messages, entries were added, but node c itself sent a single RREQ message in response to the earlier one.

When node v receives from node w an RREP message which notifies the discovery of a destination region d and forwards it to node u, node v sets the *flag* field of the entry $\langle s, d, u, w, hc, seq, False, LT\rangle$ to *True*. This means that node v forwards to the neighboring node w data packets which originally come from s and go to the final destination region d. Fig. 4(b) shows an example. On receiving an RREP message from e, node c selects node a, forwards the RREP message to a and sets *True* value to the *flag* field of the corresponding entry. Node c also receives an RREP message from node d, and sets *True* to the corresponding entry (node c does not send an RREP message at this time).

Note that the nodes in a destination region d do not have entries about d since limited flooding is used inside destination regions.

B. Mobility Support

Each node probes its neighbors and their positions by HELLO messages. If a neighbor moves and leaves from the radio range, the related entries are removed from the table. In MgCast, non-active entries (marked by *False* flag) can be used as backup entries. If an entry at node v, $\langle s, d_i, u, w, hc, seq, True, lt\rangle$, is no longer effective due to the move of w (or v), v seeks another entry with destination d_i with *False* flag, and may send an RREQ message using the entry on behalf of node s to find an alternative route. Even though this seems effective, there may be the case that an RREP may not be returned to node v. In such a case, node v

sends a *route disconnection (RDIS)* message towards node s. Then node s can send an RREQ message for the destination regions d_i.

V. Discussion

A. Complexity

In this section, brief analysis of message size and computation complexity of ND-pairs is given.

In MgCast, only RREQ messages have a variable length field which contains a set of ND-groups. Its maximum size is $O(k|D|)$ where k is the forwarding degree (usually two or three) and $|D|$ is the number of destination regions. Also we look at the processing complexity at each node to calculate a set of ND-groups. There are at most $\sum_{0 \leq i \leq |D|} {}_{|D|}C_i$ sets of ND-pairs for given destination region group D.

In practical applications which we have assumed in Section II where destination regions can be several crowded regions, $|D|$ can be regarded as a constant (for example, less than 10). The above costs are not really expensive in the context of those applications.

B. Accuracy and Availability of Position Information

It might be considered expensive to assume that every node has always knows its accurate position. Therefore it is important to see whether MgCast works fine under (i) the existence of some nodes without GPS equipments and with (ii) inaccurate position information.

For (i), if the number of such nodes is not so large, we can assume that these nodes can estimate their positions by receiving messages (HELLO, RREQ, RREP etc.) from surrounding (neighboring) nodes which include their position information. In such a case, we can regard the problem of (i) as (ii) (*i.e.*, non-equipped nodes can obtain inaccurate information). Under this assumption, in our experiments, we have evaluated the performance of MgCast with longer beacon intervals (HELLO packets intervals) and higher mobility of nodes both of which lead to inaccuracy of position information. See section VI-C for the results.

C. Affects of Lossy Wireless Links

We should also consider the case that some messages are lost by lossy wireless links. If some control messages are lost, as stated before, timeout of RREP messages occurs at a sender and the sender will retry the route discovery process or stop discovering routes. Note that to avoid frequent retransmissions in the network layer, we may use IEEE802.11 MAC sublayer notification mechanism which notify packet retransmission timeout. This can be used to increase reliability in the link layer.

For loss of data packets, we may use ACK messages from nodes in destination regions. This can be implemented in upper layers and we do not guarantee reliable communication in the network layer to keep the operation of the protocol as much as simple.

VI. Experimental Results

We have implemented MgCast on Glomosim [19] simulator.

A. Simulation Setup

In our experiments, we have used an Euclidean space of 2000m $\times$ 2000m, and rectangles of 100m up to 300m as destination regions. We randomly put at least 300 and up to 500 mobile nodes onto the space. Each node has a common radio range, and in some simulation cases where the radio range was changed, it ranges from 120m up to 200m by every 20m. We have used IEEE802.11.

