

A routing strategy and quorum based location update scheme for ad hoc wireless networks

Ivan Stojmenovic and Bosko Vukojevic

Computer Science, SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada

Ivan@site.uottawa.ca

Abstract

All previously proposed position based routing algorithms for wireless ad hoc networks were based on forwarding the actual message along multiple paths toward an area where destination is hopefully located. The significant communication overhead can be avoided if the routing strategy is changed. We propose that the source node issues several search 'tickets' (each ticket is a 'short' message containing sender's id and location, destination's best known location and time that location is reported, and constant amount of additional information) that will look for the exact position of destination node. When the first ticket arrives at the destination node D , D will report back to source with brief message containing its exact location, and possibly creating a route for the source. The source node then sends full data message ('long' message) toward exact location of destination. We then propose a quorum based location update scheme, in which nodes report their new positions to their neighbors whenever a link is broken or created. After certain number of such link changes, nodes forward their new position to all nodes located to the north and south of their current location with certain 'thickness' of reporting. The destination search then begins with two tickets being sent in the east and west direction, with certain 'thickness', looking for the most up to date information of destination's position. When the tickets reach each end of current 'row', the search is continued toward best reported destination position, with corrections along the path as better information becomes available closer to destination. One of tickets can be sent from the source directly toward destination, to take advantage of possibly correct information on destination location. We show through simulation that the proposed routing and location update schemes provide high success rates with reasonable communication overhead.

1. Introduction

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 low power and small size cards will be deployed in each car and possibly in every user terminal. For instance, Differential GPS offers accuracy of a few meters [N].

Mobile ad hoc networks consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure. Routes between two hosts in network may consist of hops through other hosts in the network. The task of finding and maintaining routes in the network is nontrivial since host mobility causes frequent unpredictable

topological changes. A number of 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]. 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, 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.

Ad hoc networks consist of autonomous nodes that run their routines in asynchronous fashion. The communication algorithms between nodes are therefore all distributed. However, Estrin, Govindan, Heidemann, and Kumar [EGHK] defined the class of so called *localized* algorithms, as distributed algorithms where simple local node behavior achieves a desired global objective. Localized algorithms therefore resemble the class of greedy sequential algorithms. They argued that localized algorithms may be necessary for sensor and ad hoc network coordination, and described localized algorithms for clustering and object location. We propose a more formal definition of localized algorithms. In a localized algorithm, each node makes the decision to which neighbor(s) to forward the

message based solely on the location of itself, its neighboring nodes, and a *constant* amount of additional information. In addition, each node is allowed to perform local computation (since it is equipped with a small but powerful processor). The decision depends on the task performed and the algorithm followed, which is part of an incoming message. In case of routing, the additional information needed is the location of the destination. The sender uses the latest available location about the position of the destination and attaches it to the message. Intermediate nodes may use their more recent location information to replace the one that comes with the message (and also to update their own information otherwise). The other six problems may not need any additional information (other than the one that arrives as part of the message) to perform a localized algorithm. Note that the requirement for the location of a destination means that each node should update locations of all other nodes in the network, so that any routing that is initiated will have a reasonably good start. However, the path adjustments can be made as the message travels closer to the destination. *Non-localized* algorithms, on the other hand, typically require the knowledge of the locations of all nodes in the network, and also the information about the existence of every edge in the graph. All non-localized routing algorithms proposed in literature are variations of shortest weighted path algorithm (e.g. [CN, LL]). Although ad hoc network is fairly accurately modeled by unit graphs, 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 but power demanding link). Thus the use of non-localized algorithms requires the regular update of existing edges in addition to nodes location, which is a quadratic (in number of nodes) communication overhead requiring considerable bandwidth and battery power. Next, some nodes in the sensor or ad hoc network may be temporarily inactive, and non-localized algorithms need to know which of the nodes are active to make their best decisions. The activity information puts additional demand on the information update. For example, static nodes may need to broadcast such information to the whole network whenever they change their activity status while, at the same time, they have no need to update their location with the rest of the nodes. In order to preserve battery power over long periods of time, nodes may change their activity on a regular basis, or it may depend on the size of the job assigned to them. While the absence of a GPS receiver at nodes forces many problems (routing, for example) to be solved by only non-localized algorithms, their application, when GPS is available, may not be justified if efficient localized alternatives are available. The power savings obtained by applying a localized rather than non-localized algorithm outweighs even the additional power imposed to nodes for carrying and using GPS receiver. This paper deals solely with localized algorithms.

