Internal node and shortcut based routing with guaranteed delivery in wireless networks

Susanta Datta[1], Ivan Stojmenovic[1,2] and Jie Wu[3]
[1]SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada ivan@site.uottawa.ca
[2]DISCA, IIMAS, UNAM, A.P. 20-726, Del. A. Obregon, Mexico D.F. 01000, Mexico
[3]Comp. Sci. & Eng., Florida Atlantic Univ., Boca Raton, FL 33431-0991, USA jie@cse.fau.edu

Abstract

Several distributed routing algorithms for wireless networks were described recently, based on location information of nodes available via Global Positioning System (GPS). In greedy routing algorithm sender or node S currently holding the message m forwards m to one of its neighbors that is the closest to destination. The algorithm fails if S does not have any neighbor that is closer to destination than S. FACE algorithm guarantees the delivery of m if the network, modeled by unit graph, is connected. GFG algorithm combines greedy and FACE algorithms. Greedy algorithm is applied as long as possible, until delivery or a failure. In case of failure, the algorithm switches to FACE algorithm until a node closer to destination than last failure node is found, at which point greedy algorithm is applied again. In this paper we further improve the performance of GFG algorithm, by reducing its average hop count. First we improve the FACE algorithm by adding a sooner-back procedure for earlier escape from FACE mode. Then we perform a shortcut procedure at each forwarding node S. Node S uses the local information available to calculate as many hops as possible and forwards the packet to the last known hop directly instead of forwarding it to the next hop. The second improvement is based on the concept of dominating sets. The network of internal nodes defines a connected dominating set, and each node must be either internal or directly connected to an internal node. We apply several existing definitions of internal nodes, namely the concepts of intermediate, inter-gateway and gateway nodes. We propose to run GFG routing, enhanced by shortcut procedure, on the dominating set, except possibly the first and last hops. We obtained localized routing algorithm that guarantees delivery and has very low excess in terms of hop count compared to the shortest path algorithm. The experimental data show that the length of additional path (in excess of shortest path length) can be reduced to about half of that of existing GFG algorithm.

1. Introduction

Wireless networks consist of static or mobile hosts (or nodes) that can communicate with each other over the wireless links without any static network interaction. Each mobile host has the capability to communicate directly with other mobile hosts in its vicinity. They can also forward packets destined for other nodes. Examples of such networks are wireless local area networks (WLANs), packet radio networks, and sensor networks. They are used in situations like disaster rescues, wireless conferences in the hall, battlefields, or monitoring objects in a possibly remote or dangerous environment.

The routing problem is the problem of finding a route for sending a message from a source to a given destination. Routing becomes very difficult in wireless networks. In highly mobile situation, the flooding scheme is the most reliable for sending data packets. However, since the link channel resource is very scarce and battery power is limited, more efficient schemes must be devised. Numerous routing protocols have been proposed in recent years [BMJHJ, RS].

Ad hoc networks are best modeled by the graphs constructed in the following way. Each node A has its transmission range $t(A)$. Two nodes A and B in the network are neighbors (and thus joined by an edge) if the Euclidean distance between their coordinates in the network is less than the minimum between their transmission radii (i.e. $d(A,B) < min \{t(A), t(B)\}$). If all transmission ranges are equal, the corresponding graph is known as the *unit graph*. These models provide acknowledgments for received messages.

The shortest path algorithm does not adapt well to the networks in which some of the nodes may be temporarily inactive due to failures or power savings. In addition to activity status and location updates for all nodes in the network, the updates on the status of every possible link in the network is needed to guarantee the availability of shortest path, which is unacceptable quadratic communication overhead. Wireless networks require *localized* algorithms, in which nodes make routing decisions based solely on the information about its neighboring nodes, and the position of destination. One such method, the greedy routing algorithm, where each node forwards message to its neighbor that is closest to destination, is based on the location information supplied by GPS. Its major drawback is high failure rate for low degree graphs. Its major advantage is that its hop count is very close to the shortest path algorithm, whenever the method is successful. A localized routing algorithm, called *FACE,* that guarantees the message delivery in connected unit graphs, and its improved version, called *GFG* (Greedy-Face-Greedy), that applies the greedy algorithm as much as possible, are described in [BMSU]. We apply the concept of internal nodes to improve the delivery rates of *GFG* algorithm. Another improvement proposed here is a shortcut procedure that allows each node in *FACE* algorithm to predict few next hops and forward the message directly to the last of these hops.

