A Location-based Directional Route Discovery (LDRD) Protocol in Mobile Ad-hoc Networks

Stephen S. Yau, Wei Gao, and Dazhi Huang
Dept. of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809, USA
{yau, w.gao, dazhi.huang}@asu.edu

Abstract – **Mobile Ad-hoc Networks (MANETs) are important in pervasive computing systems, in which users discover and utilize various services to achieve their goals. Integrating service discovery with efficient route discovery protocol in MANETs can greatly improve the efficiency of pervasive computing systems, and hence an efficient route discovery protocol in MANETs is desirable. In this paper, a Location-based Directional Route Discovery (LDRD) protocol with a node location service based on local coordinates in MANETs is presented. In our LDRD protocol, route discovery is only performed in allowed areas using directional route requests. This LDRD protocol is shown to greatly reduce the overhead of route discovery, and improve the efficiency, adaptability and applicability of MANETs in various types of network scenarios.**

Keywords – **Mobile ad-hoc network, route discovery, location-based protocol, and directional route request**

I. INTRODUCTION

Pervasive computing systems enable users to interact with information and computing resources, usually considered as services. Hence, service discovery capability is needed for users to discover and utilize various needed services to achieve their application goals.

In most cases, after a user discovers a suitable service, it is necessary to start a route discovery process to reach the service provider. Furthermore, the messages for the service discovery and route discovery will likely travel through the same nodes. The large amount of network traffic due to interactions between users and service providers can be greatly reduced by integrating route discovery with service discovery, and hence the system efficiency can be greatly improved [1, 2].

In order to achieve such integration, a highly efficient route discovery protocol is needed. Due to the dynamic network topology and resource-poor mobile nodes in MANETs, on-demand protocols [3, 4], instead of table-driven protocols [5], are widely used. However, existing on-demand protocols cause large "route redundancy" because the protocols flood route requests in the network and generate a large number of "reverse" route entries to the originator of route requests. These route entries are often useless due to the asymmetric distribution of network traffic. Such route redundancy seriously reduces the efficiency of route discovery.

Such route redundancy can be greatly reduced by restricting the influence range of route discovery requests in the entire network. In this paper, we will present a Location-based Directional Route Discovery (LDRD) protocol, which uses directional route requests to perform route discovery in allowed areas decided by the node mobility model and the location information of the route destination. We have performed simulations of LDRD to show the great improvement of the efficiency of route discovery. In addition, we will also present a Node Location Service based on Local Coordinates (NLS-LC), which not only provides the required location information for LDRD, but also makes LDRD insensitive to the accuracy of location information. This insensitivity makes LDRD highly adaptable and applicable in different types of network scenarios.

This work was supported by the National Science Foundation under grant number ITR-CYBERTRUST 0430565

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

II. CURRENT STATE OF THE ART

Location-based route discovery protocols have been widely studied [6] and can be classified in two categories: *greedy packet forwarding*, and *restricted directional flooding*. In greedy packet forwarding protocols, such as GPSR [7] and TBF [8], a packet is always forwarded to the next hop that is geographically nearest to the destination. Greedy packet forwarding protocols have good scalability, but they may cause many routing failures when nodes are not uniformly distributed.

In restricted directional flooding protocols, such as DREAM [9] and LAR [10], a route requester always uses its location information of the destination to determine the forwarding directions of route requests. In these protocols, the forwarding directions will remain the same once determined. Specifically, DREAM performs route discovery hop by hop only based on the location information of the destination, and its route discovery results cannot be reused. LAR discovers the entire route, but is sensitive to the accuracy of node locations due to the usage of global coordinates.

To acquire location information, many location-based route discovery protocols [7-10] use GPS receivers, which are not suitable for MANETs due to their high energy consumption, and low accuracy for indoor applications. GPS-free localization methods use some "beacon" nodes knowing their own locations as reference points to perform multilateration estimations for calculating the locations of other nodes [11]. In this process, physical measurements, such as RSSI [12] and ToA/TDoA [13], are used to estimate distances from a node to those beacon nodes. However, using such estimations greatly increases network communication and computation overhead, and using beacon nodes prohibits distributed localization.