This paper is organized as follows. Section 2 gives a review of localized routing algorithms and location update techniques in ad hoc networks. Section 3 proposes a new routing strategy, which separates the location update, destination search, path creation and data traffic phases in a routing process. Sections 4 and 5 propose new quorum based location update and destination search schemes. Section 6 deals with simulation and experimental results that confirm the efficiency of proposed methods. Future work is discussed in section 7.

2. Literature review

Stojmenovic and Lin [SL] described and discussed basic localized routing algorithms. Source or any intermediate node S will select one of its neighbors A according to a criterion. In the *DIR* method, derived from [BCSW, KV, KSU], the best neighbor has the closest direction (that is, angle) toward the destination. That is, neighbor with minimum angular distance from the imaginary line joining the current node and the destination is selected. *GEDIR* method [SL] selects neighbor A that is closest to the destination D . In *MFR* method [TK], the best neighbor A will minimize the dot product $DA \cdot DS$. Alternatively, one can maximize the dot product $SD \cdot SA$. Each method stops forwarding the message at a node for which the best choice is to return the message back to previous node. *GEDIR* and *MFR* methods are loop-free, while *DIR* method may create loops, unless past traffic is memorized or a timestamp is enforced [SL]. The experiments in [SL, SSW, SL2] are carried on static networks. A routing algorithm that guarantees delivery in unit graphs is described in [BMSU]. Power aware routing is discussed in [SL2]. The concept of internal nodes is introduced in [WL] and significantly improves delivery rates and hop counts in routing algorithms [SSW].

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 [S1] in order to provide uniform framework with the corresponding notions in *GEDIR* and *MFR* methods. [S1] 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 [S1] 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).

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 not to guaranty loop-free paths [SL].

In [CL], Camara and Loureiro used routing tables which are updated by mobile software agents modeled on ants. Ants are used to collect and disseminate information about nodes' location. 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. If past traffic is memorized, experiments in [SL] report 98% success rate for *LAR2* ($n=100$ nodes with average degree $k=7$) with flooding rate (ratio of number of transmissions of a given method compared to a shortest path algorithm) of 5.3. This number assumes correct information about destination and forwarding full messages to it. This is in addition to flooding effect caused by the location update schemes in [KV, BCSW].

Amouris, Papavassiliou and Lu [APL] presented a position based multi-zone routing protocol for wide area mobile ad-hoc networks. Their algorithm is based on position updates within circles of increasing radii. Each node updates its location to all nodes located within circle of radii $P, 2P, 4P, 8P, \dots$ (each subsequent circle has twice larger radius than previous one). Whenever a given node A moves outside one of these circles of radius $2^t P$ for some t , node A broadcasts its location update to all nodes located inside of circle centered at current node position, and with radius $2^{t+1} P$. The routing toward destination then follows these circles of last updates. Source nodes send message toward the last reported position of destination (using the *DIR* method), which since the last report has moved within the circle of some radius. As routing message moves closer to destination, the information about position of destination becomes more precise, and nodes are able to send message toward center of circles with twice smaller radius than

previously, until the node is eventually reached. This method is very interesting and certainly competitive. We observe that the radius of larger circles may encompass almost all nodes of the network, and that the routing paths discovered by the algorithm do not have near optimal hop counts (which may be important in quality of service applications). However, if the path quality is important, one can apply our proposed routing scheme where destination may report back to source with a better path. The authors measured communication overhead for given fixed parameter setting (speed, direction, radius, number of nodes) with respect to increasing network size. No comparison of data traffic versus location update traffic is provided.

Karumanchi, Muralidharan and Prakash [KMP] discussed information dissemination in partitionable mobile ad hoc networks. They study the problem of getting the location of some other node in the network and the surroundings of that node (e.g. firefighter) without the need to route any message to that node. Thus their performance evaluation is limited in measuring the accuracy of the obtained information (i.e. the distance between found and exact location of other node). They use a well known approach for information dissemination by replicating information at multiple nodes acting as repositories, and employing quorum based strategies to update and query information. In [KMP], n nodes are divided into $n^{1/2}$ groups with $n^{1/2}$ nodes in each in two ways (as explained below for any quorum based strategy) and preserve such quorums while nodes move. They also discussed the question when to update location, and argued that distance-based updates (based on absolute distance traveled since last update) and movement-based updates (based on the velocities of nodes) may have limited usefulness in ad-hoc networks (such location updates are used in [BCSW, KV]). For instance, nodes may move within a small circle, causing unnecessary location updates. Karumanchi, Muralidharan and Prakash [KMP] concluded experimentally that the best strategy is to update when a certain pre-specified number of links incident on a node have been established or broken since the last update. We decided to apply this strategy in our paper.