The desirable properties of any routing protocol include simplicity, loop-free operation, convergence after topological changes, storage overhead, computational and transmission overhead. The efficiency of a routing algorithm is measured by the delivery rate and the average hop count. The average hop count is the number of transmissions needed by a method between source and destination. It is compared with the average hop count of a shortest path algorithm. The delivery rate is the ratio of numbers of messages received by destination and sent by senders. For a shortest path algorithm, and for algorithms that guarantee delivery in a connected graph, the ratio is 1. Therefore the only measure used in this paper for comparing routing algorithms that guarantee the delivery is hop count.

The algorithms analyzed and proposed in this paper guarantee delivery if the position of destination is accurate. Sensor networks are important kind of wireless networks whose nodes may be considered fixed. These networks may have thousands of sensors that change their status between active and passive, and therefore shortest path algorithm is not feasible in such case. In some cases nodes in sensor networks may move (if attached to people or vehicles or if moved by wind), but the destination can be a fixed central facility known to all sensors, which collects information from sensors. When the position of destination is fixed, as in this scenario, the algorithms discussed here guarantee delivery as long as the location updates are regularly updated between moving neighboring nodes. If destination is moving, as in the case of ad hoc networks, routing algorithm may begin by sending message to the latest known location of destination, and modify the information by intermediate nodes as the message approaches destination. Since this approach may not guarantee delivery and may increase hop count significantly, we believe that a better approach will apply a destination search method using short probe messages (containing search information but not actual message), then routing from destination to sender with short message, and finally routing full (actual) message from source to destination whose location is detected. The destination search step may apply a flooding with reduced number of retransmissions (such as one based on internal nodes [SSZ]) or another method that is linked to location update scheme (e.g. [S]). In this scenario, delivery is guaranteed unless node movement is significant with respect to message speed (in which case flooding may be the only feasible solution [HOTV]). Moreover, in this application, the network can be considered static in the analysis of algorithm performance, without introducing significant errors in measurements.

This paper is organized as follows. Sections 2 and 3 present a literature reviews on GPS based routing in ad hoc networks, and the concept of dominating sets. Section 4 proposes a *shortcut* procedure as the main enhancement to *FACE* algorithm from [BMSU]. Section 5 proposes routing algorithm that restricts routing decisions to internal nodes only. Performance evaluation is given in section 6, followed by conclusion section.

2. Known routing algorithms

Finn [F] proposed a greedy routing algorithm, in which source or any intermediate node will choose the successor node that makes the best progress toward the destination. The distance to destination is used to measure the progress. When no node is closer to the destination than current node *C*, the algorithm fails. Power aware routing algorithms in unit graphs are studied in [SL2]. Routing algorithms that are based on depth first search algorithm, merged with *GEDIR*, are proposed in [SRV]. These algorithms require nodes to memorize past traffic, which is not the case with algorithms discussed here. The performance of *DFS* based routing algorithms can also be improved by applying internal node concept [SRV].

Bose, Morin, Stojmenovic and Urrutia [BMSU] described a GPS based localized routing algorithm which guarantees the delivery for wireless networks modeled by unit graphs, assuming only that the graph is connected. The only additional constraint is, of course, that the position of destination (as recorded by the sender node) is reasonably accurate. Let *G* be the graph that corresponds to the wireless network. Thus it contains *n* vertices, and two vertices are connected by an edge if and only if the distance between them is $<R$.

The algorithms works as follows. In the first phase, construct planar connected subgraph *P* of *G*. *P* is the intersection of unit graph and Gabriel graph *GG* on the set *S* of *n* given nodes, defined as follows. Let *disk(u,v)* be the disk with diameter *(u,v)*. Then, the Gabriel graph *GG(S)* is a graph in which edge *(u,v)* is present if and only if *disk(u,v)* contains no other points of *S*. The planar subgraph is constructed in localized manner. Let *A* be one of nodes, and *B* one of its neighbors. Edge *AB* belongs to *P* if and only if the circle with diameter *AB* does not contain any other node from *G*. It suffices to check, for each neighboring node *C* of *A*, whether $|CM|<|AB|/2$, where *M* is the center of the circle, that is the middle point between *A* and *B*. Alternative criterion is the following: an edge *AB* is included in the subgraph if and only if ∠*ACB* is acute, for each joint neighbor *C* of *A* and *B*. In the second phase, a path between source node *S* and destination node *D* in *P* is constructed. It is based on traversing faces that intersect imaginary line from source to destination.

