GPS based distributed routing algorithms for wireless networks

Xu Lin and Ivan Stojmenovic
Computer Science, SITE, University of Ottawa
Ottawa, Ontario K1N 6N5, Canada
ivan@site.uottawa.ca

Abstract

Recently, several fully distributed (localized) GPS based routing protocols for a Mobile Ad hoc NETwork (MANET) were reported in literature. They are variations of directional *(DIR)* routing methods, in which node *A* (the source or intermediate node) transmits a message *m* to several neighbors whose direction is closest to the direction of *D*. We also found an older, *MFR* (most forward progress within radius) method.

We propose a new location based GEographic DIstance Routing (GEDIR) algorithm. When node *A* wants to send *m* to node *D*, it forwards *m* to it's neighbor *C* which is closest to *D* among all neighbors of *A*. The same procedure is repeated until *D*, if possible, is eventually reached. 2-hop *GEDIR*, *DIR*, and *MFR* methods are also suggested, in which node *A* selects the best candidate node *C* among its 1-hop and 2-hop neighbors according to the corresponding criterion (distance, direction, and progress, respectively) and forwards *m* to its best 1-hop neighbor among joint neighbors of *A* and *C*. These basic and 2-hop methods do not require nodes to memorize past message traffic. We propose flooding *GEDIR*, *DIR* and *MFR* methods, intended to guarantee the message delivery at the expense of MANET' s partial flooding. Further, we introduce three variants of multiple path *c-GEDIR*, *c-DIR* and *c-MFR* methods, in which *m* is initially sent to *c* best neighbors according to corresponding criterion, and afterwards, on intermediate nodes, it is forwarded to only the best neighbor. They provide multiple paths with minimal flooding effects.

We show that the directional routing methods are not loop-free, while the *GEDIR* and *MFR* methods are inherently loop free. The simulation experiments with static random unit graphs show that *GEDIR* and *MFR* have similar success rates, with hop counts that are near the performance of the shortest path algorithm, while *DIR* method has comparable success rate but worse hop count. Further, the performance of *DIR* method worsened when 2-hop neighbors were taken into account, while 2-hop *GEDIR* and *MFR* have improved their performance. Flooding *GEDIR* and *MFR* methods are the first distributed methods (other than full flooding) that guarantee the delivery, and are shown to have low flooding rates. Disjoint multiple path methods are shown to provide high success rates and small hop counts for small values of *c*.

Index terms: Routing, wireless networks, distributed algorithms, mobile computing, shortest path

1. Introduction

In this paper we consider the routing task, in which a message is to be sent from a source node to a destination node (in a sensor or ad hoc wireless network). The nodes in the network may be static (e.g. thrown from an aircraft to a remote terrain or a toxic environment), static most of the time (e.g. books, projectors, furniture, motors) or moving (vehicles, people, small robotic devices).

A broad variety of location dependent services will become feasible in the near future due to the use of the Global Position System (GPS), which provides location information (latitude, longitude and possibly height) and global timing to mobile users. GPS cards will be deployed in each car and possibly in every user terminal [K, NI]. For instance, NAVSTAR Global Positioning System has a potential accuracy of about 50-100 meters and Differential GPS offers accuracy of a few meters [N]. In the USA, Federal Communications Commission has adopted rules requiring wireless service providers to supply two-dimensional location information of mobile users who request the E-911 emergency service [EMMB]. Navas and Imielinski [NI] described GPS' s application in geographic messaging to users who are located within a particular polygon or circle defined by latitude and longitude. Their method is based on a hierarchy of geographically defined routers, and the intersection of the appropriate levels of routers with the given polygon or circle.

Wireless networks of sensors are likely to be widely deployed in the near future because they greatly extend our ability to monitor and control the physical environment from remote locations and improve our accuracy of information obtained via collaboration among sensor nodes and online information processing at those nodes. Networking these sensors (empowering them with the ability to coordinate amongst themselves on a larger sensing task) will revolutionize information gathering and processing in many situations. Sensor networks have been recently studied in [EGHK, HCB, HKB, KKP]. A similar wireless network that received significant attention in recent years is ad hoc network [IETF, MC]. Mobile ad hoc networks (MANETs) consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure. Routes between two hosts in MANET may consist of hops through other hosts in the network. The task of finding and maintaining routes in MANET is nontrivial since host mobility causes frequent unpredictable topological changes. A number of MANET protocols for achieving efficient routing have been recently proposed. They differ in the approach used for searching a new route and/or modifying a known route, when hosts move. The surveys of these protocols, that do not use geographic location in the routing decisions, are given in [BMJHJ, RS]. A number of novel routing protocols are also available on the internet [IETF]. In this article we will discuss only GPS based approaches.

Macker and Corson [MC] listed qualitative and quantitative independent metrics for judging the performance of routing protocols. Desirable qualitative properties include: distributed operation, loop-freedom (to avoid a worst case scenario of a small fraction of packets spinning around in the network), demand-based operation, and 'sleep' period operation (when some nodes become temporarily inactive). Some quantitative metrics that are appropriate for assessing the performance of any routing protocol include [MC]: end-to-end data delay, average number of data bits (or control bits) transmitted per data bits delivered. In this paper we use three quantitative metrics that are similar to these described in [BMJHJ] (each of them is an average value):

- *hop count* (the number of edges, i.e. transmissions on the path from source to destination),
- *delivery rate* (the ratio of numbers of messages received by destination and sent by senders),
- *flooding rate* (the ratio of the number of message transmissions and the shortest possible hop count between two nodes). Each transmission in multiple routes is counted, and message can be sent to all neighbors with one transmission.

Although 'algorithm' and 'protocol' have the same meaning in literature, we shall have a subtle difference in our discussions. The routing methods are described by algorithms which underline only major ideas of the corresponding detailed protocol. The actual protocol may always include additional techniques, most of them already being applied in other protocols, and details of communication between nodes. This paper will focus on routing algorithms, not protocols.

Ad hoc networks are best modeled by *minpower* 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. $d(A,B) < min\{t(A), t(B)\}$) [BCSW]. If all transmission ranges are equal (to the radius R of the graph), the corresponding graph is known as the *unit graph*. These models provide acknowledgments for received messages. The minpower and unit graphs are valid models when there are no obstacles in the signal path (e.g. a building). Ad hoc networks with obstacles can be modeled by subgraphs of minpower or unit graphs. This paper deals primarily with unit graphs.

It is unlikely to expect that one routing protocol for MANET is the best approach for all networking contexts. Thus it is not surprising to find a number of hybrid methods in literature, combining several major ideas into a single protocol. In the next section, we shall review existing routing protocols [BCSW, KV, KSU, HL, NK, TK] which use geographic location in their route decisions. Variations of a directional routing algorithm are recently proposed [BCSW, KV, KSU] and are shown to perform significantly better than the methods that do not use geographic location in routing tables. Although [BCSW] claims that directional routing algorithms provide loop-free paths, we shall give a counterexample showing that undetected loops can be created. Our literature review revealed some other GPS based methods from early 1980's [HL, NK, TK]. One of them, *MFR* method [TK] is a competitive method and we prove here that it is loop-free.