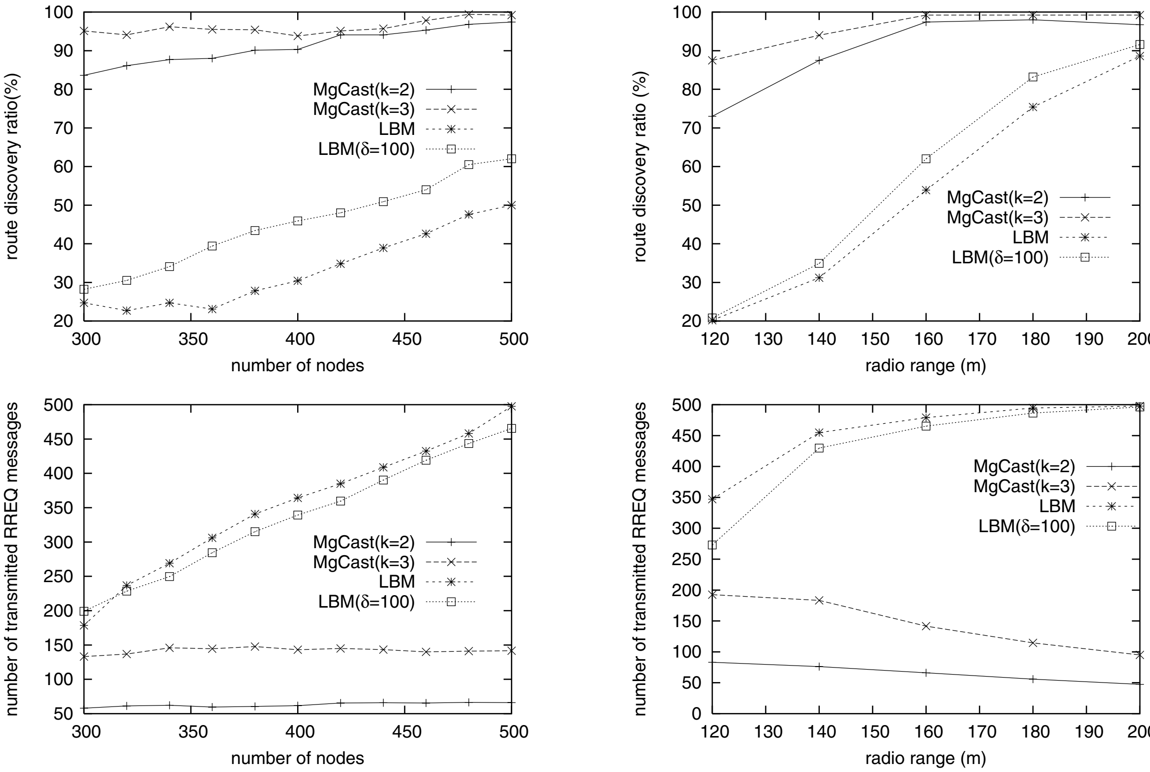
For the mobility model, we have adopted the random waypoint model provided in Glomosim with some pause time. In some simulation cases, the pause time ranges from 0 to 60 seconds by every 10 seconds. The moving speed ranges from 0.5m/s up to 7m/s where we assume slow vehicles such as bicycle or walkers. These values were determined by assuming battery-aware small devices. In each simulation case, the simulation time was 500 seconds and a randomly selected source node continued to send two data packets of 512bytes payload in every second. Bandwidth of each wireless link was 2Mbps. In all the simulation cases, for packet loss and collision, we did not use any recovery mechanism.

As comparison, we have used Location-Based Multicast(LBM) proposed in Ref. [10]. This method is based on Location-Aided Routing (LAR) presented in Ref. [11]. We have modified a Glomosim implementation of LAR [19] to implement LBM. Two values 0 and 100 were used for the parameter δ of LBM in our experiments. $\delta = 0$ means that a forwarding zone is the smallest rectangle which contains both a sender and a destination region, while $\delta > 0$ means that the width and height of a forwarding zone are both extended by 2δ (*i,e,* a large forwarding zone). Note that in case for multiple destination regions, in order to see the cost of the trees of MgCast, we have used a modified version of MgCast where a route is discovered for every destination region (that is, each destination region is treated independently of the others). This version of MgCast is referred to as “MgCast/ind”. We have also used SPTs (shortest path trees) to see optimality of the number of tree links.