The paths generated by *FACE* algorithms are considerably long. However, nodes do not need to memorize past traffic in order to forward the message correctly. Also, at each step there is exactly one copy of the message in the network. The *FACE* algorithm may be used in practice in combination with an algorithm that performs well whenever it does not fail, such as greedy routing scheme. [BMSU] proposed an algorithm that begins routing with greedy scheme until message is delivered, or fails at a node *C*. The failure node is defined as node *C* which has no closer neighbor to *D* than itself. Such choice node enables to prove that the combined algorithm remains loop-free, and guarantees delivery. At *C*, *FACE* algorithm is invoked which guarantees delivery. The algorithm is further improved, by proposing *GFG (Greedy-FACE-Greedy)* routing scheme. In this scheme, routing follows greedy method until a node *C* is reached that has no neighbor closer to destination than itself. The distance $d=|CD|$ is attached to the message, and *C* invokes *FACE* algorithm which runs until message is delivered, or a node *B* such that $|BD|<d$ is found. Routing then continues again with greedy until message delivery of the next node *C* that has no closer neighbor to destination. Thus the number of switches from greedy to *FACE* and back may be arbitrary, but each switching node is closer to destination than the previous one. The performance evaluation in [BMSU] shows that, and in case of low degree graphs with few hundred nodes, *GFG* generates paths which are on average up to 3.5 times longer than paths generated by shortest path (*SP*) algorithm. In case of high degree networks, the performance of *GFG* is very close to the performance of *SP* algorithm. Improving the algorithm performance for low degree graphs is the research problem studied in this paper.

3. Dominating sets and internal nodes

A set is dominating if all the nodes in *G* are either in the set or neighbors of nodes in the set. Nodes that belong to a dominating set will be called, in this paper, internal nodes for *G* (of course, a different definition for dominating set leads to different set of internal nodes). Routing based on a connected dominating set is frequently used approach [WL], where the searching space for a route is reduced to corresponding internal nodes. The routing process, in this approach, is divided into three steps. If source node is not an internal node, it forwards the packets to one of its adjacent nodes. This internal node then acts as a new source to route the packets in the reduced graph consisting of internal nodes only. Eventually, the packets reach the destination internal node which is either the destination node itself or neighbor of the destination node. In the later case, the destination internal node forwards the packets directly to the destination node. Such routing is suggested for the shortest path, and DSR (dynamic source routing) [WL].

Wu and Li [WL] proposed a simple and efficient distributed algorithm for calculating connected dominating set in ad hoc wireless networks, where connections of nodes are determined by their geographical distances. They introduced the concept of an *intermediate* node. A node *A* is an *intermediate* node if there exist two nodes *B* and *C* (different from *A*) such that *A* is on a shortest path between them. The length of path is equal to the number of hops (that is, hop count, or number or transmissions) needed to reach the destination from a given source. Next, they proved that a node *A* is an intermediate node if and only if *A* is on a shortest path between two nodes that are *neighbors* of *A*. This criterion is equivalent to finding whether there exist two neighbors of *A* that are not direct neighbors themselves. If location information (that is, *x*- and *y*-coordinates) of nodes are not known to nodes then the list of 2-hop neighbors (that is, neighbors of each neighbor) suffices to make such decision at each node (and these lists can be directly exchanges between neighboring nodes). On the other hand, if nodes know the location of all its neighbors, and *G* is unit graph, then each node can determine whether or not it is an intermediate nodes in $O(k^2)$ computation time (where *k* is the number of its neighbors), and without any message exchanged with its neighbors for that purpose.

Let *V'* be the set of intermediate nodes, and *G'* be the corresponding induced subraph of *G* (thus *V'* is the set of vertices of *G'*). *G'* is connected subgraph [WL]. Dijskstra's shortest path algorithm applied on intermediate nodes produces the shortest possible path between any two nodes [WL]. The maintenance consist of verifying the intermediate status for each node in the presence of topological changes, and propagating the new intermediate status to the rest of network if it changed for a given node.

Wu and Li [WL] introduced also two rules that considerably reduce the number of internal nodes in the network, and proved that these rules preserve network connectivity after eliminating additional nodes from the dominating set. Let *N(u)* be the (open) set of all neighbors of node *u*, and let $N[u]=N(u) \cup \{u\}$ be the corresponding closed neighbor set, that is the set of all neighbors and *u* itself. Suppose that each node has a unique *id* number (it may be obtained by generating a random number in [0,1], or use *x*-coordinate as *id*). Let us define *inter-gateway* nodes as nodes intermediate nodes that are not eliminated by Rule 1. Next, let the *gateway* nodes be those intermediate nodes that are not eliminated by both rules. Rule 1 [WL] is as follows.