We shall describe the *GEDIR* algorithm, and prove that it is inherently loop-free. The proof does not use unit graph properties, and is therefore valid for any kind of network, including networks in three dimensional space. Several modifications to *GEDIR*, *MFR* and *DIR* methods, which should provide a better trade-off between delivery and flooding rates are also described here. 2-hop neighbors may be used to enhance delivery rate and shorten hop count. Flooding may be used at nodes where basic method drops the packet. Finally, the sender may initialize *c* paths toward destination, to provide multiple paths that involve significantly lower flooding rates.

In all algorithms, it is assumed that each node is aware of the geographic location of all other nodes in MANET (in accordance with [BCSW, KV]). The question of location updates is not addressed in this paper, and all techniques presented in [BCSW, KV] and other sources may apply. We assume that the mobile nodes are moving in a two-dimensional plane. Since nodes may move, the actual locations may differ from the one recorded in the routing tables. If a pure unit graph model is assumed, based on the location information, each node may calculate shortest paths (using breadth first search, for example) to all other nodes (in time proportional to the number of edges), and may select the first neighbor on the route to all destinations. This algorithm provides the shortest paths if the location information is reasonably accurate and all nodes are active. However, such shortest path (*SP*) algorithm (proposed also in [BCS, SWR], and used in this paper as the benchmark) does not adapt to 'sleep' period operations, since the shortest paths can be 'broken' by inactive nodes. Even if this information is updated with node' s position, theunit graph model assumed here is merely a reasonably good approximation of the actual network. Nodes that are at distance less than *R* may have an obstacle between them blocking the communication, while two nodes at distance that exceeds *R* by a small amount may still be able to communicate between them (or a node may even choose whether to use that possible link). Thus the use of *SP* algorithm requires the regular update of existing edges in addition to nodes location, which is a quadratic (in number of nodes) overhead requiring considerable bandwidth and battery power. Estrin, Govindan, Heidemann, and Kumar [EGHK] also argued that localized algorithms (in which simple local node behavior achieves a desired global objective) may be necessary for sensor network coordination. They described clustering and object location localized algorithms.

It is assumed here that each node is aware of its inactive neighbors (and possibly inactive 2-hop neighbors). The algorithms discussed in this paper use only the location of destination and location (and activity) information of direct neighbors (and possibly 2-hop neighbors) to make a decision on forwarding the message (in distributed manner). In 1-hop and 2-hop *GEDIR*, *DIR* and *MFR* methods, there is exactly one copy of each message in MANET at all times, that is, each intermediate node will forward the message to exactly one of its neighbors. The memory requirements for storing the information about the past traffic at each node differ in algorithms that will be discussed. 1-hop and 2-hop *GEDIR*, *DIR* and *MFR* algorithms do not memorize any message previously forwarded to any of neighbors. Messages in flooding based algorithms are memorized only at special nodes (if any) while multiple path methods memorize past traffic at each node.

Several experiments are designed to measure the performance of proposed routing algorithms on static random unit graphs. Although the algorithms are designed with mobile networks in mind, the experiments are performed with static networks only for several reasons. First, the selected routing protocol should perform well on static networks, which are important special case to be considered (in other words, it makes no sense to evaluate performance of a method on moving network if that method does not perform well on static one). Nodes in some circumstances barely, if at all, move (for instance, sensors thrown from an aircraft). In some cases nodes may move, but destination could be fixed and known to nodes (e.g. police stations or collectors of sensor data). Location update needed for efficient routing in such cases is minimal, and is restricted to neighboring nodes only. Next, even the problem of routing in static networks only is far from being solved completely in this paper, and more papers on the subject are forthcoming [BMSU, SL, S1, S2]. Further, the impact of selected location update scheme or movement patterns of nodes is eliminated, thus leaving only pure routing algorithm to be investigated (in other words, the presence of an ideal location update scheme is assumed). Finally, by concentrating on static networks in the first phase in the search for ultimate routing protocol, more efforts are made toward some important properties of routing algorithms, namely loop-free design and flooding rates. These important characteristics seem to be insufficiently studied in [BCSW, KV]. Moreover, we consider several node sizes, and introduce network degree (that is, average number of neighbors of each node) as the independent variable instead of the radius of unit graph. The degree is a much clearer measure of graph density or connectivity than the radius, and is also listed as one of main network parameters in [MC]. The routing algorithm is expected to provide good delivery rates, short hop counts and small flooding rates. Therefore, the basic routing algorithms are filtered first on static networks, and then combined with location update schemes (which include sending control messages) to give GPS based routing protocols.

2. Known GPS based routing methods

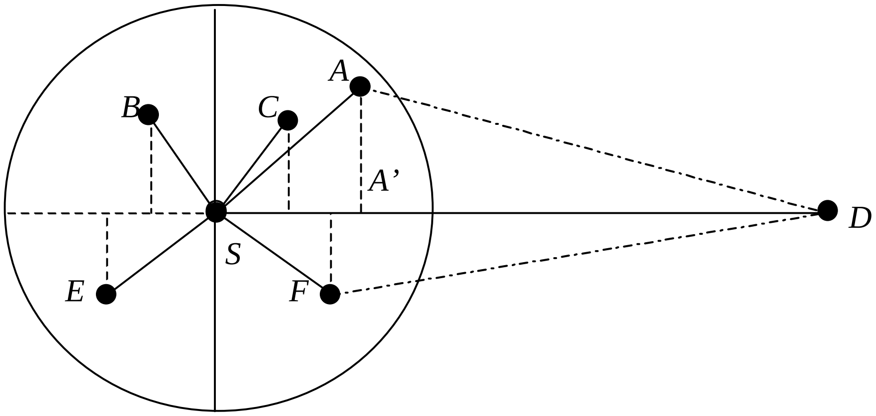


Figure 1. Progress based routing methods

Several GPS based methods were proposed in 1984-86 by using the notion of progress. Define progress as the distance between the transmitting node and receiving node projected onto a line drawn from transmitter toward the final destination. A neighbor is in forward direction if the progress is positive (for example, for transmitting node *S* and receiving nodes *A*, *C* and *F* in Fig. 1); otherwise it is said to be in backward direction (e.g. nodes *B* and *E* in Fig. 1). In the random progress method [NK], packets destined toward *D* are routed with equal probability towards one intermediate neighboring node that has positive progress. The rationale for the method is that, if all nodes are sending packets frequently, probability of collision grows with the distance between nodes (assuming that the transmission power is adjusted to the minimal possible), and thus there is a trade-off between the progress and transmission success. In [HL], packet is sent to the nearest neighboring node with forward progress (for instance, to node *C* in Fig. 1).

Takagi and Kleinrock [TK] proposed *MFR* (most forward within radius) routing algorithm, in which packet is sent to the neighbor with the greatest progress (e.g. node *A* in Fig. 1). In [HL], the method is modified by proposing to adjust the transmission power to the distance between the two nodes. We shall reformulate the *MFR* method in order to facilitate its implementation and provide a simple proof that it is loop-free. Let $a \cdot b$ denote the dot products of vectors *a* and *b*. Consider the dot products of vectors originating from destination *D* and ending at nodes in MANET. Clearly $DS \cdot DA = |DS||DA'|$ where *A'* is the projection of *A* on the line *DA* (see Figure 1). The sign is assumed here to be positive; it can be shown that, in case of negative dot product, *D* must be a neighbor of *S*. Thus the considered dot product is minimal exactly when the progress in maximal. The goal in the *MFR* algorithm [TK] is, therefore, to minimize the dot product. Note that the node that minimizes the dot product (the selected node) may not have a forward progress. Using the dot product definition, we shall prove, in the next section, that the *MFR* algorithm is loop-free.