Given a set S of n servers, a quorum system is a set of mutually disjoint subsets of S whose union is S . When one of servers require information from the other, it suffices to query one server from each quorum. It is possible to form quorums of size approximately $n^{1/2}$ [M]. For example, 25 servers can be organized into 5 rows and 5 columns. Each column serves as a quorum. Thus each node (i,j) (located in i -th row and j -th column) replicated its data to all servers (i',j) in its column. To extract the information from server (i,j) , server (i', j') may inquire within its i' -th row, and the server (i',j) will provide requested information. Variations of this scheme are used in fixed networks, where the set of queried servers is bound to contain at least one server that belonged to the quorum that received the latest update. Hence, each query returns the latest value of the queried data. Such a query and update strategies has been previously employed for location management in cellular networks [KAS, PHS, PS]. Modifications of quorum based strategy for use in dynamic or partitionable servers has been considered in [ESC, H, KMP]. The main idea in these papers [ESC, H, KMP] is that each server (or node) selects one of quorums at random, to increase the chance of obtaining relatively up to date information in several 'columns'.

3. A routing strategy for ad hoc networks

The problem of routing in mobile networks is apparently very difficult one, and so far no complete and satisfactory (in terms of almost guaranteed delivery without significant communication overhead) scheme was proposed. We propose to clearly divide the problem into several components, and study each of them separately. All the message traffic related to routing tasks is divided into four components as follows.

- 1) *Location update* messages are initiated by each node, which acts on its movement. Location updates are required by some other tasks as well (e.g. clustering, broadcasting, etc.).
- 2) *Destination search* messages, initiated by a source node, when it wants to route a message toward destination. This idea is similar to the dynamic source routing (e.g. [JM]) applied in ad hoc networks that do not use GPS in their routing decisions. However, it also differs significantly from these methods since they suggest full flooding as a means of searching, while we propose an intelligent search based on location of nodes. Also, we do not need to maintain any kind of routing tables, and believe that position information suffice to make intelligent decisions without them.
- 3) *Path creation* messages, initiated by destination upon receiving the first copy of a search message. The destination learns the location of sender from the search message and is able to find the best path accurately. Since the transmission speed is far greater than node movement speed, the path creation phase in a localized routing may, to a large extent, be considered as the operation performed on a static network. Thus routing algorithms for static networks (with known location of destination), such as those proposed in [KSU, SL, BMSU, SL2, SSW, TK], may be applied for the path creation phase. This assumption is justified since each node maintains the list of neighbors and learns the exact location of destination, which is the only information needed for making a routing decision at each node.
- 4) *Data traffic* messages, initiated by source upon receiving reply from destination containing its exact location, possibly together with the path toward destination (e.g. in a form of next hop routing information at intermediate nodes [PR]). Alternatively, the source may attempt to create another path, knowing destination location accurately, by applying any localized routing algorithm defined on static networks.

In this routing scheme, we may also divide all messages into short and long ones. Short messages do not have the real information (to be forwarded to destination) as part of message (unless it is a very brief message, e.g. alarm), and therefore has much lesser number of bits than the message that contain the real information. Location update, destination search, and path creation messages are short messages. Location update messages are generated independently on routing request, as a preparation for successful destination search. Destination search and path creation messages are generated by routing requests. They are still a communication overhead. When the real message, containing data to be forwarded to destination, is long compared to first three kinds, this routing scheme is justified.

Note that, with this general routing scheme, the routing problem is divided into two components that may be investigated separately, as follows.

Component 1: *Location update and destination search schemes.*

Component 2: *Routing to a destination whose position is known* (includes path creation from destination to the source, and data traffic from source to destination).

Satisfactory localized solutions for path creation and data traffic phases are already proposed in [BCSW, BMSU, KSU, KV, SL, SSW, SL2, S1]. Because of drawbacks of existing solutions for the location updates and destination search schemes, we shall concentrate on these two components in this paper. We shall propose new solutions for them in the next two sections.