Consider two intermediate nodes *v* and *u*. If *N[v]* $\subseteq$ *N[u]* in *G* and *id(v)* < *id(u)*, then node *v* is not an inter-gateway node. In other words, if any neighbor of *v* is also a neighbor of *u*, and *v* is connected to *u* and has lower *id*, then any path via *v* can be replaced by a path via *u*, thus node *v* is not needed as internal node. We may also say that node *v* is 'covered' by node *u*. Observe that the hop count between any two nodes does not increase by applying Rule 1, since a segment *pvq* of a path between source and destination is replaced by a segment *puq*, which does not change the hop count.

The number of internal nodes (that is, gateways) can be further reduced by applying Rule 2 [WL], as follows. Assume that, after applying Rule 1, *u* and *w* are two gateway neighbors of a gateway node *v*. If *N(v)* $\subseteq$ *N(u)* $\cup$ *N(w)* in *G* and *id(v)* = min {*id(v), id(u), id(w)*}, then node *v* is declared a non-gateway node. In other words, if each neighbor of *v* is a neighbor of *u* or *w*, where *u* and *w* are two connected neighbors of *v*, then *v* can be eliminated from the list of gateway nodes (when, in addition, *v* has lowest *id* among the three). The hop count between a source and destination node may increase by one in this process, since a segment *pvq* of the path between them is replaced by one hop longer segment *puwq*. Observe that, for both rules, each node may decide to withdraw from the list of internal nodes by knowing its 1-hop neighbors only, since location information suffices to check whether all them are neighbors of neighboring internal nodes. Otherwise, list of 2-hop neighbors is necessary for that decision.

Stojmenovic, Seddigh and Zunic [SSZ] proposed to replace node *id*s with a record (*degree, x, y*), where *degree* is the number of neighbors of a node, and *x* and *y* are its two coordinates in the plane (if available, otherwise they may use a random number instead). In both rules from [WL], nodes compare first their degrees, and node with higher degree has greater chances of remaining an internal node. In case of ties, *x*-coordinate is used to resolve. The information about the degree of neighboring nodes may be gathered together with information about their location.

4. Shortcut FACE algorithm

One of main problems with *FACE* algorithm, causing its increased hop count, is that the planar subgraph construction phase favors short edges over long ones. A route between source and destination is therefore likely to contain a number of short edges. This means that a node *A* in the path might have, in addition to forwarding node, few more nodes in the same path as its neighbors. Can *A* predict few hops in the path? Consider an example where node *A* is neighbor of nodes *B, C* and *E* on a path *ABCEF…* toward destination. Each node makes forwarding decision based on the location of its neighbors, destination, and the last intersection *X* of an edge with direction *SD* (source-destination). Thus node *A* needs to know the location of each neighbor of *B, C* and *E,* in order to predict that the message will traverse indicated path. Instead of forwarding the message to *B, A* can forward the message directly to *F,* making a shortcut. Note that *A* can predict next hops even if face is changed in the process.

In order to apply the shortcut procedure, neighboring nodes need to exchange the list and location of their neighbors, in addition to their own location, whenever a change in local topology occurs. In other words, 2-hop local information is needed.

5. Routing via internal nodes

We propose to apply *GFG* algorithm on the set of internal nodes. The message is sent from source node to one of its neighboring internal nodes that is closest to destination (node that such internal node does not need to be closer to destination than the source itself). The routing algorithm is then restricted to the set of internal nodes. The route follows *GFG* routing algorithm, non-internal nodes being ignored. The algorithm terminates whenever an internal node that is connected to destination *D* is reached. At this point, the message is delivered to *D*. One possible improvement for *GFG* algorithm can be defined as follows. We shall define *GFG-sooner-back* algorithm that runs *FACE* algorithm until a node *B* is found that has at least one neighbor that has distance <*d* to destination. The algorithm switches back to greedy algorithm at node *B*. This *GFG-sooner-back* version is likely to return back to greedy algorithm from *FACE* mode few hops sooner than *GFG* will do. If *B* has few neighbors at distance <*d* from destination, the closest one is chosen, according to greedy scheme. In the performance evaluation given in the next section, *GFG* refers, in fact, to *GFG-sooner-back* version.

Further improvement of *GFG* algorithm is obtained by applying shortcut procedure, as described in the previous section, where *GFG-S* algorithm is described. The performance evaluation given in the next section separates *GFG* and *GFG-S* algorithms.