Recently, three articles [BCSW, KV, KSU] independently reported variations of fully distributed routing protocols based on direction of destination. In these directional routing methods, node *A* uses the location information for *B* and its one hop neighbors to obtain *B*'s direction, and then transmits a message *m* to several neighbors whose direction (looking from *A*) is closest to the direction of *D*. The methods differ in the choice of direction ranges.

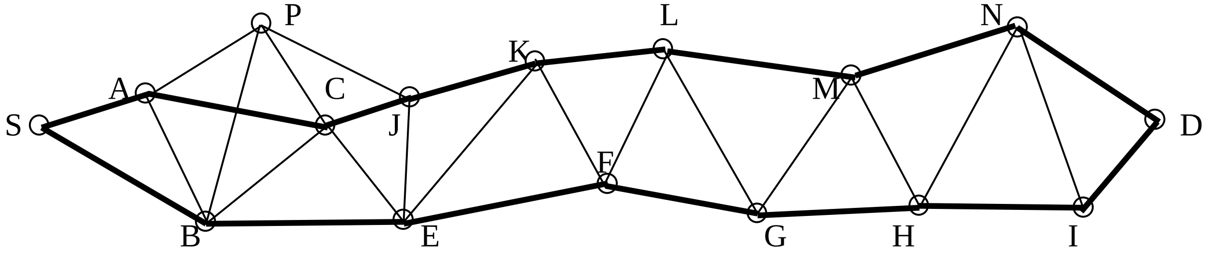


Figure 2. Paths selected by *DIR (SACJKLMND)* and *GEDIR (SBEFGHID)* algorithms

In the *compass routing* method (referred here also as the *DIR* method) proposed by Kranakis, Singh and Urrutia [KSU], the source or intermediate node *A* uses the location information for the destination *D* to calculate its direction. The location of one hop neighbors of *A* is used to determine for which of them, say *C*, is the direction *AC* closest to the direction of *AD* (that is, the angle *CAD* is minimized). The message *m* is forwarded to *C*. This process repeats until the destination is, hopefully, reached. Consider MANET on Fig. 2, where the radius is equal to edge *EF*. The direction *AC* is closest to direction *AD* among candidate directions *AS, AB, AC,* and *AP*. The path selected by *DIR* method is *SACJKLMND* and consist of eight hops. Although the authors describe their method for static networks only (for finding routes using only compass and the geographic road maps), it may be used also in ad hoc networks. They gave a counterexample showing that the compass routing is not loop-free even for static networks modeled by planar graphs embedded in plane (geometric graphs, which differ from unit graphs). The authors modify their algorithm to avoid loops and guarantee delivery for the special case of planar graphs with convex regions and few other cases, which do not correspond to realistic ad hoc networks.

Basagni, Chlamtac, Syrotiuk and Woodward [BCSW] described a distance routing effect algorithm for mobility (DREAM). The source or any intermediate node *A* calculates the direction of destination *D* and, based on the mobility information about *D*, chooses an angular range. The message *m* is forwarded to all neighbors whose direction belongs to the selected range. The range is determined by the tangents from *A* to the circle centered at *D* and with radius equal to a maximal possible movement of *D* since the last location update. The area containing the circle and two tangents is referred as the request zone in [KV]. DREAM algorithm [BCSW] incorporates the idea of triggering the sending of location updates by the moving nodes autonomously at a rate and hop distance that correspond to the node's mobility rate. Ko and Vaidya [KV] described, independently at the same conference, an almost identical algorithm, and a few modifications of it. In the location aided routing *(LAR)* algorithm [KV], the request zone is fixed from the source, and a node which is not in the request zone does not forward a route request to its neighbors. If the source has no neighbors within the request zone, the zone is expanded to include some. The size of the request zone depends on the average speed of the destination's movement and time elapsed since the last known location of the destination was recorded [BCSW, KV].

The definition of the request zone [BCSW, KV] was modified in [MS] in order to provide uniform framework with the corresponding notions in *GEDIR* and *MFR* methods. [MS] discusses the *V-GEDIR*, *CH-MFR* and *R-DIR* methods, in which *m* is forwarded to exactly those neighbors which may be best choices for a possible position of destination (using the appropriate criterion). The request zone in *R-DIR* method [MS] may include one or two neighbors that are outside of angular range, because they can have the closest direction for the tangents to the circle. In *V-GEDIR* method, these neighbors are determined by intersecting the Voronoi diagram of neighbors with the circle (or rectangle) of possible positions of destination, while the portion of the convex hull of neighboring nodes is analogously used in the *CH-MFR* method.

Ko and Vaidya [KV] discussed various enhancements to their basic technique. The *LAR scheme 1* [KV] proposes an alternative definition of the request zone, as the smallest rectangle that includes current location of *S* and the expected zone of destination (a circular region). The request zone is thus increased, with increased chances of reaching destination but also with increased flooding. The modifications in [KV] include sending route requests before the message itself [JM]. Note that a route request may be considered as a routing of short messages. Nodes may update their location information with each exchange of messages between them. Messages may contain source location also to update location information at intermediate nodes. Recovery procedures based on partial or full flooding, to start flooding if the given algorithm fails to find the route within a timeout interval, are proposed by both papers [BCSW, KV].

Ko and Vaidya [KV] also proposed the *LAR scheme 2*. In this scheme, the source or each intermediate node *A* will forward the message to all nodes that are closer to the destination than *A* is (more precisely, at most δ farther from the destination than node *A*, to account for possible location error). This scheme therefore suggests the use of geographic distance instead of direction.

The routing algorithms in [BCSW, KV] are fully distributed, and robust, since they provide multiple routes. They are also demand-based and adapt well to 'sleep' period operation. Simulation results presented in [BCSW] using a discrete event simulator show that the dynamic source routing protocol [JM] has a 25% to 250% larger end-to-end delay than the *DREAM* protocol. The average number of data bits transmitted per data bits delivered is consistently lower for both *LAR* schemes as compared to flooding [KV]. Therefore adding location information to the routing tables in all nodes resulted in significant improvement in the performance over the existing methods that do not use such information. Despite these advantages, the proposed methods [BCSW, KV] have some drawbacks. They have considerable flooding rates, and the directional methods are shown (in this paper) not to guaranty loop-free paths. This paper attempts to improve on these two measures.

In [CL], routing tables are used which are updated by mobile software agents modeled on ants. Ants are used to collect and disseminate information about nodes' location.