The main difference between our proposed location update and destination search strategies and previously proposed analogous solutions (including non-GPS based route discoveries and route maintenance) is that full flooding was previously used as regular technique to construct the route, maintain the route or update the location in many cases. For example, when destination moves extensively but far way from the source, no solution other than full flooding was suggested. We propose to deal with such movement pattern by reducing full flooding to row and column paths of certain thickness.

4. Location update

In this paper, we shall adopt the quorum based idea to enable efficient routing in mobile ad hoc networks. Clearly, nodes in ad hoc network do not stay in the same 'column', and the distributed information may easily disperse due to node movement. Moreover, it is not clear what the 'column' is, and how all the nodes in a column, once defined, will receive latest updates. Nevertheless, we believe that this idea is worth pursuing.

The basic update procedure is performed by each moving node whenever it observes that, due to its movement, an existing edge will be broken (that is, the distance between two nodes becomes $>R$). The node will broadcast a message containing its new location information to all neighbors which are at distance tR . For $t=1$, the radius is same as the radius for transmitting data traffic. In this case, former neighboring node is assumed to have received new location information and will also record broken edge. However, nodes may adjust transmission power, and spend more energy for short messages. Lin and Liu [LL] discussed this difference and even proposed an extreme difference in radii for short and long messages. Nodes in [LL] are able to send their new location to all other nodes in network with a single broadcast (single-hop network for location updates). However, when sending exact data, the network is treated as multi-hop one. Note that the single-hop location update broadcast may fail to reach a number of nodes due to obstacles in the field. Next, those obstacles are not accounted for in the shortest weighted path QoS routing algorithm [CN]. Thus we believe that assuming single-hop network for location updates may not be justified, since it may require too much power from nodes to reach all other nodes (the transmission power is proportional to square, or higher degree, of the transmission radius). However, we still allow larger power for location updates, and may consider location update with radius tR , where t is network parameter. Spending larger power for update may be justified by better

destination search efficiency. Another possible solution is to keep transmission radius at R , but retransmit from each of neighbors few hops away. However, these re-transmissions also require power (from neighboring nodes), and may cause broadcast storm problem [NTCS], and therefore its efficiency is doubtful. We thus assume a single broadcast at distance tR . Each node which is, at the moment of transmission, located inside that circle (of radius tR) is assumed to receive the new location accurately, without acknowledging the message. However, new neighbors will reply to node (using radius R), and other nodes within radius R will also learn updated position of that node.

Each node A will send similar location updates (at radius tR , where t could be same or different parameter) when it discovers new neighbors, according to the last information they have about location of that new neighbor. That new neighbor B will acknowledge with exact location information, if indeed neighbor, and will use R as the transmission radius. If B is not within distance R but within tR , it will respond, by adjusting transmission radius to exact distance, and other nodes within that transmission radius also hear that transmission. If B is not within tR , it will not respond, and A will adjust location information for that node to new position, at distance $(t+1)R$, in direction of old position (keeping old timing information, though).

The main location update method is to forward the new location information (and node's identifier) within a 'column' in the network, in the following way. Each node uses a counter to count the number of previously made changes in edge existence (the number of created or broken edges). When the counter reaches a fixed threshold value e , location information is forwarded along the 'column', and e is reset to 0. Node A uses radius pR , where p is network parameter ('thickness' of column). A initiates two routing messages, in the directions north and south. Each follows variation of the *MFR* algorithm [TK], with destination always to the north of current node, as follows. A first transmits update information over radius pR , and each node within distance pR receives the update. Only one of nodes, the most northern (and similarly the most southern) will retransmit, and is decided as follows. A sends separate message, with radius R , looking for such a node. The message is sent to the northernmost neighbor B of A , and retransmitted in the same fashion. If B has neighbors within pR of A , which are more north than B , B will send the message (also at radius R) to the northernmost among them. This search continues until a node B has no neighbor within pR which is more to the north than B . B will retransmit, with radius pR , and process continues until a node cannot find any neighbor to the north. Note that, for $p=1$, the method can be simplified easily to only one transmission. This process continues toward the northernmost and southernmost nodes. The initiator of location update sends the update in both directions, while other nodes follow only one of directions. Note that, for some reasons (e.g. in graphs of low density), the column may terminate without reaching nodes that are furthest north from A . Increased transmission radius (pR) is proposed to reduce the chances of such an event. Additional techniques (e.g. restrictions to the internal nodes [SSW, WL] may also be applied).

