Location-Aided Routing (LAR) in Mobile Ad Hoc Networks *

Young-Bae Ko and Nitin H. Vaidya

Department of Computer Science
Texas A&M University
College Station, TX 77843-3112
{youngbae,vaidya}@cs.tamu.edu

Abstract

A mobile ad hoc network consists of wireless hosts that may move often. Movement of hosts results in a change in routes, requiring some mechanism for determining new routes. Several routing protocols have already been proposed for ad hoc networks. This paper suggests an approach to utilize *location information* (for instance, obtained using the global positioning system) to improve performance of routing protocols for ad hoc networks.

By using location information, the proposed *Location-Aided Routing (LAR)* protocols limit the search for a new route to a smaller "request zone" of the ad hoc network. This results in a significant reduction in the number of routing messages. We present two algorithms to determine the *request zone*, and also suggest potential optimizations to our algorithms.

1 Introduction

Mobile ad hoc networks consist of wireless mobile hosts that communicate with each other, in the absence of a fixed infrastructure.[1] Routes between two hosts in a Mobile Ad hoc NETwork (MANET) may consist of hops through other hosts in the network [7]. Host mobility can cause frequent unpredictable topology changes. Therefore, the task of finding and maintaining routes in MANET is nontrivial. Many protocols have been proposed for mobile ad hoc networks, with the goal of achieving efficient routing [6, 9, 11, 12, 14, 16, 17, 18, 21, 23, 24, 28]. These algorithms differ in the approach used for searching a new route and/or modifying a known route, when hosts move.

In this paper, we suggest an approach to decrease overhead of route discovery by utilizing location information for mobile hosts. Such location information may be obtained using the global positioning system (GPS) [10, 22]. We demonstrate how location information may be used by means of two *Location-Aided Routing* (LAR) protocols for route discovery. The LAR protocols use location information (which may be out of date, by the time it is used) to reduce the search space for a desired route. Limiting the search space results in fewer route discovery messages.

* Research reported is supported in part by Texas Advanced Technology Program grants 010115-248 and 009741-052-C.

[1] We will use the terms *host* and *node* interchangeably.

2 Related Work

Design of routing protocols is a crucial problem in mobile ad hoc networks [7, 25], and several routing algorithms have been developed (e.g., [6, 9, 11, 12, 14, 16, 17, 18, 21, 23, 24, 28]). One desirable qualitative property of a routing protocol is that it should adapt to the traffic patterns [8]. Johnson and Maltz [15, 16] point out that conventional routing protocols are insufficient for ad hoc networks, since the amount of routing related traffic may waste a large portion of the wireless bandwidth, especially for protocols that use periodic updates of routing tables. They proposed using Dynamic Source Routing (DSR), which is based on *on-demand* route discovery. A number of protocol optimizations are also proposed to reduce the route discovery overhead. Perkins and Royer [23] present the AODV (Ad hoc On Demand Distance vector routing) protocol that also uses a demand-driven route establishment procedure. More recent TORA (Temporally-Ordered Routing Algorithm) [21] is designed to minimize reaction to topological changes by localizing routing-related messages to a small set of nodes near the change. Hass and Pearlman [12] attempt to combine proactive and reactive approaches in the Zone Routing Protocol (ZRP), by initiating route discovery phase on-demand, but limits the scope of the proactive procedure only to the initiator's local neighborhood. Also, ZRP limits topology update propagation to the neighborhood of the change. There is a recent approach for comparative performance evaluation of several routing protocols proposed in MANET [26].

The existing MANET routing algorithms do not take into account the physical location of a destination node. In this paper, we propose two algorithms to reduce route discovery overhead using location information. Similar ideas have been applied to develop *selective paging* for cellular PCS (Personal Communication Service) networks [4]. In selective paging, the system pages a selected subset of *cells* close to the last reported location of a mobile host. This allows the location tracking cost to be decreased. We propose and evaluate an analogous approach for routing in MANET. Metricom is a packet radio system using location information for the routing purpose [19]. The Metricom network infrastructure consists of fixed base stations whose precise location is determined using a GPS receiver at the time of installation. Metricom uses a geographically based routing scheme to deliver packets between base stations. Thus, a packet is forwarded one hop closer to its final destination by comparing the location of packet's destination with the location of the node currently holding the packet. In a survey of potential applications of GPS, Dommety and Jain [10] briefly suggest use of location information in ad hoc networks, though they do not elaborate on how the information may be used. Other researchers have also suggested that location information should be used to improve (qualitatively or quantitatively) performance of a mobile computing system [27, 29]. A routing and addressing method to integrate the concept of physical location (geographic coordinates), into the current design of the Internet, has been investigated in [13, 20].

3 Location-Aided Routing(LAR) Protocols

3.1 Route Discovery Using Flooding

In this paper, we explore the possibility of using location information to improve performance of routing protocols for MANET. As illustration, we show how a route discovery protocol based on *flooding* can be improved. The route discovery algorithm using flooding is described next (this algorithm is similar to Dynamic Source Routing [15, 16]). When a node S needs to find a route to node D, node S broadcasts a *route request* message to all its neighbors[2] – hereafter, node S will be referred to as the *sender* and node D as the *destination*. A node, say X, on receiving a route request message, compares the desired destination with its own identifier. If there is a match, it means that the request is for a route to itself (i.e., node X). Otherwise, node X broadcasts the request to its neighbors – to avoid redundant transmissions of route requests, a node X only broadcasts a particular route request once (repeated reception of a route request is detected using sequence numbers). Figure 1 illustrates this algorithm. In this figure, node S needs to determine a route to node D. Therefore, node S broadcasts a route request to its neighbors. When nodes B and C receive the route request, they forward it to all their neighbors. When node X receives the route request from B, it forwards the request to its neighbors. However, when node X receives the same route request from C, node X simply discards the route request.

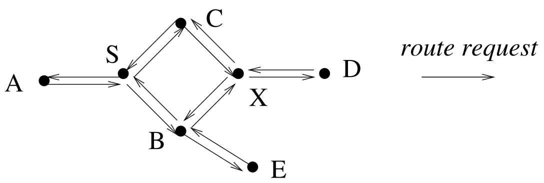


Figure 1: Illustration of flooding

As the route request is propagated to various nodes, the path followed by the request is included in the route request packet. Using the above flooding algorithm, provided that the intended destination is reachable from the sender, the destination should eventually receive a route request message. On receiving the route request, the destination responds by sending a *route reply* message to the sender – the route reply message follows a path that is obtained by reversing the path followed by the route request received by D (the route request message includes the path traversed by the request).