B. Performance in Route Discovery

First, we have measured the performance of MgCast in route discovery processes by measuring the following metrics. Therefore, in the experiments in this section, we did not consider the mobility of nodes (the next section we will take mobility into account). Hereafter, a *single route discovery process* is referred to as a single trial of a source node to discover a tree.

- *Route discovery ratio*. This is the successful ratio of a single route discovery process. We say that a single route discovery process is successful if and only if a tree involving all the destination regions is found.
- *The number of transmitted RREQ messages*. This is the total number of RREQ messages transmitted during a single route discovery process.
- *Tree cost*. This is the number of tree links when a tree is found by a single route discovery process.



(a) vs. # of Nodes (b) vs. Radio Range

Fig. 5. Route Discovery Ratio (top) and Number of Transmitted RREQ Messages (bottom) under $|D| = 1$ (default settings: radio range $d = 160m$, # of nodes$=500$ and $IDD = 900m$)

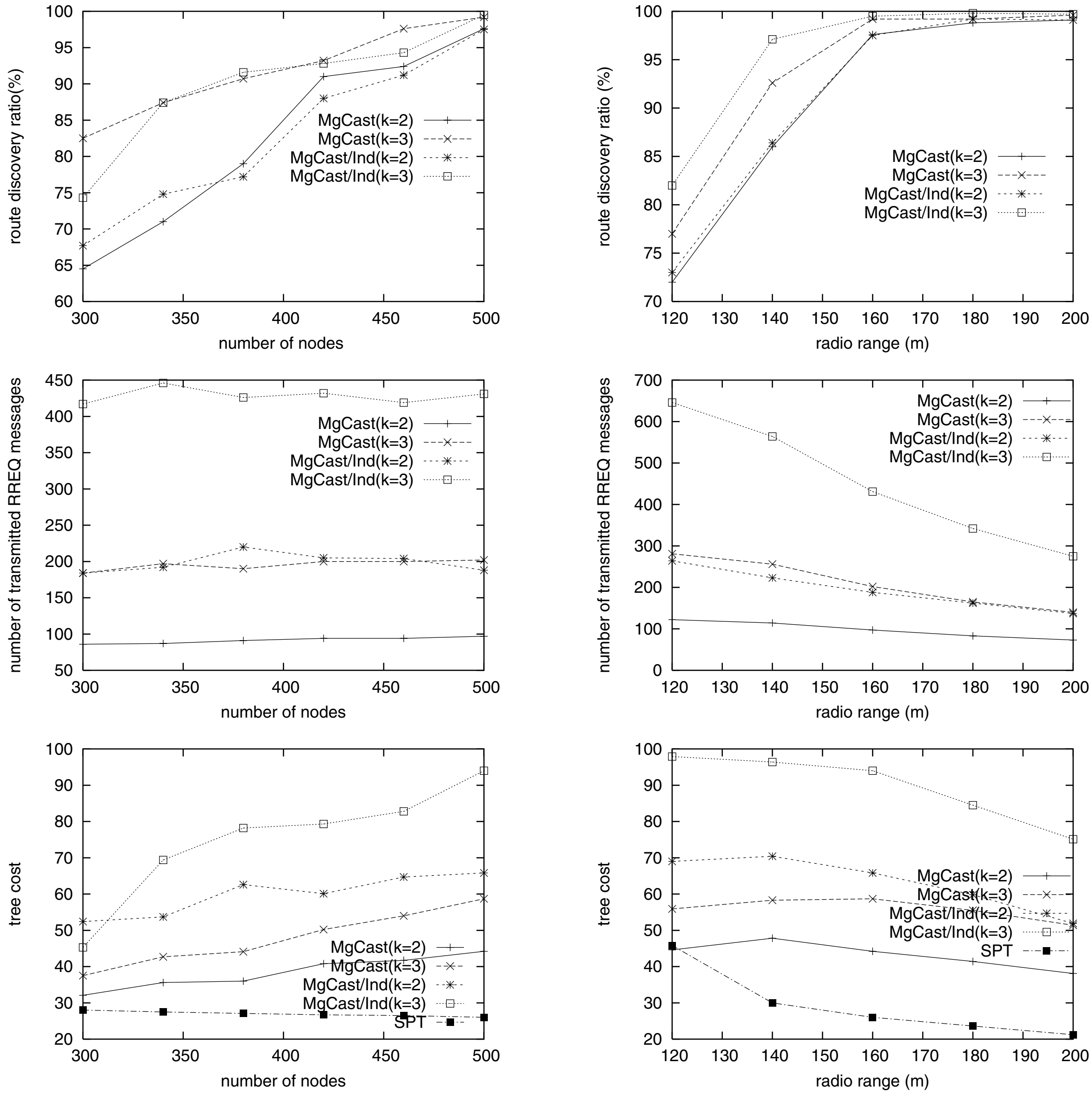
We have measured these values by executing route discovery processes in six different settings. In each setting, we have varied the number N of nodes, radio range d of nodes, the number $|D|$ of destination regions, or Inter-Destination Distance (IDD) which is the maximum Euclid distance between two centers of destination regions (thus a smaller IDD means that regions are closer to each other). In each setting, we have carried out route discovery processes 500 times and presented their average values. Note that we have only used node allocations where at least one tree has existed logically and we have set the forwarding degree k to 2 or 3.