3. Loop-freedom of directional and MFR methods

The authors [BCSW] claim that their algorithm provides loop-free paths (no proof was given). However, Fig. 3 shows a counterexample of a loop that consists of 16 nodes, denoted A_1 to A_{16}, positioned at two close circles centered at the destination D (approximately located at nodes of two regular octagons). The graph is an unit graph with the radius equal to the length of longer edge e.g. A_1A_2 in the loop. Let the source be any node in the loop, e.g. A_1. Node A_i selects node A_{i+1}, i=1,2,3,…,16, to forward the message, because the direction of A_{i+1} is closer to D than the direction of its other neighbor A_{i-1} $(A_{17}=A_1, A_0=A_{16})$. Additional node C can be taken just outside the polygon defined by the loop, near the middle of the larger side e.g. A_5A_6 of the 16-gon. It can be verified that there exist a nonempty region inside the 16-gon (loop), reachable to C but not reachable to any point on the loop. Any node B inside that region can be reached from C and is able to reach the destination. Node C can be selected such that the node A_8 has closer direction to D than C, measured from node A_7 (thus A_7 forwards message to A_8, not to C). The example shows that the loop (indicated by arrows) can be created non-locally, and with static nodes. The nodes on the loop are not able to recognize the loop unless message *id* is memorized (for each forwarded message!).

The example in Fig. 3 is not restricted to the unit graph model of MANET. Clearly, such example may exist in any kind of random network model (models where each edge is selected with certain nonzero probability), in a subgraphs of unit graphs that model MANET with obstacles, or in any model that generalizes unit graphs (e.g. minpower graphs), or in any graph model that includes unit graphs as subgraphs. Finally, static network is special case of a moving network, so the counterexample is valid for ad hoc networks. Thus we have proven the following theorem.

Theorem 1. Any routing algorithm for ad hoc wireless networks in which a node currently holding the message forwards it to its neighbor with closest direction toward destination (and to some other nodes) is not a loop-free algorithm.

We shall now prove that the *MFR* algorithm [TK] is loop-free. Suppose that, on the contrary, there exists a loop in the algorithm. Let $A_1, A_2, \ldots A_n$ be the nodes in the loop, so that A_1 sends the message to A_2, A_2 sends the message to $A_3, \ldots, A_{n-1}$ sends the message to A_n and A_n sends the message to A_1(see Fig. 4). According to the choice of neighbors and the *MFR* algorithm (using the dot product formulation given above) it follows that $DA_n \cdot DA_1 > DA_2 \cdot DA_1$ since the node A_1 selects A_2, not A_n, to forward the message. Therefore $DA_n \cdot DA_1 > DA_1 \cdot DA_2 > DA_2 \cdot DA_3 > \ldots > DA_{n-1} \cdot DA_n > DA_n \cdot DA_1$, which is a contradiction. In order to provide for loop-free method, we assume that, in case of ties for the choice of neighbors, if one of choices is the previous node, the *MFR* algorithm will select that node (that is, it will stop or flood the message).

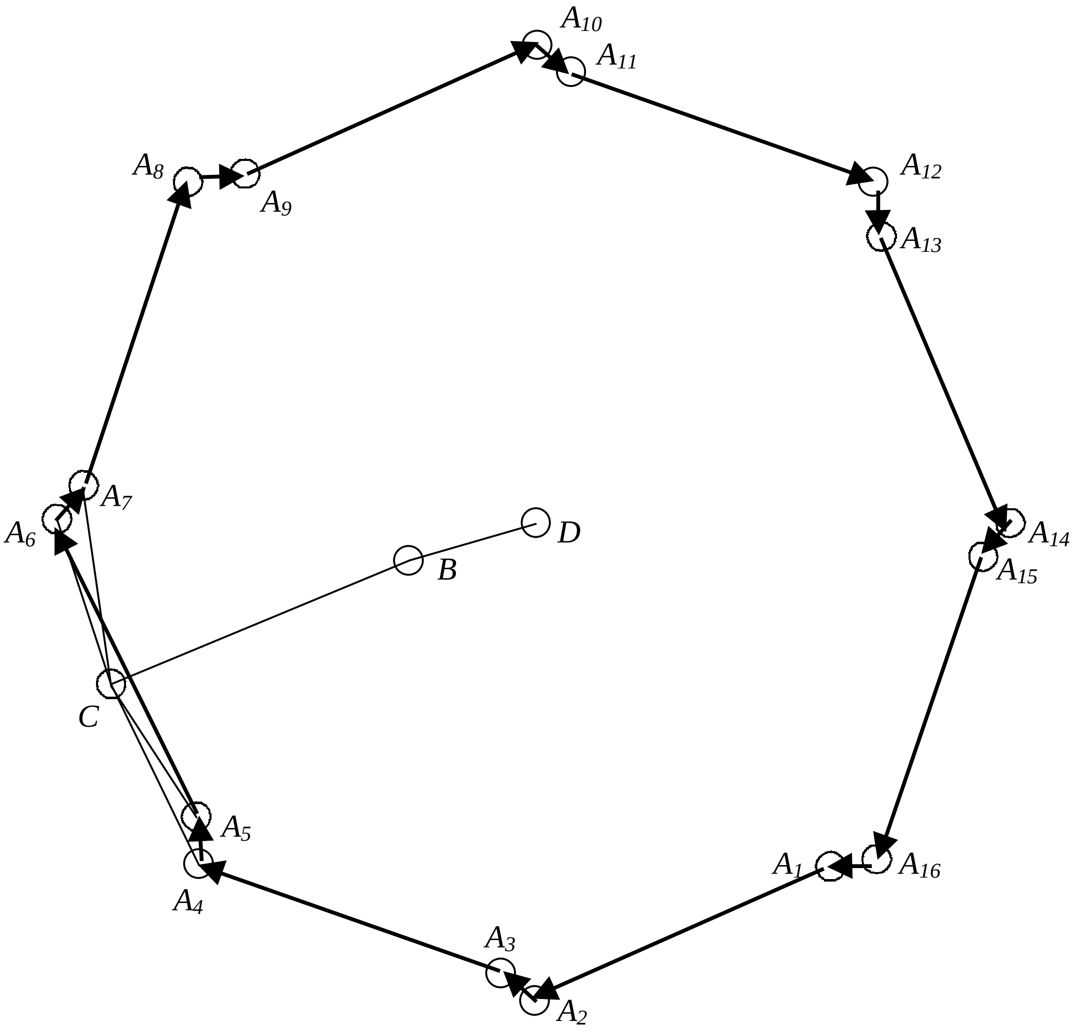
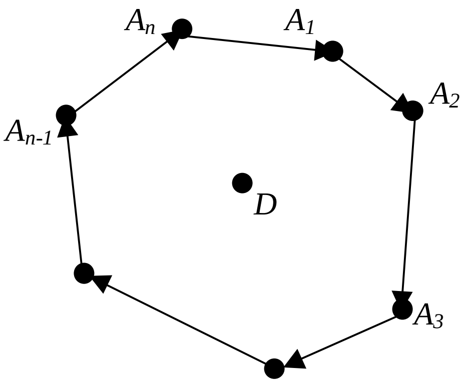


Figure 3. A loop in the directional routing



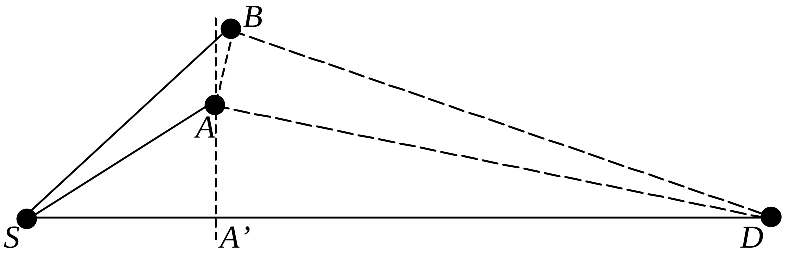


Figure 5. *GEDIR* and *MFR* may choose different node

Figure 4. *MFR* and *GEDIR* algorithms are loop-free

4. Geographic distance routing methods

We introduce a new routing algorithm for a MANET based on the locations (e.g. latitude and longitude) of all nodes. Each node is aware of contains geographic coordinates of all its direct neighbors. The sender node is also aware of the location of the destination, which is forwarded with the message. Node *A* that wants to send a message *m* to destination *D* uses the location information for *D* and for all its one hop neighbors to determine the neighbor *C* which is closest to *D* among all neighbors of *A*. The message is forwarded to *C,* and the same procedure is repeated until *D,* if possible, is eventually reached. The algorithm is, therefore, fully distributed. In example on Fig. 2, sender *S* selects node *B* which is closer to *D* than *A.* The path selected by the algorithm is *SBEFGHID* and consists of seven hops.