It is possible that the destination will not receive a route request message (for instance, when it is unreachable from the sender, or route requests are lost due to transmission errors). In such cases, the sender needs to be able to re-initiate route discovery. Therefore, when a sender initiates route discovery, it sets a timeout. If during the timeout interval, a route reply is not received, then a *new* route discovery is initiated (the route request messages for this route discovery will use a different sequence number than the previous route discovery – recall that sequence numbers are useful to detect multiple receptions of the same route request). Timeout may occur if the destination does not receive a route request, or if the route reply message from the destination is lost.

[2] Two nodes are said to be neighbors if they can communicate with each other over a wireless link.

Route discovery is initiated either when the sender S detects that a previously determined route to node D is broken, or if S does not know a route to the destination. In our implementation, we assume that node S can know that the route is broken only if it attempts to use the route. When node S sends a data packet along a particular route, a node along that path returns a *route error* message, if the next hop on the route is broken. When node S receives the route error message, it initiates route discovery for destination D.

When using the above algorithm, observe that the route request would reach every node that is reachable from node S (potentially, all nodes in the ad hoc network). Using location information, we attempt to reduce the number of nodes to whom route request is propagated.

Dynamic source routing (DSR) [15, 16] and ad hoc on-demand distance vector routing (AODV) [23] protocols proposed previously are both based on variations of flooding. DSR and AODV also use some optimizations - several of these optimizations as well as other optimizations suggested in this paper can be used in conjunction with the proposed algorithms. However, for simplicity, we limit our discussion to the basic flooding algorithm, and location-aided route discovery based on "limited" flooding.

3.2 Preliminaries

Location Information

The proposed approach is termed Location-Aided Routing (LAR), as it makes use of location information to reduce routing overhead. Location information used in the LAR protocol may be provided by the Global Positioning System (GPS) [2, 3, 10, 22]. With the availability of GPS, it is possible for a mobile host to know its physical location[3]. In reality, position information provided by GPS includes some amount of *error*, which is the difference between GPS-calculated coordinates and the real coordinates. For instance, NAVSTAR Global Positioning System has positional accuracy of about 50-100 meters and Differential GPS offers accuracies of a few meters [2, 3]. In our initial discussion, we assume that each host knows its current location *precisely* (i.e., no error). However, the ideas suggested here can also be applied when the location is known only approximately – the Performance Evaluation section considers this possibility.

In this paper, we assume that the mobile nodes are moving in a two-dimensional plane.

Expected Zone and Request Zone

Expected Zone: Consider a node S that needs to find a route to node D. Assume that node S knows that node D was at location L at time t_0, and that the current time is t_1. Then, the *"expected zone"* of node D, from the viewpoint of node S at time t_1, is the region that node S expects to contain node D at time t_1. Node S can determine the expected zone based on the knowledge that node D was at location L at time t_0. For instance, if node S knows that node D travels with average speed v, then S may assume that the expected zone is the circular region of radius $v(t_1 - t_0)$, centered at location L (see Figure 2(a)). If actual speed happens to be larger than the average, then the destination may actually be outside the expected zone at time t_1. Thus, expected zone is only an estimate made by node S to determine a region that potentially contains D at time t_1.

If node S does not know a previous location of node D, then node S cannot reasonably determine the expected zone – in this case, the entire region that may potentially be occupied by the ad

[3] Current GPS provides accurate three-dimensional position (latitude, longitude, and altitude), velocity, and precise time traceable to Coordinated Universal Time(UTC) [1]

hoc network is assumed to be the expected zone. In this case, our algorithm reduces to the basic flooding algorithm. In general, having more information regarding mobility of a destination node, can result in a smaller expected zone. For instance, if S knows that destination D is moving north, then the circular expected zone in Figure 2(a) can be reduced to a semi-circle, as in Figure 2(b).

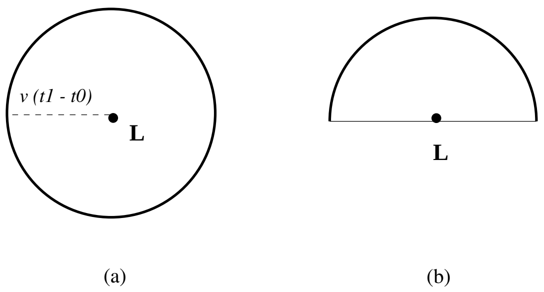


Figure 2: Examples of *expected zone*

Request Zone: Again, consider node S that needs to determine a route to node D. The proposed LAR algorithms use flooding with one modification. Node S defines (implicitly or explicitly) a *request zone* for the route request. A node forwards a route request *only if* it belongs to the request zone (unlike the flooding algorithm in Section 3.1). To increase the probability that the route request will reach node D, the request zone should include the *expected zone* (described above). Additionally, the request zone may also include other regions around the request zone. There are two reasons for this:

- When the expected zone does not include host S, a path from host S to host D must include hosts outside the expected zone. Therefore, additional region must be included in the request zone, so that S and D both belong to the request zone (for instance, as shown in Figure 3(a)).
- The request zone in Figure 3(a) includes the expected zone from Figure 2(a). Is this an adequate request zone? In the example in Figure 3(b), all paths from S to D include hosts that are outside the request zone. Thus, there is no guarantee that a path can be found consisting only of the hosts in a chosen request zone. Therefore, if a route is not discovered within a suitable timeout period, our protocol allows S to initiate a new route discovery with an expanded request zone – in our simulations, the expanded zone includes the entire network space. In this event, however, the latency in determining the route to D will be longer (as more than one round of route request propagation will be needed).

  Note that the probability of finding a path (in the first attempt) can be increased by increasing the size of the initial request zone (for instance, see Figure 3(c)). However, route discovery overhead also increases with the size of the request zone. Thus, there exists a trade-off between latency of route determination and the message overhead.

3.3 Determining Membership of Request Zones

As noted above, our LAR algorithms are essentially identical to flooding, with the modification that a node that is not in the request zone does not forward a route request to its neighbors.[4] Thus, implementing LAR algorithm requires that a node be able to determine if it is in the request zone for a particular route request – the two LAR algorithms presented here differ in the manner in which this determination is made.

[4] Recall that, in the flooding algorithm, a node forwards a route request if it has not received the request before and it is not the intended destination.