Figs. 5, 6 and 7 present the simulation results under six different settings. In each setting, we have aligned the results of route discovery ratio, the number of transmitted RREQ messages and tree cost from top to bottom.

In Figs. 5(a) and 5(b), we have presented the results in two different settings of $|D| = 1$ (a single destination region) where the number N of nodes and radio range d had been varied respectively. We do not present the tree cost because $|D| = 1$ in these settings. Compared with LBM, MgCast with forwarding degrees two and three achieved higher route discovery ratios and smaller numbers of transmitted RREQ messages. Note that, as seen in Fig. 5(b), the discovery ratios in LBM were largely affected by radio ranges. This means that, LBM suffers from the number of nodes in forwarding areas (a smaller radio range will reduce the number of neighboring nodes in a forwarding zone).

We have presented in Figs. 6(a) and 6(b), the results in two different settings of $|D| = 3$ (multiple destination regions) where the number N of nodes and radio range d had been varied respectively. The discovery ratios of MgCast and MgCast/ind were not so different for the same forwarding degree k. However, the numbers of RREQ messages and the tree costs were very different. With the same forwarding degree, MgCast outperformed MgCast/ind. This have shown that MgCast could construct effective shared trees. Note that we have also measured the tree costs of SPTs (Shortest Path Trees) which were constructed by an off-line algorithm as benchmarks. Of course there was certain difference because MgCast set its forwarding degree more than one to increase route discovery ratios. However, the difference is still reasonable compared with MgCast/ind.

Finally, in order to see the affect of the number $|D|$ of regions and their locations, we have presented the results in other two different settings in Figs. 7(a) and 7(b) where $|D|$ and IDD (Inter-Domain Distance) had been varied, respectively. As seen in Fig. 7(a), the larger $|D|$ decreased the discovery ratios of both MgCast and MgCast/ind, however, it increased the numbers of RREQ messages and tree costs of MgCast/ind faster than those of MgCast. From Fig. 7(b), we cannot see drastic change of performance by varying IDD.