Note that, in this basic version, *A* does not compare its own distance against distances of its neighbors. Thus, even if *A* is closer to the destination than *C,* the message is still forwarded to *C,* with the hope that *C* will find another neighbor which is closer to destination than *A* is. Otherwise, *C* will return the message to *A* and a local loop (between *A* and *C*) is created. We will prove that this is the only kind of loop that may be formed in MANET using proposed distance based routing (unless nodes move very fast). Since such loop can be obviously detected by nodes *A* and *C,* they can stop forwarding *m* and prevent it from spinning between them. This simplest version of our algorithm will be referred to as *GEDIR* (GEographic DIstance Routing) algorithm.

The proof that *GEDIR* algorithm is inherently loop-free goes as follows. Suppose that there exists a loop in a distance routing algorithm, and let A_1 be the node on the loop that is closest to the destination (see Fig. 4). According to *GEDIR* algorithm A_1 forwards the message to its neighbor A_2, which then forwards to one of its neighbors, A_3 (following the created loop), which is closest to destination *D* among all neighbors of A_2. Thus A_3 is closer to *D* than A_1 is, which is a contradiction. This proof also suggests that, in case of equal distances from destination, current node should choose the node that forwarded the message to it. For instance, if $|A_1 A_2|=|A_2A_3|$, A_2 should send the message back to A_1, to avoid possible star shaped loop.

Both proofs of loop-free properties (for MFR and GEDIR algorithms) do not refer to the unit graphs and are valid in three-dimensional space. Thus, they are applicable to any model of MANET. The exclusion is, of course, the unrealistic case when nodes move purposely (combined with selected location update scheme) in such a way to maintain a loop (e.g. nodes of a regular polygon moving toward the center (destination) always just before the message is sent to them and returning back afterwards). In the absence of such a purpose, message will exit such a temporary loop, and therefore we have proven the following theorem.

Theorem 2. Routing algorithms in wireless networks in which nodes forward the message to several neighbors closest to destination or with most forward progress (i.e. *MFR* and *GEDIR* algorithms and their enhancements: flooding, 2-hop, multiple path) are inherently loop-free.

In order to provide uniform and fair treatment of all three basic algorithms, we assume that the message is dropped at an intermediate node *A* if the node *C,* selected for forwarding by *A* using the corresponding algorithm, is exactly the node which sent the message to *A* in the previous step. Such a node *A* will be referred to as the *concave* node (in each of corresponding methods). Concave node *A* in *GEDIR* algorithm is therefore a node which is closer to destination *D* than any of its neighbors, and node *C,* the closest to *D* among *A*'s neighbors, has itself no closer (to *D*) neighbor than *A.* Similar definitions, using direction or dot product instead of distance, for corresponding concave node in the compass (i.e. *DIR*) or *MFR* routing algorithms can be given. Thus the stoppage criteria is the same for all three basic algorithms. Since a node which has closer direction may be actually farther away from destination, compass routing may exhibit loops, as shown in Fig. 3. Note that the selected neighbor in *MFR* method may be also farther from the destination, but the loop is never created. It is easy to find examples in which one of basic methods delivers the message to the destination while the others do not. Similarly, it is easy to construct examples in which the path length or number of hops for one method is smaller than for the other methods. Finally, one can construct examples showing that the ratio of hop count by one of algorithms over the shortest hop count may be arbitrarily large.

The delivery rate of *GEDIR*, *DIR* or *MFR* algorithms can be improved if nodes exchange information about their neighbors, and each node is aware of its 2-hop neighbors (neighbors of its neighbors). In this case, node *A* currently holding the message may choose the node closest to the destination *D* among all direct (1-hop) and 2-hop neighbors, and forward the message to its neighbor that is connected to the choice. In case of ties (that is, more than one neighbor connected to the closest 2-hop neighbor), choose the one that is closest to destination. We will refer to this method as *2-hop GEDIR*. *2-hop DIR* and *2-hop MFR* can be similarly defined, by replacing all references to distance by direction and progress, respectively (with respect to *AD*). The abbreviated names *GEDIR-2, DIR-2* and *MFR-2* for 2-hop methods will be also used in the sequel. There are no multiple copies of the message in MANET at any transmission step in these 2-hop methods.

We propose a modification to all three basic algorithms to avoid message dropping. Each algorithm proceeds as described until the message is supposed to be dropped by the corresponding algorithm at a concave node *A*. Modifications differ in the way concave nodes act. If an alternate network is available to the MANET for occasional use (for example, a satellite or other technology), the concave node may use it. Otherwise, we propose flooding as a solution. Full flooding, initiated at a concave node and performed afterwards at any node receiving the message, will certainly suffice to reach the destination, but the flooding rate will be affected. In order to enable this solution, messages should carry a bit of information about existence of a concave node on its previous path, so that receiving nodes may decide how to proceed. We propose to perform flooding only at concave nodes, while every other intermediate node should act with receiving message as in the corresponding basic routing algorithm. After forwarding the packet to all its neighbors, a concave node shall mark packet *id* in the entry corresponding to given destination, and refuse to accept the same packet from any of its neighbors. Upon receiving a rejection message from a concave node, intermediate nodes will select the next best neighbor instead. In effect, the concave node has disconnected itself with respect to given packet. It is not necessary to carry additional flooding bit with the packet. The delivery of the packet to the destination is guaranteed (assuming that MANET is connected graph). The methods will be refereed to as *flooding GEDIR*, *flooding DIR,* and *flooding MFR* routing methods (abbreviated as *f-GEDIR, f-DIR* and *f-MFR*). It is possible to construct examples showing that even full flooding at concave nodes does not guaranty message delivery unless concave nodes reject further copies of the same message.