6. Performance evaluation

The experiments were carried using random connected unit graphs, which were generated as follows. Each of *n* nodes is chosen by selecting its *x* and *y* coordinates at random in the interval [*0,100*). In order to control the average node degree *d*, we sort all *n(n-1)/2* (potential) edges in the network by their length, in increasing order. The radius *R* that corresponds to chosen value of *d* is equal to the length of *nd/2*-th edge in the sorted order. Dijkstra’s shortest path algorithm can be used to test whether a graph is connected.

Comparison of intermediate and inter-gateway node based routings show that intermediate node definition performs notably worse but not by large difference. The

FACE algorithm alone has up to 8% longer paths, while *GFG* and *GFG-S* algorithm have up to 3% longer paths. The differences are larger for smaller degree graphs. A very close performance of *GFG* and *GFG-S* on the inter-gateway and gateway node dominating sets was observed, with differences being always below 1%. Although gateway node definition apparently reduced the size of dominating set, it increases the path lengths at the same time, and, interestingly, the same result at the end is obtained. Because of simplicity and preserving path lengths, we conclude that inter-gateway node definition is the one that has the best performance.

Table 1 gives average hop counts for several routing methods that guaranty message delivery. Averages are taken over 20 connected graphs (different for different *d*) and each source-destination pairs. The number of nodes ranges from *n*=40 to *n*=100, and degrees are from 3 to 10. Data for only inter-gateway node definition of dominating sets are shown. Four such methods are indicated with suffix 'IG'. Methods that use shortcut procedure have suffix 'S'.

Degree	3	4	5	6	7	8	9	10
FACE	33.0	26.4	15.9	12.1	10.7	9.5	7.8	7.6
FACE-S	26.1	18.5	10.3	7.1	5.8	4.9	4.0	3.6
FACE-IG	23.3	18.8	13.0	10.2	9.3	8.3	7.2	6.9
FACE-IG-S	19.5	14.2	8.8	6.4	5.5	4.6	3.8	3.5
GFG	20.5	14.4	7.6	5.2	4.2	3.6	3.1	2.9
GFG-S	17.8	11.7	6.5	4.7	3.9	3.5	3.1	2.9
GFG-IG	15.3	11.5	6.8	4.9	4.1	3.6	3.1	2.9
GFG-IG-S	14.2	10.2	6.1	4.5	3.8	3.4	3.1	2.8
SP	8.9	6.4	5.0	4.1	3.6	3.3	3.0	2.8

Table 1. Hop counts for considered routing methods for *n*=60 and *d*=3-10

Let us define the *dilation* as the ratio of hop count for given method and the hop count of *SP* (shortest path) algorithm. It is desirable to have the ratio as close to 1 as possible. The largest dilation reported in [BMSU] for *GFG* algorithm (without shortcut procedure) was 3.5 (for *n*=100 and *d*=4), and our goal was to reduce that number as much as possible. For *n*=60, the largest dilation reported in [BMSU] was 2.3, for *d*=4 (no values were reported for *d*=3).

It can be observed that *FACE-S* algorithm, that is *FACE* algorithm enhanced with shortcut procedure, has very low dilation for high degree graphs (under 1.7 for *d* ≥ 7). *FACE-IG-S* (the previous algorithm applied on inter-gateway nodes) comes even closer, under 1.6 for *d* ≥ 5. *GFG-S* reduces the dilation from 2.3 to under 1.9 for *d≥4*, *GFG-IG* to under 1.8, and finally *GFG-IG-S* to under 1.6. Thus we obtained localized routing algorithm with guaranteed delivery whose excess in hop count with respect to the shortest path algorithm is under 60% for *d≥3.* Thus improvement of adding internal node concept and shortcut procedure over former *GFG* algorithm in terms of dilation is about 30% for lower degree graphs. In fact, the improvement is much higher if measured in terms of excess path rather than full path. Obviously we cannot improve the length of the shortest path, thus we can improve only the length of additional path. In this respect, the 1.3 'added dilation' of *GFG* [BMSU] is reduced to 0.6 added dilation, thus over 53% reduction.