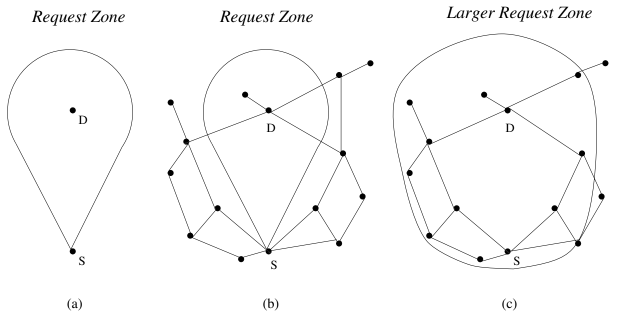


Figure 3: Request zone: An edge between two nodes means that they are neighbors

LAR Scheme 1

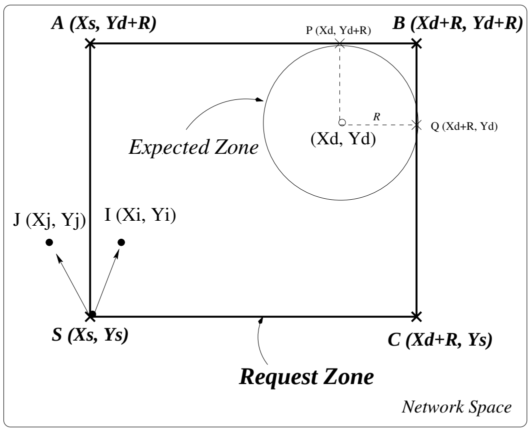
Our first scheme uses a request zone that is rectangular in shape (refer to Figure 4). Assume that node S knows that node D was at location (X_d, Y_d) at time t_0. At time t_1, node S initiates a new route discovery for destination D. We assume that node S also knows the average speed v with which D can move. Using this, node S defines the expected zone at time t_1 to be the circle of radius $R = v(t_1 - t_0)$ centered at location (X_d, Y_d).

In our first LAR algorithm, we define the request zone to be the smallest rectangle that includes current location of S and the expected zone (the circular region defined above), such that the sides of the rectangle are parallel to the X and Y axes. In Figure 4(a), the request zone is the rectangle whose corners are S, A, B and C, whereas in Figure 4(b), the rectangle has corners at points A, B, C and G – note that, in this figure, current location of node S is denoted as (X_s, Y_s).

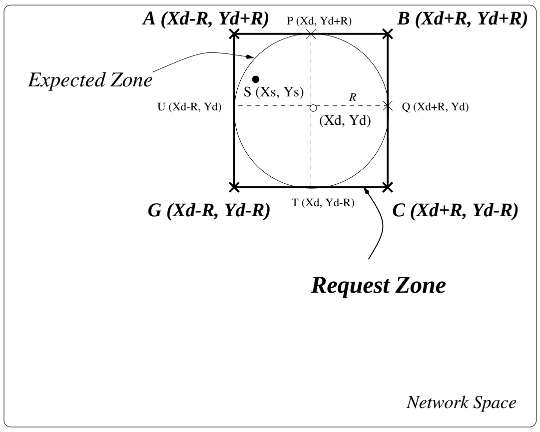
The source node S can thus determine the four corners of the expected zone. S includes their coordinates with the route request message transmitted when initiating route discovery. When a node receives a route request, it discards the request if the node is not within the rectangle specified by the four corners included in the route request. For instance, in Figure 4(a), if node I receives the route request from another node, node I forwards the request to its neighbors, because I determines that it is within the rectangular request zone. However, when node J receives the route request, node J discards the request, as node J is not within the request zone (see Figure 4(a)).

When node D receives the route request message, it replies by sending a route reply message (as in the flooding algorithm). However, in case of LAR, node D includes its current location and current time in the route reply message. When node S receives this route reply message (ending its route discovery), it records the location of node D. Node S can use this information to determine the request zone for a future route discovery. (It is also possible for D to include its current speed, or average speed over a recent time interval, with the route reply message. This information could be used in a future route discovery. In our simulations, we assume that all nodes know each other's average speed.)

Size of the request zone: Note that the size of the rectangular request zone above is proportional to (i) average speed of movement v, and (ii) time elapsed since the last known location of the destination was recorded. In our implementation, the sender comes to know location of the destination only at the end of a route discovery (as noted in the previous paragraph). At low speeds, route discoveries occur after long intervals, because routes break less often (thus, $t_1 - t_0$ is large). So, although factor (i) above is small, factor (ii) becomes large at low speeds, potentially resulting in a larger request zone. At high speeds as well, for similar reasons, a large request zone may be observed. So, in general, a smaller request zone may occur at speeds that are neither too small, nor too large. For low speeds, it is possible to reduce the size of the request zone by piggybacking the location information on other packets, in addition to route replies (this optimization is not evaluated here).



(a) Source node outside the Expected Zone



(b) Source node within the Expected Zone

Figure 4: LAR scheme 1

LAR Scheme 2

In LAR scheme 1, source S explicitly specifies the request zone in its route request message. In scheme 2, node S includes two pieces of information with its route request:

- Assume that node S knows the location (X_d, Y_d) of node D at some time t_0 – the time at which route discovery is initiated by node S is t_1, where $t_1 \geq t_0$. Node S calculates its distance from location (X_d, Y_d), denoted as $DIST_s$, and includes this distance with the route request message.
- The coordinates (X_d, Y_d) are also included with the route request.

When a node I receives the route request from sender node S, node I calculates its distance from location (X_d, Y_d), denoted as $DIST_i$, and:

- For some parameter δ, if $DIST_s + \delta \geq DIST_i$, then node I forwards the request to its neighbors. When node I forwards the route request, it now includes $DIST_i$ and (X_d, Y_d) in the route request (i.e., it replaces the $DIST_s$ value received in the route request by $DIST_i$, before forwarding the route request).
- Else $DIST_s + \delta < DIST_i$. In this case, node I discards the route request.

When some node J receives the route request (originated by node S) from node I, it applies a criteria similar to above: If node J has received this request previously, it discards the request. Otherwise, node J calculates its distance from (X_d, Y_d), denoted as $DIST_j$. Now,

- The route request received from I includes $DIST_i$. If $DIST_i$ $+\delta \geq DIST_j$, then node J forwards the request to its neighbors (unless node J is the destination for the route request). Before forwarding the request, J replaces the $DIST_i$ value in the route request by $DIST_j$.
- Else $DIST_i + \delta < DIST_j$. In this case, node J discards the request.

Thus, a node J forwards a route request forwarded by I (originated by node S), if J is "at most δ farther" from (X_d, Y_d) than node I. For the purpose of performance evaluation, we use $\delta = 0$ in the next section. Non-zero δ may be used to trade-off the probability of finding a route on the first attempt with the cost of finding the route. Non-zero δ may also be appropriate when *location error* is non-zero, or when the hosts are likely to move significant distances during the time required to perform route discovery.