Next, we propose *c-GEDIR*, *c-DIR* and *c-MFR* methods, in which message is initially sent to *c* neighbors which are closest to destination (whose direction or dot product are best, respectively), and afterwards, on intermediate nodes, it is forwarded to only one neighbor. These methods provide multiple paths (robustness) without much flooding. We shall describe three variants of *c-GEDIR* algorithm (by analogy, same three variants may apply to *c-DIR* and *c-MFR* methods). In the *original c-GEDIR* method, every intermediate node will forward the message to its best neighbor. Thus for *c=1* it is equivalent to basic *GEDIR* method. Although the method may work without memorizing past traffic at each node, many nodes (close to destination) are anticipated to receive multiple copies of the message, and thus we implemented a variant in which every intermediate node will forward only the first received copy of each message. In the *alternate c-GEDIR* algorithm, each intermediate node forwards *i*-th received copy of the same message to *i*-th best (closest to destination) neighbor (for *i*=1, 2, 3, …), disregarding neighbors message came from. Thus concave nodes do not stop transmitting in this method. In the *disjoint c-GEDIR* method, each intermediate node *A*, upon receiving the message, will forward it to its best neighbor among those who never received the same message before. After forwarding the message, node *A* becomes inactive with respect to that message, and rejects further copies of it. The *disjoint c-GEDIR* algorithm attempts to create *c* disjoint paths between source and destination nodes. A node in *alternate* or *disjoint c-GEDIR* method stops forwarding the message if there is no enough neighbors to choose one. Both methods are therefore loop-free although, in the *alternate c-GEDIR*, a message may return few times to the same node.

The improvements mentioned in [BCSW, KV] for their directional algorithms to obtain actual protocols for each of our proposed algorithms can be easily incorporated, giving additional variety to geographically based routing methods. We note that flooding effect may be related to the urgency of the message itself; in other words, messages may have some priority identifiers that will be related to the flooding rate.

5. Performance evaluation

The routing protocols designed in literature are, in most cases, evaluated by using a discrete event simulator on certain kind of graphs, with particular parameter values (e.g. topological rate of change, various traffic patterns, mobility patterns, fraction and frequency of sleeping nodes [MC]). While such evaluation is an ultimate goal for GPS based routing protocols, the scope of our paper is to study candidate GPS based routing algorithms that will serve as a basis for the design of routing protocols. In the presence of a number of possible algorithms that we proposed, the performance evaluation should begin with the case of static nodes, for which routing does not require control messages. After the best algorithms are filtered, each of them may be combined with few different methods for sending control messages to determine the best GPS based protocol.

It is unlikely to expect that one routing protocol for MANET is the best approach for all networking contexts. According to [MC], parameters that define a networking context, in case of static network with nodes of equal range and capacity, are network size *n* (the number of nodes), and network degree (i.e. connectivity) *d*. Our experiments were designed to compare all methods in terms of their average delivery rates, hop counts and flooding rates. The Dijkstra's shortest path algorithm *(SP)* was used as a benchmark (it was also used to test whether a graph is connected).

The experiments were carried using random unit graphs. 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. Generated graphs which were disconnected are ignored.

The first test series evaluated the performance of basic, *2-hop* and *flooding GEDIR, DIR* and *MFR* methods. For each selected pair *(n,d),* a total of 20 connected graphs are generated. We experimented with the following network sizes: *n*= 20, 50, and 100. For *n*= 20, the average degrees tested were *d*= 2, 3, 4 and 5; for *n*= 50, *d* ranges between 4 and 8; and for *n*= 100, *d* is between 4 and 14. For each graph, 50 random source-destination pairs are chosen, and the routing was performed in both directions (thus 100 routing tasks per graph). Averages over all 20 graphs with the same parameters are then found. We shall present here only some of results for *n=100*. Complete results, including more measurements, and results for *n=20* and *n=50*, will be published in the master thesis of the first author.

LAR2 scheme from [KV] is added in the experiments, since it had best performance among schemes proposed by the same authors, according to their measurements. In one transmission step (of broadcast type), the source or each intermediate node *A* will forward the message to all nodes that are closer to the destination than *A* is (thus we selected value $\delta=0$). Authors did not mention whether nodes memorize messages to reduce flooding rate. Experiments in [CL] compared ants based method with *LAR2* without memorizing past traffic and reported flooding ratio in *LAR2* over thousand times higher than in ant based method. We therefore assumed that nodes in *LAR2* do memorize messages and do not transmit the same message more than ones. Nodes in *LAR2* which have no closer neighbor to destination than themselves do not retransmit the message. Thus the flooding rate in *LAR2* is simply the ratio of nodes that transmit the message. Possible message collisions in *LAR2,* flooding and multiple path methods are ignored in our experiments.

Degree	4	5	6	7	8	9	10
SP	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
GEDIR	49.70%	61.55%	77.30%	81.40%	90.05%	92.25%	96.80%
DIR	50.60%	63.95%	79.10%	83.35%	91.20%	93.20%	97.05%
MFR	49.50%	62.30%	78.40%	82.45%	90.50%	92.85%	96.20%
GEDIR-2	57.90%	71.85%	84.90%	87.15%	93.75%	94.75%	98.05%
DIR-2	49.70%	60.05%	75.30%	75.90%	85.25%	86.75%	91.10%
MFR-2	60.45%	73.45%	86.80%	89.10%	94.25%	95.50%	98.15%
f-GEDIR	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
f-DIR	99.75%	99.75%	99.70%	99.80%	99.95%	100.00%	100.00%
f-MFR	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
LAR2	77.50%	89.75%	95.45%	98.05%	98.75%	99.00%	99.65%

Table 1. Delivery rates for *n*=100 nodes

Table 1 shows the delivery rates for *n*=100 nodes. The success rates for *DIR*, *GEDIR* and *MFR* methods are comparable (about 50% for *d*=4, 62-64% for *d*=5, 77-79% for *d*=6, 81-83% for *d*=7, about 90%, 93% and 97% for *d*=8, 9, 10, respectively). Thus success rate greatly depends on network degree but not much on basic method selected! While the success rate for the very basic method on high degree network is already impressive (over 90%), very low degree networks require enhancements to basic methods (e.g. half messages not delivered for *d*=4). 2-hop *GEDIR (GEDIR-2)* and *MFR-2* have increased their success rates compared to 1-hop variants (by 7-10% for low degrees, 1% for high degrees) while *2-DIR* decreased its success rate for 1-8% compared to *DIR*. The reason for success drop for *2-DIR* method is that a 2-hop neighbor *C* of *A* with closest direction *AC* with respect to *AD* may be very far from optimal direction with respect to *BD* where *B* is the common neighbor of *A* and *C*. *f-GEDIR* and *f-MFR* have *100%* success as expected, while *f-DIR* may fail (due to possible undetected loop creation). *LAR2* method did not offer reliable success at low degrees (78% for *d*=4) and was inferior to flooding methods.

Table 2 presents average hop counts for methods studied. They are calculated as the sum of hop counts for all the successful transmissions over total number of successful transmissions, for each individual method. For methods where a message can be delivered several times, the copy with shortest hop count is considered. SP method does not give smallest numbers in the table because it provides longer paths in cases where other methods fail. The hop counts for *DIR* based methods are consistently (but not significantly) higher than those for *GEDIR* and *MFR* methods. Similar results were obtained for other cases. *GEDIR* and *MFR* methods have shown consistently close success rates and hop counts in all cases. The differences in both the success rates and hop counts were less than 1% on the average, with no difference for many of graphs considered. When there was a difference, it appears that one of them was a 'winner' by a random choice, with slight overall advantage in favor of *GEDIR* method. A closer analysis reveals the reason why the path selected by *GEDIR* and *MFR* methods were identical in more than 99% of cases. Consider Fig. 5. Let *A* and *B* be two different nodes selected by the *GEDIR* and *MFR* methods, respectively, when packet is to be forwarded from node *S*. Suppose that they are located on the same side of *SD.* $|AD|<|BD|$, since *GEDIR* selects *A.* Node *B* cannot be selected within triangle *SAA'* where *A'* is the projection of node *A* on direction *SD,* since *B* has more progress than *A*. However, the angle *SAB* is then obtuse, and $|SB|>|SA|$. Since *A* and *B* are likely to be close to each other, the remaining path may coincide, or at least the chances for delivery are similar. However, when *A* and *B* are on the opposite sides of *SD* then a difference in success or hop count is more likely.