Each node receiving new location information for A , by means of any of transmissions, will record it, together with date of update. Moreover, each node transmitting any kind of message will include its own location with the message, thus providing an update on its own position to its neighbor within the selected transmission radius.

5. Destination search

The search for destination is performed in the following way. The source S broadcasts search request using radius sR , where s is another network parameter. If destination is within sR , it will reply, adjusting its transmission radius to exact distance to sender. The search messages always carry the location of sender (and time), and nodes receiving the information will update sender's location in their tables. If there is no reply, the search continues in east-west direction, in similar way as for the location update (that is, easternmost and westernmost neighbors will retransmit at same radius sR). The search message includes time of last available information, and other nodes are requested to provide more up to date information, if they have. This process goes as follows. Let A be the node that transmitted with radius sR . Each node C , $|AC| < sR$, which has no neighbor F such that $|FA| < sR$ and $|AF| > |CF|$, will start response process within circle of radius sR . If node C has more up to date information about D than A , then it sends it to one of its neighbors that is closest to A (it sends both the time of information and location of D at that time). Other nodes C inside the circle (of radius sR) will wait certain time (proportional to their distance from the boundary of the circle) for reaction of its neighbors that are further from A . If C receives one or few such messages, it will choose the most up to date among them (including its own information), and forward it to its neighbor that is closest to A . If C receives no message, and has no better information, it remains silent. Eventually, node A may receive a better information about location of D . The best information is then forwarded in given direction (eastward or westward) to furthest possible neighbor, using radius R for transmission, and retransmitted similarly until a node is reached that has no neighbor in given direction but is still inside the circle. That neighbor repeats the process by transmitting with higher radius sR .

When easternmost and westernmost nodes are reached, the search strategy changes. The message search is then oriented toward the destination, using latest available information for each search message. There are three searches initiated. The first one originates at sender node S , using the best information collected within circle of radius sR centered as S . This search does not need to wait for the result of searches in the east and west directions. The other two searches are initiated by easternmost and westernmost nodes in a given 'row'. Each of three search tasks follows a path toward destination, using *GEDIR* strategy [SL]. That is, at each step, the neighbor closest to destination is selected to forward the message. Since any of messages between two nodes can be heard by all nodes within radius R , if any of them has a better information about destination, it will react by using power $2R$, so that receiver nodes will also hear it and possibly adjust the algorithm. *GEDIR* algorithm stops if the best neighbor for a node is the neighbor that actually sent the message. Such node, called concave node in [SL], will stop the search algorithm to avoid the creation of local loop. Before terminating the search, that node will make one last attempt to gain better information, and will broadcast destination search message using radius sR . If a better information is received, and new neighbor selected accordingly, the search may continue. Otherwise it terminates.

Note that the location of source node is updated at each node that hears any of destination search or path creation messages.