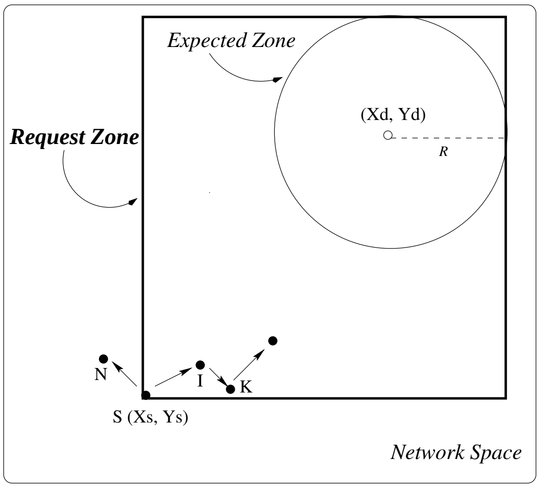
Figure 5 illustrates the difference between the two LAR schemes. Consider Figure 5(a) for LAR scheme 1: When nodes I and K receive the route request for node D (originated by node S), they forward the route request, as both I and K are within the rectangular request zone. On the other hand, when node N receives the route request, it discards the request, as N is outside the rectangular request zone. Now consider Figure 5(b) for LAR scheme 2 (assume $\delta = 0$): When nodes N and I receive the route request from node S, both forward the route request to their neighbors, because N and I are both closer to (X_d, Y_d) than node S. When node K receives the route request from node I, node K discards the route request, as K is farther from (X_d, Y_d) than node I. Observe that nodes N and K take different actions when using the two LAR schemes.

Error in Location Estimate

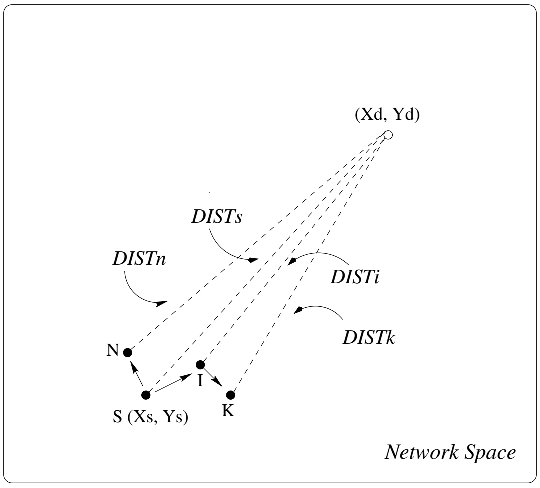
In the above, we assume that each node knows *its own* location accurately. However, in reality there may be some error in the estimated location. Let e denote the maximum error in the coordinates estimated by a node. Thus, if a node N believes that it is at location (X_n, Y_n), then the actual location of node N may be anywhere in the circle of radius e centered at (X_n, Y_n).

In the next section, we will refer to e as *location error*. In the above LAR schemes, we assume that node S obtained the location (X_d, Y_d) of node D at time t_0, from node D (perhaps in the route reply message during the previous route discovery). Thus, node S does not know the actual location of node D at time t_0 – the actual location is somewhere in the circle of radius e centered at (X_d, Y_d).

To take the location error e into account, we modify LAR scheme 1 so that the expected zone is now a circle of radius $e + v(t_1 - t_0)$. The request zone may now be bigger, as it must include the larger request zone. Apart from this, no other change is needed in the algorithm. As the request zone size increases with e, the routing overhead may be larger for large e. We make no modifications to LAR scheme 2, even when location error e is non-zero. However, the performance of scheme 2 may degrade with large location error, because with larger e, there is a higher chance that the request zone used by the scheme will not include a path to the destination (resulting in a timeout and another route discovery). We briefly evaluate the case of $e > 0$ at the end of the next section.



(a) LAR scheme 1



(b) LAR scheme 2

Figure 5: Comparison of the two LAR schemes

4 Performance Evaluation

To evaluate our schemes, we performed simulations using modified version of a network simulator, MaRS (Maryland Routing Simulator) [5]. MaRS is a discrete-event simulator built to provide a flexible platform for the evaluation and comparison of network routing algorithms. Three routing protocols were simulated – flooding, LAR scheme 1 and LAR scheme 2. We studied several cases by varying the number of nodes, transmission range of each node, and moving speed.

4.1 Simulation Model

Number of nodes in the network was chosen to be 15, 30 and 50 for different simulation runs. The nodes in the ad hoc network are confined to a 1000 unit x 1000 unit square region. Initial locations (X and Y coordinates) of the nodes are obtained using a uniform distribution.

We assume that each node moves continuously, without pausing at any location. Each node moves with an *average* speed v. The actual speed is uniformly distributed in the range $v - \alpha$ and $v + \alpha$ units/second, where, we use $\alpha = 1.5$ when $v < 10$ and $\alpha = 2.5$ when $v \geq 10$. We consider average speeds (v) in the range 1.5 to 32.5 units/sec.

Each node makes several "moves" during the simulation. A node does *not* pause between moves. During a given move, a node travels distance d, where d is exponentially distributed with mean 20 units. The direction of movement for a given move is chosen randomly. For each such move, for a given average speed v, the actual speed of movement is chosen uniformly distributed between $[v - \alpha, v + \alpha]$. If during a move (over chosen distance d), a node "hits" a wall of the 1000x1000 region, the node bounces and continues to move after reflection, for the remaining portion of distance d.

Two mobile hosts are considered disconnected if they are outside each other's transmission range. All nodes have the same transmission range. For the simulations, transmission range values of 200, 300, 400, and 500 units were used. All wireless links have the same bandwidth, 100 Kbytes per second.

In our simulation, simulation time is inversely proportional to the average speed. For instance, simulations for average speed 1.5 units/sec run 4000 seconds of execution, whereas about 1333 seconds for average speed 4.5 units/sec. As the average speed is increased, for a given simulation time, the number of moves simulated increases. Thus, although the simulations at different speeds are for the same mobility model, as speed is increased, a particular configuration (for instance, partition) that may not have occurred at a lower speed can occur at the higher speed. On the other hand, a configuration that did occur at a lower speed lasts a shorter time when the speed is higher.

For the simulation, a sender and a destination are chosen randomly. Any data packets that cannot be delivered to the destination due to a broken route are simply dropped. The source generates 10 data packets per second (on average), with the time between two packets being exponentially distributed. The data rate was chosen low to speed up the simulation. However, this has the impact of sending small number of packets between two route discoveries (as compared to when the source continuously sends packets). This, in turn, results in higher *number of routing packets per data packet* (defined below).