Degree	4	5	6	7	8	9	10
SP	8.78	7.12	5.82	5.25	4.48	4.35	3.90
GEDIR	5.72	5.53	5.13	4.72	4.29	4.19	3.87
DIR	6.03	5.92	5.55	5.13	4.63	4.55	4.17
MFR	5.75	5.61	5.16	4.78	4.33	4.23	3.89
GEDIR-2	6.23	5.86	5.29	4.81	4.31	4.20	3.87
DIR-2	6.10	5.82	5.55	4.92	4.54	4.41	4.05
MFR-2	6.40	5.93	5.36	4.91	4.36	4.24	3.88
f-GEDIR	12.59	9.55	7.22	6.39	5.01	4.83	4.12
f-DIR	12.75	9.74	7.42	6.57	5.28	5.11	4.41
f-MFR	12.55	9.50	7.17	6.38	5.01	4.82	4.14
LAR2	7.51	6.65	5.65	5.20	4.47	4.31	3.90

Table 2. Hop counts for *n*=100 nodes

When compared to the shortest path algorithm, (1-hop) *GEDIR/MFR* methods have shown encouraging results (taking into account that they are just basic methods that involve no flooding effect). Their success rate for *n=20* nodes was about 67% for *d*=2, 81% for *d=3, 89%* for *d=4, 94%* for *d=5*. For *n=50* the success rate was about 69% for *d*=4, *80%* for *d=5, 87%* for *d=6, 91%* for *d=7* and *94%* for *d=8.* The hop counts for *GEDIR/MFR* are comparable to hop counts in *SP*.

Degree	4	5	6	7	8	9	10
SP	1	1	1	1	1	1	1
GEDIR	0.56	0.70	0.83	0.87	0.93	0.96	0.99
DIR	0.59	0.76	0.91	0.95	1.01	1.04	1.07
MFR	0.57	0.72	0.84	0.89	0.95	0.97	0.99
GEDIR-2	0.62	0.77	0.88	0.90	0.95	0.96	0.99
DIR-2	0.58	0.72	0.87	0.88	0.96	0.97	1.01
MFR-2	0.65	0.78	0.90	0.92	0.96	0.97	1.00
f-GEDIR	4.87	4.46	3.11	2.95	1.96	1.69	1.32
f-DIR	4.72	4.12	3.00	2.62	1.91	1.73	1.39
f-MFR	4.79	4.52	3.03	2.88	1.94	1.66	1.42
LAR2	1.75	2.80	4.34	5.34	6.81	7.96	9.46

Table 3. Flooding rates for *n*=100 nodes

Table 3 shows flooding rates for each method for *n*=100 nodes. Both successful and unsuccessful deliveries are considered. Numbers less than 1 in many cases are obtained because concave nodes are detected much sooner than destination in *SP* method for the same routing tasks. In order to provide fair comparison with *LAR2* method, all nodes in flooding methods were assumed to memorize past traffic and do not forward the same message twice. This modification had no impact on success rates and hop counts. Flooding based methods, which guaranty delivery (*f-GEDIR* and *f-MFR*) did not significantly flood the network with higher degrees (<2 for *d*=8, 9, 10; between 5 and 10% of nodes are flooded), while for low degrees the effect was notable (>4 for *d*=4 and 5; up to 40% of nodes were flooded). *LAR2* method had the reverse effect. The flooding rate increased significantly with the degree (from about twice *SP* flooding at *d*=4 or 15% of nodes to >9 at *d*=10 and about 14 at *d*=14 or over 40% of nodes). Without memorizing messages, the flooding rates of *LAR2* would be much higher (they would increase $O(d^2)$ times). Let us compare *LAR2* methods with flooding based ones. *f-GEDIR* and *f-MFR* methods guaranty delivery, require less memory (only concave nodes need to memorize messages), and have significantly lower flooding rates at moderate and high degrees (from *d*=6 for *n*=100). *LAR2* has lower hop counts, but the difference is significant only for small degree networks. Thus our flooding based methods are superior to *LAR2* for higher degree networks, while guaranteed delivery offers satisfactory compensation for higher flooding rate for lower degree networks. We have also measured how many neighbors of destination would deliver message to it, and established that the number is 1 or very close to 1 for all methods except for *LAR2*, for which that number is > *d/2*.

Table 4 presents experimental results on delivery rates of multiple path methods for *n*=100 and *d*=6. The improvements obtained by adding multiple paths are notable, but less than anticipated. The success rate increases by about 3-5% from *c*=1 to *c*=2, by additional 2% from *c*=2 to *c*=3, and by 1% from *c*=3 to *c*=4. Alternative methods have about 5% higher success rates than original ones for all *c* values. Disjoint methods have about 15-17% better success rate than the corresponding original ones, for all values of *c*. Similar results were obtained for *n*=100 and *d*=4, 5, and 7. It is worth to note that disjoint methods achieve almost same success rate as *LAR2* even at *c*=1 value, and involve almost no unnecessary flooding.

Table 5 presents hop counts for multiple path methods. Alternate methods have slight hop count increase while disjoint methods have about one extra hop, compared to original methods. Table 6 gives the corresponding flooding rates, with numbers around *c*, which is expected.

C Value	1	2	3	4
SP	100.00%	100.00%	100.00%	100.00%
orig. GEDIR	77.30%	80.70%	81.95%	82.70%
orig. DIR	79.10%	81.60%	83.00%	83.90%
orig. MFR	78.40%	81.70%	83.00%	83.70%
alt. GEDIR	80.70%	86.05%	87.65%	88.10%
alt. DIR	82.85%	86.95%	88.65%	89.10%
alt. MFR	81.70%	86.55%	87.85%	88.35%
disj. GEDIR	92.10%	96.20%	97.55%	97.80%
disj. DIR	90.90%	95.10%	96.90%	97.30%
disj. MFR	92.25%	96.10%	97.75%	98.00%

Table 4. Delivery rates for multiple path methods for *n*=100 and *d*=6

C Value	1	2	3	4
SP	5.816	5.816	5.816	5.816
orig. GEDIR	5.1285	5.173	5.1985	5.2105
orig. DIR	5.5515	5.454	5.4735	5.4885
orig. MFR	5.161	5.1995	5.227	5.2465
alt. GEDIR	5.411	5.5535	5.6035	5.596
alt. DIR	5.947	5.9295	5.932	5.8995
alt. MFR	5.444	5.5665	5.573	5.576
disj. GEDIR	6.447	6.2055	6.087	6.057
disj. DIR	6.628	6.303	6.232	6.1925
disj. MFR	6.348	6.134	6.093	6.0525

Table 5. Hop counts for multiple path methods for n=100 and d=6

C Value	1	2	3	4
SP	1	1	1	1
orig.c_GEDIR	0.833	1.1905	1.4345	1.6225
orig.c_DIR	0.909	1.2035	1.459	1.645
orig.c_MFR	0.8445	1.214	1.4635	1.6545
alt.c_GEDIR	1.0425	2.171	3.2345	4.0165
alt.c_DIR	1.1375	2.4175	3.5655	4.3875
alt.c_MFR	1.048	2.1925	3.253	4.0555
disj.c_GEDIR	1.1625	2.481	3.6375	4.522
disj.c_DIR	1.194	2.45	3.59	4.381
disj.c_MFR	1.142	2.4555	3.608	4.4935

Table 6. Flooding rates for multiple path methods for n=100 and d=6

Conclusion

The proposed demand-based distributed algorithms operate in the same manner if some nodes are in the 'sleep' mode. The only modification is to include a condition at each node to ignore its neighbors that are temporarily not receiving messages. If nodes that are in the 'sleep' mode are actual destinations, the messages for them should be stored until they are ready to receive them.