III. OVERVIEW OF OUR LDRD PROTOCOL

In our LDRD protocol, we assume that the nodes are stationary or moving, and we have *a priori* knowledge of the range of their moving speeds. Based on this assumption, our LDRD protocol uses the location information of the route destination to calculate a destination area and a respondable area to indicate the direction for forwarding the route discovery request and restrict the influence range of the request. During our route discovery process, LDRD refines the respondable area and destination area on each intermediate node to improve the efficiency of route discovery.

The LDRD can be summarized as follows.

1) A route requester calculates a destination area and a respondable area. The descriptions of these areas are encapsulated in the route request before it is sent out.

2) Upon receiving a route request, a node generates its response as follows:

2a) If the node is the destination, it sends a reply to the route requester. Otherwise, go to *2b)*.

2b) The node determines its qualification for responding to the request by deciding whether it is in the destination area or the respondable area described in the request. If the node is qualified, go to *2c)*. Otherwise, the route request is discarded.

2c) If the qualified node has a route to the destination, it sends the route to the route requester. Otherwise, go to *2d)*.

2d) The node uses the location information of the route destination in its own storage to refine the destination area and respondable area in the route request, and then broadcasts the refined route request.

IV. OUR NLS-LC

In our NLS-LC, a node N_i maintains a location table containing the location information of other nodes relative to the local coordinate system of N_i, with N_i itself at the origin. Each record in the location table of a node contains the node's coordinates, IP address, a global timestamp *origintime* indicating the record generation time, and the elapsed time (*elapsetime*) since *origintime*. *elapsetime* indicates the validity level of a record.

In our NLS-LC, each node is equipped with a timer to trigger location update periodically. The location update process in our NLS-LC can be summarized as follows:

N1) Each node acquires the locations of its neighbors using RSSI [12] physical measurements.

N2) Each node broadcasts location update messages to its neighbors. The messages contain part of the records in the location table of the node. The records are selected based on their *elapsetime*.

N3) Upon receiving a location update message, the receiver node uses the message to update its own location table based on the validities of location records included in the message.

❖ *Location update message handling*

A node N_0 first uses RSSI to update the location of a neighbor N_1 upon receiving a location update message m from N_1. Then, N_0 uses m to update its location table T. For a record r_j in m, N_0 searches T to find a record r_k with the same IP address in r_j. If r_k cannot be found, r_j is added into T. Otherwise, N_0 compares the *elapsetime* values in r_j and r_k, and stores the record with a smaller *elapsetime* into T. If the *elapsetime* of r_j is smaller than that of r_k, coordinate transformation is performed on r_j before storing r_j into T. Assume that node A sends a location update message to node B. B knows that the current location of A is (x_A, y_A). B transforms the location of node C, (x_C, y_C), in the update message from A, to (x_C+x_A, y_C+ y_A) before storing it.

❖ *Selecting appropriate interval of location updates*

An important parameter for our NLS-LC is the interval of location updates. If the interval is too large, the *elapsetime* value in location records will increase, and their validities will reduce. If the interval is too small, the network overhead will increase. To determine the proper interval, we performed comparative experiments under different environment settings, including *SIZE* and *SPEED, where SIZE* is the edge length of the square area where all the mobile nodes are deployed, and *SPEED* is the estimated average speed of the nodes. We conclude that the interval of location updates in our NLS-LC should be *SIZE/10/SPEED*.