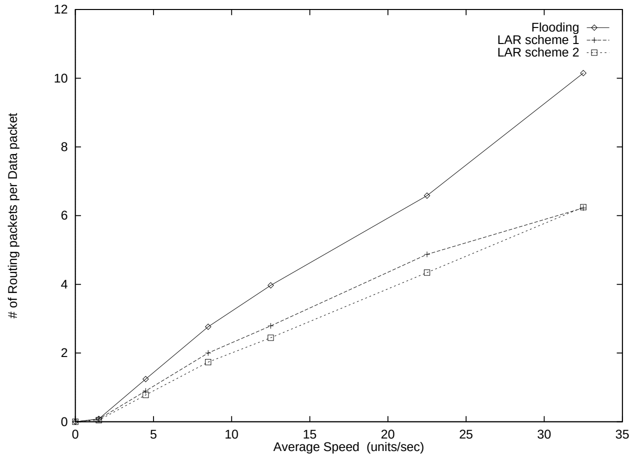
When using the LAR schemes for route discovery, the sender first uses our algorithm to determine a route – if a route reply is not received within a timeout interval, the sender uses the flooding algorithm to find the route. The timeout interval is 2 seconds on average.

In our simulations, we do not model the delays that may be introduced when multiple nodes attempt to transmit simultaneously. Transmission errors are also not considered.

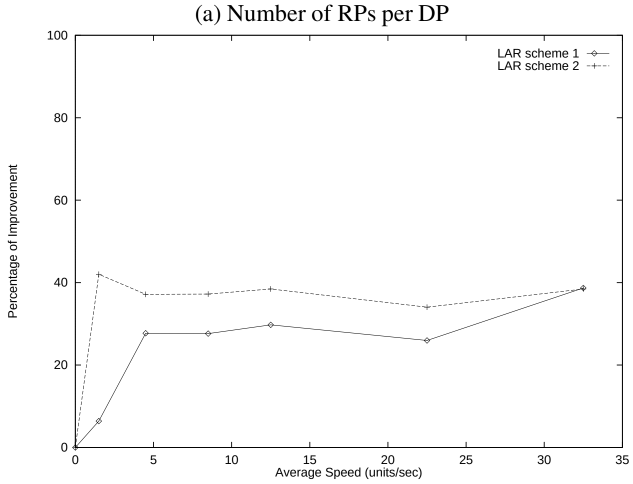
4.2 Simulation Results

Initially, we assume that a node knows its current location accurately, without any error. At the end of this section, we briefly consider the impact of *location error* on performance of our algorithms.

In the following, the term "data packets" (or DP) is used to refer to the data packets *received* by the destination – the number of data packets received by the destination is different from number of data packets sent by the sender, because some data packets are lost when a route is broken. In the following, the term "routing packets" (or RP) is used to refer to the routing related packets (i.e., route request, route reply and route error) *received* by various nodes – number of such packets received is different from number of packets *sent*, because a single broadcast of a *route request* packet by some node is received by *all* its neighbors (also, some of these packets could be lost due to broken routes).



(a) Number of RPs per DP



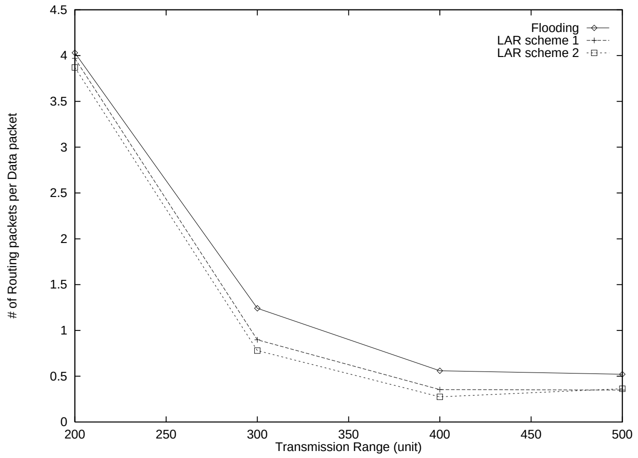
(b) Percentage Improvement

Figure 6: For 30 nodes, and transmission range 300 units: (a) Number of RPs per DP versus Average Speed, (b) Percentage of Improvement versus Average Speed

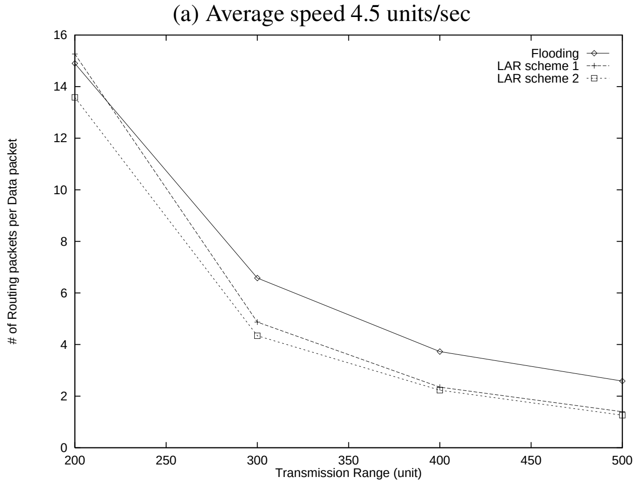
We compare the results from LAR scheme 1 and LAR scheme 2 with those from the flooding algorithm. In each run, one input parameter (e.g. average speed, number of nodes, or transmission range) was varied while the other parameters were kept constant. Our simulation results are an average over 30 runs, each with a different mobility pattern (different mobility patterns were obtained by choosing different seeds for a random number generator).

The number of routing packets (RP) per data packet (DP) is depicted in Figure 6(a) as a function of average speed. This is calculated as the ratio of the number of routing packets, and the number of data packets received by the destination. Figure 6(b) shows the same data, but plotted as the percentage improvement using LAR, relative to flooding algorithm.

Figures 6(a) and (b) show that the number of routing packets per data packet is consistently lower for both LAR schemes as compared to flooding. As the speed of mobile hosts is increased, the number of routing packets begins to increase for all routing protocols. With higher speed, the frequency of route breaking increases, so routing overhead to discover new routes also increases. However, LAR schemes 1 and 2 provide a lower rate of increase than flooding. This is because, with LAR, number of route requests is significantly reduced by limiting route discovery to a smaller request zone.