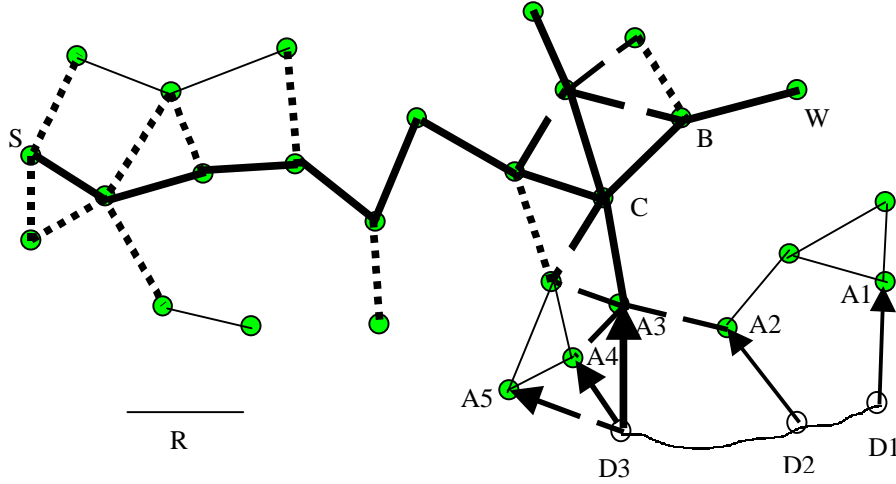


Figure 1. Location update from D3 and destination search from S

Figure 1 illustrates location update and destination search schemes, where we assume $t=p=s=1$. Destination node D moves (other nodes in Fig. 1 are assumed to be static) from positions $D1$ to $D2$ and finally to $D3$, and causes some edges to brake or to be created (indicated by arrows, with nodes $A1$, $A2$, $A3$ and $A4$). At position $D3$, it decides to send location update in its current 'column' by sending its position in the north direction (it does so in the south direction as well, but there is no neighbor in south direction in Fig. 1). The main update path is indicated in bold lines, and bold long dashed lined indicate some other nodes that hear the update message. Source node S initiates destination search in the east-west direction (in Fig. 1 it has only neighbors in west direction). The search path is indicated in bold line, and some other nodes that observe the search are connected by bold short dashed lines. The location update column and destination search row intersect in Fig. 1 in seven nodes. The easternmost and westernmost points in the row, and node S then turn the search toward learned position of D . In Fig. 1, point W learned the up to date information, and may find the destination by applying any localized routing algorithm for static nodes (in Fig. 1, *GEDIR* algorithm would produce path $W-B-C-A3-D3$).

6. Simulation

The parameters in the simulation as follows. The number n of nodes in the network is set to $n=100$. The average node degree k is set to $k=7$. The corresponding value for radius R is found as in [SL] (by sorting all possible edges and choosing R to provide exactly $nk/2$ edges). 20 random unit graphs generated. For each graph, n nodes select their x and y coordinates at random from the interval $[0,100]$. Graphs are not checked for connectivity, since node movement may disconnect each graph and reconnect it frequently. Thus initial connectivity does not secure connectivity during node movements.

The parameters t , s , p for the link update, column update and destination searches are tested for values 1, 2, and 3 each, with all possible 27 variations. The rate e of column (or quorum) update received values $e=3, 5, 10 \dots$

The movement of nodes is similar to the one used in [CN, JM, MZ]. Each node generates a random number *wait* (rounded to nearest integer) in interval $[0..maxwait]$. The node does not move for *wait* seconds. This is called station time. When this time expires, node generates a random number *travel* in interval $[0,maxtravel]$ (rounded to nearest integer), and generates a new position within the same square at random. Node then moves from old position to new position along the line segment joining them at equal speed for the duration of *travel* seconds. Upon arriving at new location, node again chooses waiting period etc. Mobility rate is given by formula $mobrate = maxtravel/(maxtravel+maxwait)$. Parameter *mobrate* received values 10%, 20%, 30%, 50%, 100%, while *maxtravel* was set to 100 seconds.

Routing rate *routrate* has an impact on the communication overheads and success rates. It is defined as the number of routing tasks initiated in each second. The tested values for *routrate* are 1, 5, 10, 20,...

We assume that the time needed for a location update or a destination search is expressed in milliseconds and that nodes do not move significantly during that time. In each second, the following events occur.

- 1) *routrate* routing tasks are performed, one after the other (sequentially). For each of them, source and destination nodes are chosen at random (they must be distinct). Each node has correct list of its neighbors, since the location update between neighboring nodes is regular and occurs whenever topology changes. Thus even if the actual distance is a bit over R (due to small inaccuracies in recorded mutual positions of nodes), we assume that neighboring nodes receive correctly the message.
- 2) All nodes move to a new position.
- 3) Each node that detects a broken or created link, based on the recorded position of its neighbors (neighbors may not be at that position), performs local or even quorum based location update.

No routing tasks are issued in first 100 seconds. In the next 900 seconds, *routrate* routing tasks are issued in each second. The following performance measurements are taken in the experiment. Each value is an average over 20 graphs.

- 1) Success rate, equal to the percentage of successful destination search requests.
- 2) Numbers of transmissions with each of radii R , $2R$, sR , pR , tR mentioned in the scheme, each divided by the number of routing tasks issued altogether, are reported separately. This is communication overhead.

As in [APL], no link details (such as MAC protocol, link errors, frame retransmissions etc.) are modeled here. We assume that no collisions occur in any of simultaneous message transmissions, and all transmitted messages are correctly received (depending only on transmission radius). Thus each node is assumed to be an infinite-buffer, store-and-forward queuing station. Error free transmission in any direction, or in all directions simultaneously, is also assumed.