(a) vs. # of Nodes (b) vs. Radio Range

Fig. 6. Route Discovery Ratio (top), Number of Transmitted RREQ Messages (mid) and Tree Cost (bottom) under $|D| = 3$ (default settings: radio range $d = 160m$, # of nodes= 500 and $IDD = 900m$)

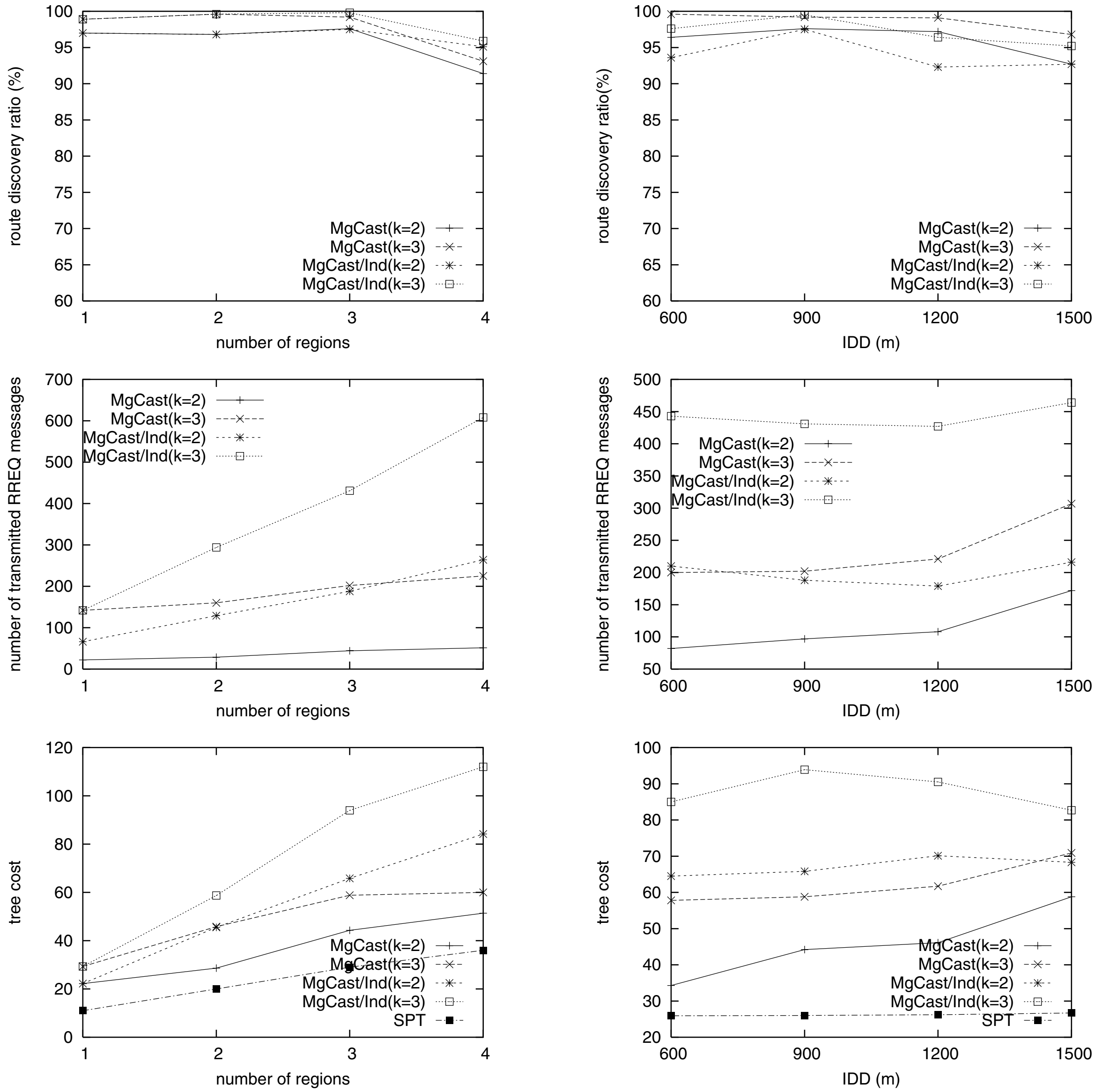
C. Performance under Mobility of Nodes

Secondly, we have measured the following metrics assuming mobility of nodes.

- *Data packet arrival ratio.* Let us define a single data packet's arrival ratio as the ratio of the total number of destination regions which received the packet over the number of destination regions. Data packets arrival ratio is the average data packet arrival ratio of all the data packets throughout a simulation case.
- *Message overhead.* This is the total byte amount of all the control messages (RREQ, RREP, HELLO and RDIS) throughout a simulation case.