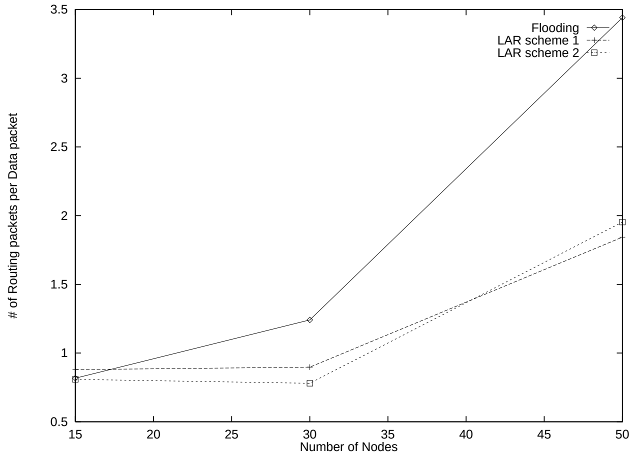
(a) Average speed 4.5 units/sec



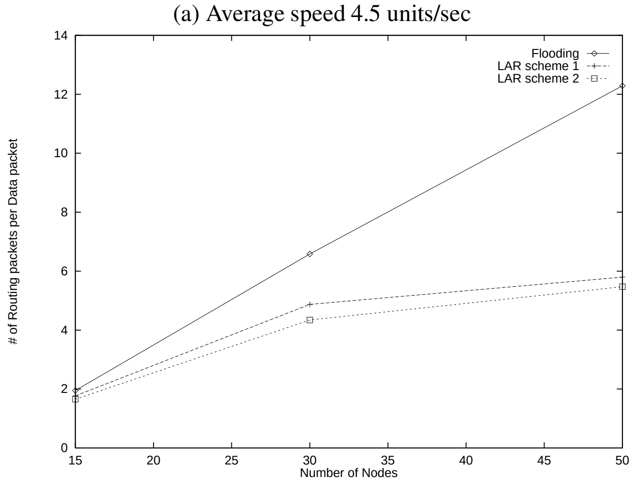
(b) Average speed 22.5 units/sec

Figure 7: Number of RPs per DP versus Transmission Range (with 30 nodes)

Figure 7 shows the effect of varying the transmission range. Typically, the routing overhead decreases with increasing transmission range. With a larger transmission range, the frequency of route discovery should be smaller, as wireless links will break less frequently. This factor contributes to a decrease in routing overhead for all three schemes. Our schemes continue to perform better than flooding. However, with a smaller transmission range (200 units in Figure 7), performance of our schemes is not much better than flooding. In Figure 7(b), LAR scheme 1 performs even worse than flooding. When a node forwards a route request, it broadcasts the request to all its neighbors. With a smaller transmission range, number of neighbors for each node decreases. This factor decreases the probability of a route discovery within the timeout interval, using the initial request zone. Recall that, in this case, our schemes allow the sender to initiate a new route discovery using the flooding algorithm. We believe that this is the reason why LAR schemes do not perform too well when transmission range is small. The different request zones used in the two LAR schemes result in different routing overhead for the two schemes.



(a) Average speed 4.5 units/sec



(b) Average speed 22.5 units/sec

Figure 8: Number of RPs per DP versus Number of Nodes (Transmission range 300 uni ts)

The effect of varying the number of nodes is shown in Figure 8. Amount of routing overhead for the flooding algorithm increases much more rapidly than LAR schemes, when number of nodes is increased. As noted earlier in the discussion of Figure 7(b), smaller probability of success of route discovery using initial request zone contributes to a larger routing overhead. Similar to the case of small transmission range, the LAR schemes do not perform much better than flooding with a small number of nodes (15 nodes in Figure 8).

Figure 9 shows the number of routing packets per route discovery. As can be seen in the graph, LAR scheme 2 has the smallest number of routing packets per route discovery even though LAR scheme 1 also has smaller values than the flooding algorithm.

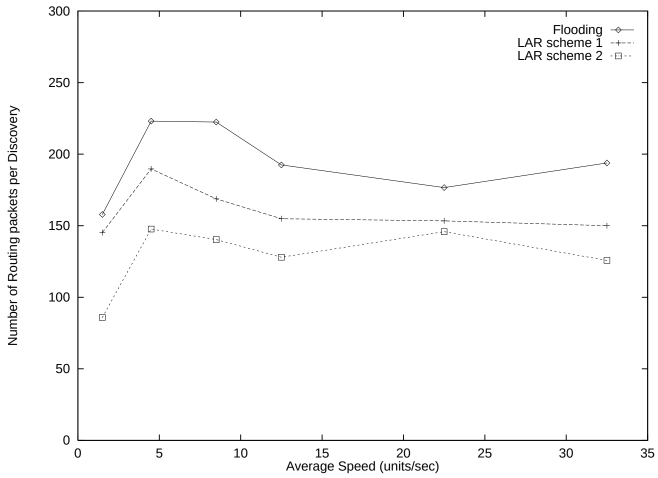


Figure 9: For 30 nodes, and transmission range 300 units: Number of RPs per Discovery versus Speed

Impact of Location Error

As noted at the end of the previous section, the location of a node estimated using GPS may include some error, say e, which causes each estimated coordinate (X and Y) to be in error by at most e units. In the above simulations, we assumed $e = 0$. Figure 10(a) shows how the location error affects routing overhead (i.e., number of routing packets per data packet).

In Figure 10, our schemes continue to perform better than flooding for the chosen parameters (i.e., average speed, number of nodes, transmission range). Typically, routing overhead for LAR schemes increases with increasing location error. However, although it is hard to see in Figure 10(a), the curve for LAR scheme 1 is not monotonically increasing. Note that the number of routing packets(RP) per data packet(DP) at $e = 75$ is smaller than that at $e = 50$.

With a larger location error, the size of request zone increases (See Figure 11(a) and (b)). This factor usually contributes to an increase in routing overhead. However, routing overhead, when location error is increased, may decrease. This is because, when the size of request zone is larger, the probability that the discovery will succeed on the first attempt is larger, which can result in smaller number of RPs per DP.

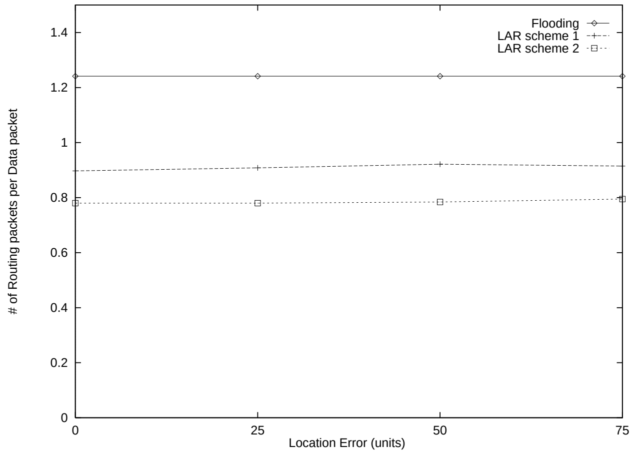
Figure 10(b) plots the relative increase in the routing overhead of LAR schemes 1 and 2, when location error is non-zero, as compared to when the error is 0. Observe that the increase in routing overhead is small.