The obtained data will be included in the paper after the experiments are completed.

The actual routing tasks, and even the reporting back from destination to source, are simulated separately. The same simulation is done in other papers [SL] using static nodes and several localized routing algorithms, but assuming that each graph is connected. The fact that nodes need only local information (position of neighbors) and precise information about destination, and the relative low speed of nodes compared to speed of transmissions, is equivalent to considering routing tasks with static nodes. Our obtained hop counts and success rates for GEDIR method are somewhat worse than reported data, but not significantly (because of possibly disconnected graphs). We did not report the obtained data since we believe that the problem of routing with precise information on destination location requires a separate investigation [SL].

7. Future work

This paper proposed location update and destination search algorithms. Once destination is found, an algorithm described in [BMSU] may be used for path creation and data traffic messages with guaranteed delivery. It is interesting to observe that this proposed scheme contains elements of all three basic routing algorithms: *GEDIR*, *DIR* and *MFR*. *MFR* algorithm is used for location update and first part of destination search, *DIR* algorithm is used in the *FACE* portion of *GFG* algorithm [BMSU] when *GEDIR* fails, and *GEDIR* is used otherwise.

Our location update scheme works well even if nodes move together to a new area (e.g. group of vehicles or soldiers). The columns and rows defined in our location update scheme are relative, not absolute. The only problem could be if the movement is very synchronous (vehicles on a highway), keeping mutual distances but moving at high speed. This case is not dealt with in our experiments, but can be treated separately, by identifying such pattern of movement (this may involve the use of sensors with line of sight direction measurements [EGHK]), and reducing location update to a reasonable level. It is not difficult to incorporate such a change in the actual routing protocol.

Unsuccessful searches for destination may be converted into full flooding at termination nodes, if guaranteed delivery is required. If this event occurs very rarely, it shall have no significant impact on communication overhead. In case of frequent failures, the quorum based strategy may need further improvements. It is easy to construct examples where our 'columns' for update and 'rows' for search do not intersect at all, even for static nodes. Our location update scheme may be improved by sending location update into both row and column containing node, and also by searching both row and column for location of destination. Further 'columns' and 'rows' definitions may also include lines at different angles (other than horizontal and vertical) if the nodes are aware of global network configuration and a different column and row directions are justified (e.g. at angles $\pi/4$). If global network configuration is not known, columns may be chosen in a random direction, perhaps adding orthogonal direction for location updates as well. The destination searches may also proceed in a random direction, and its orthogonal one, to assure a good intersection with recent location updates. A variation of MFR algorithm, where destination is changing so that it points in selected direction from current node, may be applied, to replace the notion of east, west, north and south

directions. Thus, instead of merely comparing x or y coordinates and choosing minimum or maximum one, one can select the maximal dot product $d \cdot SA$ between fixed direction d and the vector joining current node S with candidate node A .

The communication overhead may be reduced, with small impact on success rate, if the updates and searches using 'big' circles (radii pR and sR) are performed from first nodes outside previous 'big' circle, instead of last node located inside.

The concept of internal nodes [SSW, WL] may be used to improve the performance of in destination search, location update and path creation phases. Six definitions of internal nodes are proposed in [SSW]. The simplest of them is defined as follows. A node is intermediate node if there exist two of its neighbors that are not directly connected [WL]. Thus intermediate node belongs to a shortest path between some pairs of nodes. Nodes which are not intermediate may stop forwarding the message, if it is sent to them in any of the phases (of course, such nodes may hear transmissions, but should not be selected as destination by any of the paths in any of the phases). The performance of any of the phase may be greatly improved by restricting the paths to internal nodes only. Longer and more successful path will be created that way.

A variant of the method may choose to send location updates in a different way. Instead of counting the number of the number of link changes, node may act based on distance traveled since last location update, and act when this distance reaches tR , for some value of t ($t=1, 2, 3, \dots$).

Thus we believe that quorum based idea for routing in ad hoc networks has the potential to be very efficient, in terms of small hop counts, almost guaranteed delivery, and small communication overhead.