The obtained experimental results show that *DIR* method does perform well in practice, as claimed in [BCSW, KV], and its superiority to non-GPS based methods is therefore not surprising. However, we showed that it can be further improved in various ways. For instance, *DIR* method is not loop-free while *GEDIR* and *MFR* are loop-free. Hop counts for later two methods were slightly better for all graphs, while success rates were comparable. The *GEDIR* and *MFR* algorithms, on the other hand, differed by less than 1% on each metric and routing method. If one of them is to be selected, *GEDIR* has a slight advantage in its conceptual simplicity and in using shorter edges on average, which may provide some power savings [SL] and somewhat fewer transmission conflicts.

Similarly, we have shown overall superiority of flooding based methods (*f-GEDIR* and *f-MFR*) over *LAR2*. They guarantee delivery and require less memorization. Their flooding rates are superior for moderate and higher degree graphs. While flooding rates of *LAR2* is lower for lower degree graphs, their failure rate is also significant for these graphs. Thus the choice of flooding based methods even for lower degree graphs is justified by the guaranteed delivery. Full flooding at concave nodes may be replaced by a kind of controlled one, but we were unable to find a way that still guarantees delivery. The advantage of *LAR2* might only be only multipath provision. However, our experiments show that *c-GEDIR* and *c-MFR* may provide multiple paths with comparable success rates and a much smaller flooding rates. Moreover, our experiments clearly show that multiple paths do not add much to success rates. Second path improves success by only about 5%, while additional paths add only 1% each. Adding memory seems to have more impact, especially for disjoint methods that achieve similar success rates as *LAR2* even for the one path case.

The search for distributed routing methods that have excellent delivery rates, short hop counts, small flooding ratios and power efficiency is far from over even for the case of static nodes. 2-hop variants of flooding or multiple path methods may be studied. Since the battery power is not expected to increase significantly in the future [SWR] and the ad hoc networks, on the other hand, are booming, power aware routing schemes need further investigation. We prepared a separate paper on the subject [SL]. Next, [BMSU] designed a routing algorithm that guarantees the message delivery in unit graphs without the use of any flooding based approach or any memorization technique at the nodes (the same assumptions as in *GEDIR* algorithm). The delivery rate of several routing algorithms is improved in [S2] by ignoring non-intermediate nodes in routing decisions. Finally, [S1] presented routing algorithms that are also suitable for geocasting. Further research is then needed to identify the best GPS based routing protocols for various network contexts. These contexts include nodes positioned in three-dimensional space and obstacles, nodes with unequal transmission powers, or networks with unidirectional links. Simulations with moving nodes is, of course, the ultimate goal. Experiments with static networks will provide best candidates for the design of routing protocols in mobile networks. The candidate methods that perform best for static nodes shall be combined with known and novel control messages schemes for location updates to obtain improved GPS based routing protocols.

Acknowledgement

This research is partially supported by NSERC.

References

[BCSW] S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), Proc. MOBICOM, 1998, 76-84.

[BCS] S. Basagni, I. Chlamtac, V.R. Syrotiuk, Dynamic source routing for ad hoc networks using the global positioning system, Proc. IEEE Wireless Communications and Networking Conference, New Orleans, September 1999.

[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.

[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, to appear.

[CL] D. Camara and A.F. Loureiro, A novel routing algorithm for ad hoc networks, Proc. HICSS, Hawaii, January 2000, to appear.

[EGHP] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, Proc. MOBICOM, 1999, Seattle, 263-270.

[EMMB] R. Estrada, D. Munoz-Rodriquez, C. Molina, K. Basu, Cellular position location techniques, a parameter detection approach, Proc. 49th IEEE Int. Vehicular technology Conference VTC'99, Houston, May 1999, 1166-1171.

[HCB] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient routing protocols for wireless microsensor networks, Proc. HICSS, Hawaii, January 2000, to appear.

[HKB] W.R. Heinzelman, J. Kulik and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, Proc. MOBICOM, 1999, Seattle, 174-185.

[HL] T.C. Hou and V.O.K. Li, Transmission range control in multihop packet radio networks, IEEE Transactions on Communications, 34, 1, 1986, 38-44.

[IETF] IETF Manet charter, http://www.ietf.org/html.charters/manet-charter.html .

[JM] D. Johnson, D. A. Maltz, Dynamic source routing in ad hoc wireless networks, in Mobile Computing (T. Imielinski and H. Korth, eds.), Kluwer Acad. Publ., 1996.

[K] E.D. Kaplan (ed.) Understanding GPS: Principles and Applications, Artech House, 1996.

[KKP] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: Mobile networking for 'smart dust', Proc. MOBICOM, 1999, Seattle, 271-278.

[KSU] E. Kranakis, H. Singh and J. Urrutia, Compass routing on geometric networks, Proc. 11th Canadian Conference on Computational Geometry, Vancouver, August, 1999.

[KV] Y.B. Ko and N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, MOBICOM, 1998, 66-75.

[MC] J.P. Macker and M.S. Corson, Mobile ad hoc networking and the IETF, ACM Mobile Computing and Communications Review, 2, 1, 1998, 9-14.

[N] NAVSTAR GPS operations, http://tycho.usno.navy.mil/gpsinfo.html, and Iowa State University GPS page, http://www.cnde.iastate.edu/gps.html .

[NI] J.C. Navas, T. Imielinski, GeoCast -geographic addressing and routing, MOBICOM, 1997, 66-76.

[NK] R. Nelson and L. Kleinrock, The spatial capacity of a slotted ALOHA multihop packet radio network with capture, IEEE Transactions on Communications, 32, 6, 1984, 684-694.

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

[SL] I. Stojmenovic and Xu Lin, Power-aware routing in ad hoc wireless networks, SITE, University of Ottawa, TR-98-11, December 1998.

[S1] I. Stojmenovic, Geocasting and routing in mobile ad hoc networks, manuscript, 1999.

[S2] I. Stojmenovic, Location and internal nodes based routing with guaranteed delivery in wireless networks, 1999.

[SWR] S. Singh, M. Woo, C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, Proc. MOBICOM, 1998, 181-190.

[TK] H. Takagi and L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Transactions on Communications, 32, 3, 1984, 246-257.