LAR schemes use location information to attempt to improve routing performance. Intuition suggests that, when location error is very large, such schemes would not be very effective. Further work is needed to determine at what location error levels proposed LAR schemes become ineffective.

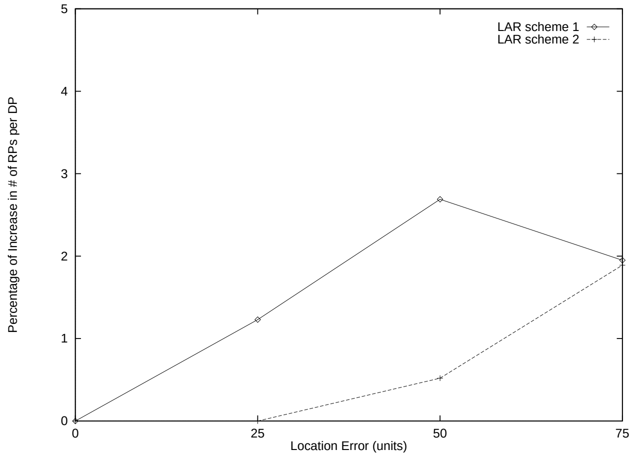
5 Variations and Optimizations

Alternative Definitions of Request Zone

In this paper, we consider two ways of defining a request zone. Several other alternatives may be conceived. For instance, in the rectangular request zone of LAR scheme 1, sender node S may be on the border of the zone (refer Figure 4(a)). Instead, one may define a larger rectangle as the request zone. Also, in LAR scheme 1, the sides of the rectangle are always parallel to the X and Y axes. It is possible to remove this restriction when defining the rectangular region.



(a) Number of RPs per DP



(b) Percentage increase in RPs per DP

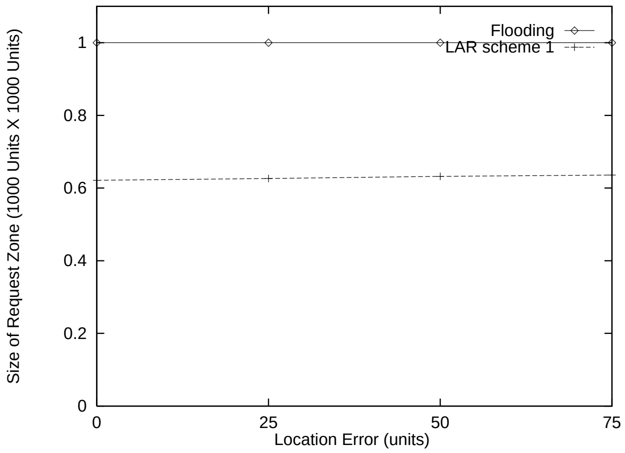
Figure 10: For 30 nodes, average speed 4.5 units/sec, and transmission range 300 u nits: (a) Number of routing packets per data packet versus location error, (b) Percentage increase in number of routing packets per data packet versus location error

In our simulation for the two LAR schemes, the request zone is expanded to the entire network space when a sender using our algorithm fails to find the route to a destination within a timeout interval. This simple strategy of expanding the request zone causes performance degradation of LAR schemes with a smaller transmission range and number of nodes. This scheme may be improved by increasing the request zone gradually.

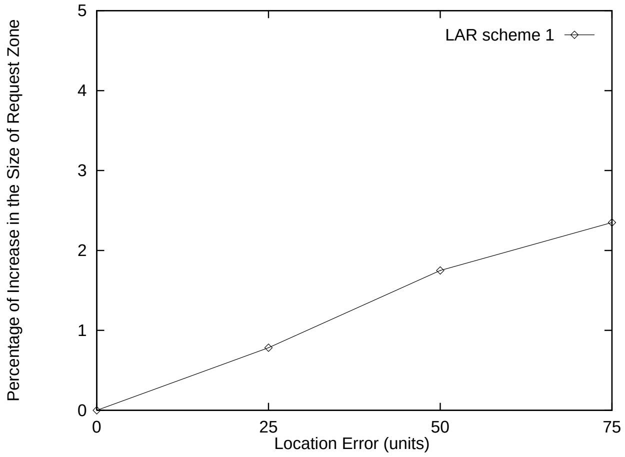
Definition of a request zone is also dependent on how much information regarding the mobile hosts is available. We assume that only average speed of the nodes is known. It is interesting to consider situations wherein additional information may be available (for instance, direction of movement). The impact of alternative definitions of request zone is a topic for further work.

Adaptation of Request Zone

Accuracy of a request zone (i.e., probability of finding a route to the destination) can be improved by adapting the request zone, initially determined by the source node S, with up-to-date location information for host D, which can be acquired at some intermediate nodes. Let us consider the case that node S starts search of a destination node D within a request zone Z at time t_1, which is based on location information about D learned by S at time t_0. Let us assume that the *route request* includes the timestamp t_0, because the location of node D at time t_0 is used to determine the request zone. Also, location of node S and the time t_1 when the request is originated are also included. Now suppose that some intermediate node I within Z receives the route request at time t_2, where $t_1 < t_2$. More recent location information for D may potentially be known by node I (as compared to node S), and the expected zone based on that information may be different from previous request zone Z. Therefore, request zone initially determined at a source node may be adapted at node I. For instance, when using LAR scheme 2, node I may calculate distance from the more recent location of destination D that it knows, and use this distance in the decision rule (to decide whether to discard a route request) of scheme 2.



(a) Size of Request Zone



(b) Percentage increase in the Size of Request Zone

Figure 11: For 30 nodes, average speed 4.5 units/sec, and transmission range 300 nits: (a) Size of request zone versus location error, (b) Percentage increase in the size of request zone

Propagation of Location and Speed Information

Initially, in ad hoc network environments, a node may not know the physical location (either current or old) of other hosts. However, as time progress, each node can get location information for many hosts either as a result of its own route discovery or as a result of message forwarding for another node's route discovery. For instance, if node S includes its current location in the route request message, and if node D includes its current location in the route reply message, then each node receiving these messages can know the locations of nodes S and D, respectively. In general, location information may be propagated by piggybacking it on any packet. Similarly, a node may propagate to other nodes its average speed (over a recent interval of time) information. In our simulations, we assume that average speed is constant and known to all nodes. In practice, the average speed could be time-variant.