References

- [APL] K.N. Amouris, S. Papavassiliou, M. Li, A position based multi-zone routing protocol for wide area mobile ad-hoc networks, Proc. 49th IEEE Vehicular Technology Conference, 1999, 1365-1369.
- [BCSW] S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), Proc. MOBICOM, 1998, 76-84.
- [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, 48-55.
- [CL] D. Camara and A.F. Loureiro, A novel routing algorithm for ad hoc networks, Proc. HICSS, Hawaii, January 2000, to appear.
- [CN] S. Chen and K. Nahrstedt, Distributed quality-of-service routing in ad hoc networks, IEEE J. Selected Areas in Communications, 17, 8, August 1999, 1488-1505.
- [EGHK] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, Proc. MOBICOM, 1999, Seattle, 263-270.

- [ESC] A. El Abbadi, D. Skeen, F. Cristian, An efficient fault-tolerant algorithm for replicated data management, Proc. 5th ACM SIGACT-SIGMOD Symp. Principles of Database Systems, 215-229, 1985.
- [H] M. Herlihy, Dynamic quorum adjustment for partitioned data, ACM Transactions on Database Systems, 12, 2, 170-194, June 1987.
- [JM] D. Johnson and D. Maltz, Dynamic source routing in ad hoc wireless networks, in Mobile Computing (T. Imielinski and H. Korth, eds), Norwell, MA: Kluwer, 1996.
- [KAS] G. Krishnamurthy, A. Azizoglu, A.K. Somani, Optimal location management algorithms for mobile networks, Proc. 4th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking MOBICOM'98, 223-232, 1998.
- [KMP] G. Karumanchi, S. Muralidharan and R. Prakash, Information dissemination in partitionable mobile ad hoc networks, Proc. IEEE Symp. On Reliable Distributed Systems, Lausanne, Oct. 1999.
- [KSU] E. Kranakis, H. Singh and J. Urrutia, Compass routing on geometric networks, Proc. 11th Canadian Conf. on Computational Geometry, Vancouver, August, 1999.
- [KV] Y.B. Ko and N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, Proc. MOBICOM, 1998, 66-75.
- [LL] C. R. Lin and J.S. Liu, QoS routing in ad hoc wireless networks, IEEE J. Selected Areas in Communications, 17, 8, Aug. 1999, 1426-1438.
- [M] M. Maekawa, A $n^{1/2}$ algorithm for mutual exclusion in decentralized systems, ACM Transactions on Computer Systems, 14-159, May 1985.
- [MC] J.P. Macker and M.S. Corson, Mobile ad hoc networking and the IETF, Mobile Computing and Communications Review, 2, 1, 1998, 9-14.
- [MZ] A.B. McDonald and T.F. Znati, A mobility-based framework for adaptive clustering in wireless ad hoc networks, IEEE J. Selected Areas in Communications, 17, 8, August 1999, 1466-1487.
- [N] NAVSTAR GPS operations, <http://tycho.usno.navy.mil/gpsinfo.html>, and Iowa State University GPS page, <http://www.cnde.iastate.edu/gps.html>.
- [NTCS] S.Y. Ni, Y.C. Tseng, Y.S. Chen, J.P. Sheu, The broadcast storm problem in a mobile ad hoc network, Proc. MOBICOM, Seattle, Aug. 1999, 151-162.
- [PR] C. Perkins and E.M. Royer, Ad hoc on demand distance vector (AODV) routing, internet draft, August 1998.
- [PHS] R. Prakash, Z.J. Haas, M. Singhal, Load balanced location management for mobile systems using dynamic hashing and quorums, TR-UTDCS-05-97, The University of Texas at Dallas, October 1997.
- [PS] R. Prakash and M. Singhal, A dynamic approach to location management in mobile computing systems, Proc. 8th Int. Conf. on Software Eng. & Knowledge Eng. SEKE'96, 488-495, Lake Tahoe, Nevada, June 1996.
- [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 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, to appear.
- [SL2] I. Stojmenovic and Xu Lin, Power-aware routing in ad hoc wireless networks, SITE, University of Ottawa, TR-98-11, December 1998.

- [SSW] M. Seddigh, I. Stojmenovic, J. Wu, Location and internal nodes based routing in wireless networks, SITE, University of Ottawa, TR-99-08, September 1999.
- [S1] I. Stojmenovic, Geocasting and routing in mobile ad hoc networks.
- [TK] H. Takagi and L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Trans. on Communications, 32, 3, 1984, 246-257.
- [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.