We have measured these values by executing 20 simulation cases varying the maximum velocity V_{max} of nodes and presented their average values. This measurement is different from the one in the previous section in the points that a route discovery process may also be triggered by nodes' movement and disconnection of routes in this experiment and that routing tables are maintained. The results are shown in Fig. 8. Note that as we discussed in Section V-B, we have varied the period B (sec.) to exchange HELLO messages (i.e. beacon interval) as well as maximum velocity to see the affects of accuracy of position information.

From the results in Fig. 8(a), as V_{max} and B grew, the data packet arrival ratios became smaller. This indicates that



(a) vs. # of Regions

(b) vs. IDD

Fig. 7. Route Discovery Ratio (top), Number of Transmitted RREQ Messages (mid) and Tree Cost (bottom) under the Different Number of Destination Regions and Their Locations (default settings: radio range $d = 160m$, # of nodes= 500, # of regions= 3 and $IDD = 900m$)

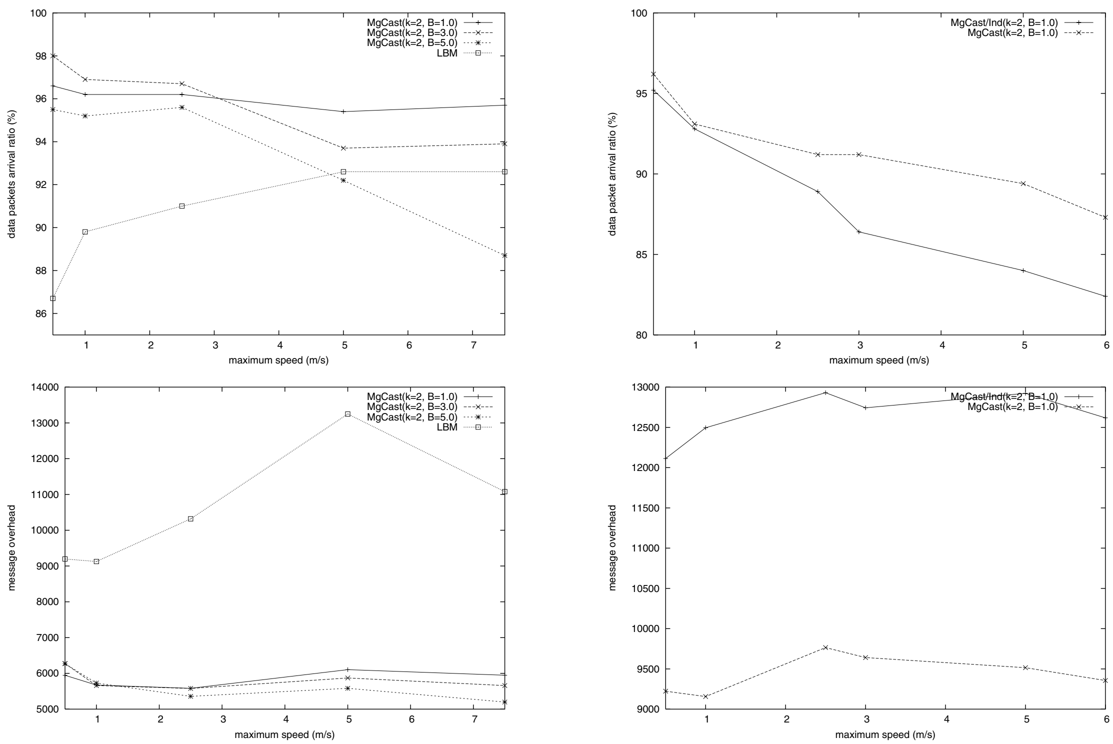
the accuracy actually affected the MgCast protocol, however, it is still reasonable compared with LBM. Here, LBM is a stateless protocol, so we have applied route discovery process of LBM whenever a route was broken. Therefore the high message overhead and high data packet arrival ratios of LBM were very essential in this context. Nevertheless, MgCast could achieve reasonable discovery ratio compared with LBM.