Table 2 shows the dilations for number of nodes ranging from 40 to 100, and degrees between 3 and 10. Clearly, the proposed localized algorithm compares very well with the global shortest path algorithm, especially for dense networks.

n\d	3	4	5	6	7	8	9	10
40	1.45	1.26	1.15	1.08	1.03	1.02	1.01	1.01
50	1.52	1.43	1.25	1.08	1.04	1.04	1.04	1.02
60	1.60	1.58	1.24	1.10	1.06	1.04	1.01	1.01
70	1.85	1.53	1.26	1.11	1.12	1.02	1.02	1.02
80	2.00	1.74	1.26	1.18	1.10	1.07	1.04	1.02
90	2.17	1.69	1.41	1.16	1.10	1.08	1.04	1.02
100	2.27	2.08	1.41	1.17	1.11	1.04	1.03	1.03

Table 2. Dilations of *GFG-IG-S* algorithm

Conclusions

The performance evaluation of our algorithm demonstrated that very significant improvements were obtained, and that proposed improvements to routing algorithm lead to algorithm that perform very close to the performance of the shortest path algorithm for all graph average degrees.

We are currently developing power aware routing algorithms with guaranteed delivery [SD], by merging algorithms from [SL2] with improved *FACE* algorithm. One of the problems associated with dominating set definition used in this paper is their fixed status in the absence of node mobility, causing more energy consumption at internal nodes. We are investigating [WGS] an alternate definition of dominating set, that will take node remaining battery level into consideration. This will enable change of internal node status in favor of nodes with more battery power.

The *GFG* algorithm, enhanced by shortcut procedure in *FACE* mode, and applied on internal nodes, may be applied to solve geocasting problem. In geocasting problem, a message is to be sent from a node to all nodes within a region (e.g. circle). *GFG* may be used to route toward the center of the geocasting region until a point inside the region is reached. The algorithm then switches to flooding with retransmission savings [SSZ].

It is also an interesting problem to find another kind of planar subgraphs which is denser than Gabriel graph (GG). *FACE* routing algorithm may perform better if more faces and edges are present in the subgraph. The new kind of subgraphs needs also to be defined locally, without any message exchange. Delaunay triangulation (DT), for instance, is denser subgraph but is not locally defined. GG is subgraph of DT. It is well known [BM] that GEDIR guarantees delivery in DTs, thus DT is a good candidate for study. More precisely, to include the largest possible portion of DT in the planar subgraph, which can be locally found. For instance, each triangle in DT contains no other point from the set, which can be used to include all such triangles whose all sides have size comparable to unit graph radius and therefore locally detectable. Note that the circumscribed circle of that triangle should also be of locally manageable size. Further improvements in the algorithm performance could be obtained by following this approach.

References

[BM] P. Bose and P. Morin, Online routing in triangulations, Proc. of the 10th Annual Int. Symp. on Algorithms and Computation ISAAC, 113-122, 1999.

[BMSU] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, 3rd int. Workshop on Discrete Algorithms and methods for mobile computing and communications, Seattle, August 20, 1999, 48-55.

[BMJHJ] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, Proc. MOBICOM, 1998, 85-97.

[F] G.G. Finn, Routing and addressing problems in large metropolitan-scale internetworks, ISI Research Report ISU/RR-87-180, March 1987.

[RS] S. Ramanathan and M. Steenstrup, A survey of routing techniques for mobile communication networks, Mobile Networks and Applications, 1, 2, 1996, 89-104.

[SD] I. Stojmenovic, S. Datta, Power aware routing with guaranteed delivery in wireless networks, manuscript in preparation.

[SSZ] I. Stojmenovic, M. Seddigh and J. Zunic, Internal node based broadcasting in wireless networks, Proc. IEEE Hawaii Int. Conf. on System Sciences, January 2001, to appear.

[SRV] I. Stojmenovic, M. Russell, and B. Vukojevic, Depth first search and location based localized routing and QoS routing in wireless networks, IEEE International Conference on Parallel Processing, August 21-24, 2000, Toronto, 173-180.

[SL1] I. Stojmenovic and X. Lin, GEDIR: Loop-free location based routing in wireless networks, IASTED Int. Conf. on Parallel and Distributed Computing and Systems, Nov. 3-6, 1999, Boston, MA, USA, 1025-1028.

[SL2] I. Stojmenovic and Xu Lin, Power aware localized routing in wireless networks, IEEE Int. Parallel and Distributed Processing Symp., Cancun, Mexico, May 1-5, 2000, 371-376.

[WL] J. Wu and H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, Proc. DIAL M, Seattle, Aug. 1999, 7-14.

[WGS] J. Wu, M. Gao and I. Stojmenovic, On calculating power-aware connected dominating sets for efficient routing and broadcasting in wireless networks, in preparation.

This research is partially supported by NSF, NSERC, and REDII grants.