❖ *Selecting location records to be broadcasted*

When location update messages are sent, the contained location records need to be carefully selected for conserving resources. Location records with very large or small *elapsetime* should not be included in the messages since records with very large *elapsetime* are likely to be obsolete, and records with very small *elapsetime* indicate very small location changes. Therefore, proper upper and lower bounds of *elapsetime* need to be selected as record filters. To determine such upper and lower bounds, we conducted the same experiments as before, and select the interval of location updates as the upper bound of *elapsetime*, and 30% of the interval as the lower bound. Furthermore, a location record whose *elapsetime* exceeds *SIZE/2/SPEED* will be deleted.

V. LOCATION-BASED DIRECTIONAL ROUTE DISCOVERY

A. Directional Route Request

In our LDRD, the direction for forwarding a route request is determined by a destination area, called *DST*, and a respondable area, called *RESP*. The *DST* is enclosed by a circle centered at the destination node. The radius R of this circle is calculated by

$$R = (d_{max} + elapsetime) \times v_{max} \quad (1)$$

where d_{max} is the predefined maximum possible delay of route discovery, *elapsetime* is retrieved from the location record of the destination in the location table of the route requester. v_{max} is the maximum moving speed of mobile nodes. This area covers the maximum possible moving range of the destination node during the route discovery process. The analytical expression of the *DST* relative to the local coordinate system of the route requester is given by

$$C: (x - x_d)^2 + (y - y_d)^2 = R^2 \quad , \quad (2)$$

where (x_d, y_d) is the coordinate of the destination.

To restrict the influence range of a route discovery request, a *RESP* is generated based on the *DST*. Fig. 1 depicts a *DST* and the corresponding *RESP*, whose width is defined to be the angle a. In LDRD, only the nodes in the *DST* or *RESP* can respond to the route discovery request.

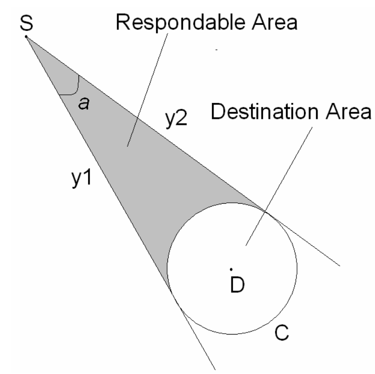


Fig. 1 *DST* and *RESP*

Under the local coordinate system of the requester, the two tangent lines in Fig. 1 are defined as follows:

$$y_1 = k_1 x \quad \text{and} \quad y_2 = k_2 x \quad (3)$$

where

$$k_1 = \frac{2x_d y_d - \sqrt{\Delta}}{2(x_d^2 - R^2)}, \quad k_2 = \frac{2x_d y_d + \sqrt{\Delta}}{2(x_d^2 - R^2)} \quad (4)$$

and

$$\Delta = 4x_d^2 y_d^2 - 4(x_d^2 - R^2)(y_d^2 - R^2) \quad (5)$$

The set of parameters used in (1) – (5), including k_1, k_2, R, x_d and y_d, are encapsulated in the route requests for other nodes to determine their qualification for processing the requests.

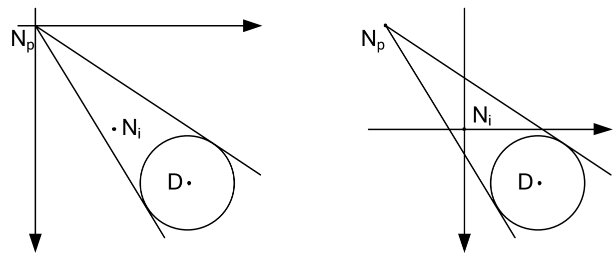
B. Behavior of intermediate nodes

Having received a route request for destination D from the previous hop N_p, an intermediate node N_i first transforms the set of parameters, S, in the route request to the local coordinate system of N_i, and then checks its qualification for processing the route request. The expression of the *DST* derived from *S* is *C* given by (2), and the *RESP* is given by (3). Since N_i is a neighbor of N_p, N_i has the up-to-date N_p location, namely (x_p, y_p), the expressions after coordinate transformation will be:

$$C': (x - x_d - x_p)^2 + (y - y_d - y_p)^2 = R^2 \quad (6)$$

$$y_1' = k_1(x - x_s) + y_s,\ y_2' = k_2(x - x_s) + y_s \quad (7)$$

This coordinate transformation is illustrated in Fig. 2.