What we should also observe here is that message overhead did not increase as V_{max} increased and with larger B. This feature is very important, because we want to avoid higher protocol overhead which consumes a lot of batteries at nodes.

Then we concentrate on seeing the affects of maximum velocity in the case of $|D| = 3$ (Fig. 8(b)). MgCast/ind was more affected by increase of V_{max} than MgCast. This is because MgCast/ind has more (independent) routes than MgCast and thus each route is likely to be broken by node mobility. Due to the same reason, MgCast/ind transmitted much more amount of bytes (messages) than MgCast.

VII. Conclusion

In this paper, we have proposed a location-aware MANET multicast called MgCast for multiple geographic regions. Given a source node and a set of geographical destination regions, MgCast constructs and maintains a routing tree from the source to nodes which reside in the regions, in a decentralized manner. In MgCast, assuming that each node only knows the location information of the neighboring nodes, each node forwards RREQ (route request) messages only to the



(a) vs. V_{max} ($|D| = 1$) (b) vs. V_{max} ($|D| = 3$)

Fig. 8. Data Packets Arrival Ratio (top) and Message Overhead (bottom) under Different Maximum Velocities (default settings: radio range= $160m$, # of nodes= 150 and $IDD = 900m$)

few number of neighboring nodes which are close to the destination regions and which are not within each other's radio range, in order to decrease redundant RREQ messages and to increase route discovery successful ratios. Moreover, for multiple destination regions, a shared tree which involves these regions is constructed so that the number of links of the tree can be small.

Conducting simulation experiments to compare the performance with the other geocast protocols is part of our future work.

REFERENCES

[1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. of ACM/IEEE Mobicom*, 1999, pp. 151–162.

[2] T. Imielinski and J. Navas, "GPS-based geographic addressing, routing, and resource discovery," *Communications of the ACM*, pp. 86–92, 1999.

[3] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, 2001.

[4] I. Stojmenovic, "Position based routing in ad hoc wireless networks," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–134, 2002.

[5] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. of ACM/IEEE Mobicom*, 1998, pp. 76–84.

[6] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of ACM/IEEE Mobicom*, 2000, pp. 243–254.

[7] I. Stojmenovic, A. Ruhil, and D. Lobiyal, "Voronoi diagram and convex hull based geocasting and routing in wireless networks," in *Proc. of IEEE Symp. on Computers and Communications (ISCC)*, 2003, pp. 51–56.

[8] X. Lin and I. Stojmenovic, "Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 1, pp. 22–32, 2003.

[9] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, "Location aware, dependable multicast for mobile ad hoc networks," *Computer Networks*, vol. 36, no. 5–6, pp. 659–670, 2002.

[10] Y.-B. Ko and N. H. Vaidya, "Geocasting in mobile ad hoc networks; location-based multicast algorithms," in *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, 1999, pp. 101–110.

[11] ——, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proc. of ACM/IEEE Mobicom*, 1998, pp. 66–75.

[12] 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.

[13] C.-Y. Chang, C.-T. Chang, and S.-C. Tu, "Obstacle-free geocasting protocols for single/multi-destination short message services in ad hoc networks," *Wireless Networks*, vol. 9, no. 2, pp. 143–155, 2003.

[14] B. An and S. Papavassiliou, "Geomulticast: architectures and protocols for mobile ad hoc wireless networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 182–195, 2003.

[15] T. Camp and Y. Liu, "An adaptive mesh-based protocol for geocast routing," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 196–213, 2003.

[16] J. Boleng, T. Camp, and V. Tolety, "Mesh-based geocast routing protocols in an ad hoc network," in *IEEE Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2001, pp. 184–193.

[17] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proc. of ACM Mobihoc*, 2001.

[18] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proc. of ACM Mobihoc*, 2003.

[19] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *Glomosim: A Scalable Simulation Environment for Wireless and Wired Network Systems*, http://pcl.cs.ucla.edu/projects/domains/glomosim.html.