Local Search

In our protocol, any intermediate node I detecting routing failure (due to a broken link) informs the source node S by sending a *route error* packet (see Figure 12(a)). Then, S initiates a new route discovery (using a request zone), to find a path to the destination D. As we have already seen, if we use location information, routing messages can be reduced by limiting propagation of route request packets to the request zone determined (implicitly or explicitly) by node S, as shown in Figure 12(b). Figure 12(c) shows how this scheme may be improved to reduce the size of request zone as well as latency of route re-determination for node D. This can be done by allowing any intermediate node I detecting route error to initiate a route discovery using a request zone based on its own location information for node D. Such a *local search* may result in a smaller request zone (as shown in Figure 12(c)) because node I may be closer to D than S. Smaller request zone could reduce routing overhead. The time to find the new path to D may also be reduced, as a smaller request zone is searched.

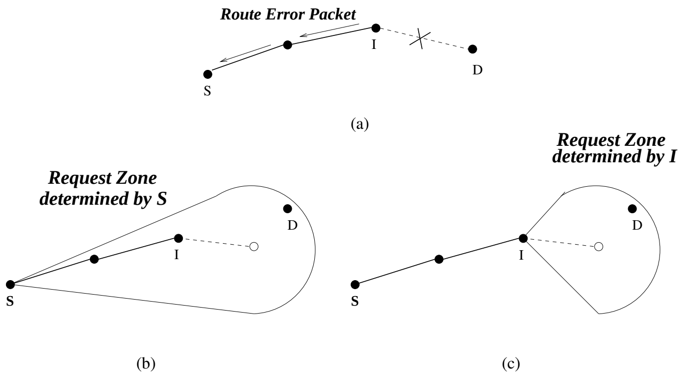


Figure 12: Local Search to Re-establish a Broken Route

6 Conclusion

This paper describes how location information may be used to reduce the routing overhead in ad hoc networks. We present two location-aided routing (LAR) protocols. These protocols limit the search for a route to the so-called request zone, determined based on the expected location of the destination node at the time of route discovery. Simulation results indicate that using location information results in significantly lower routing overhead, as compared to an algorithm that does not use location information.

We also suggest some optimizations that can improve the performance of proposed LAR schemes. Further work is required to evaluate efficacy of these optimizations, and also to develop other ways of using location information in ad hoc networks.

Acknowledgements

We thank the referees for their helpful comments.

References

[1] "GPS and precision timing applications." Web site at http://wwwtmo.external.hp.com/tmo/pia/infinium/PIATop/Notes/English/5965-2791E.html.

[2] "Iowa State University GPS page." Web site at http://www.cnde.iastate.edu/gps.html.

[3] "NAVSTAR GPS operations." Web site at http://tycho.usno.navy.mil/gpsinfo.html.

[4] I. F. Akyildiz, S. M. Joseph, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 94–104, 1996.

[5] C. Alaettinoglu, K. Dussa-Zieger, I. Matta, and A. U. Shankar, "MaRS user's manual - version 1.0," Tech. Rep. TR 91-80, The University of Maryland, June 1991.

[6] M. S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *ACM J. Wireless Networks*, vol. 1, no. 1, pp. 61–81, 1995.

[7] S. Corson, S. Batsell, and J. Macker, "Architectural considerations for mobile mesh networking (Internet draft RFC, version 2)," May 1996.

[8] S. Corson and J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations (Internet-Draft)," Mar. 1998.

[9] B. Das, E. Sivakumar, and V. Bhargavan, "Routing in ad-hoc networks using a spine," in *IEEE International Conference on Computer Communications and Networks '97*, 1997.

[10] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," Tech. Rep. TR-24, CS Dept., The Ohio State University, April 1996.

[11] R. Dube, C. D. Rais, K. Wang, and S. K. Tripathi, "Signal stability based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Personal Communication*, Feb. 1997.

[12] Z. J. Haas and M. R. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks (Internet-Draft)," Aug. 1998.

[13] T. Imielinski and J. C. Navas, "GPS-based addressing and routing," Tech. Rep. LCSR-TR-262, CS Dept., Rutgers University, March (updated August) 1996.

[14] M. Jiang, J. Li, and Y.-C. Tay, "Cluster based routing protocol (CBRP) functional specification (Internet-Draft)," Aug. 1998.

[15] D. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (T. Imielinski and H. Korth, eds.), Kluwere Academic Publishers, 1996.

[16] D. Johnson, D. A. Maltz, and J. Broch, "The dynamic source routing protocol for mobile ad hoc networks (Internet-Draft)," Mar. 1998.

[17] Y.-B. Ko and N. H. Vaidya, "Location-aided routing in mobile ad hoc networks," Tech. Rep. 98-012, CS Dept., Texas A&M University, June 1998.

[18] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan, "A cluster-based approach for routing in adhoc networks," in *USENIX Symposium on Location Independent and Mobile Computing*, Apr. 1995.

[19] Metricom Web Page, "`http://www.metricom.com`."

[20] J. C. Navas and T. Imielinski, "Geocast - geographic addressing and routing," in *Proc. of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '97)*, 1997.

[21] V. D. Park and M. S. Corson, "Temporally-ordered routing algorithm (TORA) version 1 functional specification (Internet-Draft)," Aug. 1998.

[22] B. Parkinson and S. Gilbert, "NAVSTAR: global positioning system - ten years later," in *Proceeding of IEEE*, pp. 1177–1186, 1983.

[23] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector (AODV) routing (Internet-Draft)," Aug. 1998.

[24] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM Symposium on Communication, Architectures and Protocols*, 1994.

[25] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communication networks," *Mobile Networks and Applications*, pp. 89–104, 1996.

[26] R. D. Samir, C. Robert, Y. Jiangtao, and S. Rimli, "Comparative performance evaluation of routing protocols for mobile, ad hoc networks," in *Proc. of IEEE the Seventh International Conference on Computer Communications and Networks (IC3N '98) – to appear*, Oct. 1998.

[27] M. Spreitzer and M. Theimer, "Providing location information in a ubiquitous computing environment," in *Proc. Symposium on Operating System Principles*, Dec. 1993.

[28] C.-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," *Wireless Personal Communication*, Jan. 1997.

[29] M. Weiser, "The computer for the 21st century," *Scientific American*, pp. 94–104, Sept. 1991.