(a) before transformation (b) after transformation

Fig. 2 Coordinate transformation

N_i determines its qualification for processing the route request by checking whether it is in the *RESP* or *DST* decided by (6) and (7). The rule to decide such qualification can be summarized as follows:

$$k_1 < y_s/x_s < k_2$$

$$\text{or} \quad \sqrt{(x_d + x_s)^2 + (y_d + y_s)^2} < R \quad (8)$$

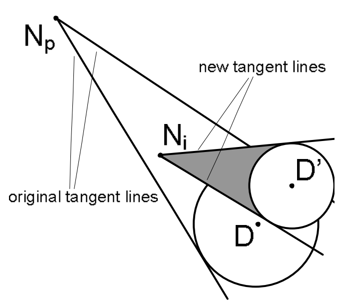


Fig. 3 Refining *DST* and *RESP*

If N_i is the route destination *D*, or has a route to *D*, it sends a route to the route requester. Otherwise, if N_i is qualified for processing the route request based on (8), it refines the *DST* and *RESP*, and updates the route request accordingly before broadcasting the request. The refinement of the two areas is needed to trace the location changes of the route destination due to node mobility. Fig. 3 illustrates such a refinement when the route destination moved from *D* to *D'*.

The refinement of the *DST* and *RESP* is done by recalculating the two areas using the latest location information of the route destination. Based on the LDRD protocol, N_i is nearer to the destination than N_p, and hence the *elapsetime* of *D'* is generally smaller than that of *D*, which indicates that *D'* is more accurate than *D*. Hence, the sizes of the recalculated *RESP* and *DST* are decreasing per hop, and the direction of route discovery is being refined more accurately.

Mobile nodes not qualified for responding the received route requests automatically drop the received requests, while qualified nodes build route entries and send updated requests. Consequently, LDRD protocol ensures that each hop produces a part of the wanted route.

As shown in Section IV, every node in LDRD uses RSSI to acquire the locations of their neighbors, but RSSI may be inaccurate in practice due to various environmental factors, such as multipath fading in wireless channels and object movements. However, LDRD is insensitive to the accuracy of location information of route destinations because a dynamic destination area, instead of a single point, is used to indicate the direction of route discovery. Consequently, the estimated location information, even with some errors, is sufficient for LDRD to complete route discovery successfully since the major part of the destination area will remain the same.

C. Expanding respondable area during route discovery

In LDRD, the respondable area is only decided by the destination area. Hence, when the nodes are not uniformly distributed, it is possible that no qualified nodes can be found to forward route requests. This will cause the route requester to send requests repetitively, and increase route discovery delay or even cause many data packets to be dropped.

We have developed two methods to overcome this difficulty. Both methods aim at expanding the respondable area by enlarging the radius of the destination area. The first method is to ensure that the respondable area has a larger width than a predefined threshold. When a node cannot find the next hop to forward the route request, the node checks whether the width of the current respondable area is smaller than the predefined threshold. If yes, the width is set to the threshold. The second method increases the radius of the destination area gradually according to the number of unsuccessful route discovery attempts, and hence also enlarges the respondable area. (9) shows how to increase the radius,

$$R = R_{original} \times (1 + (req\_count - 1) \times e_{incr}) \quad (9)$$

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

where $R_{original}$ is the original radius calculated by (1), *req_count* is the number of unsuccessful route discovery tries, and e_{incr} is a radius expansion factor defined by

$$e_{incr} = \log(SIMULATION\_SIZE \,/\, R_{original}) \qquad (10)$$

If e_{incr} is calculated to be larger than 1, it is set to be 1.

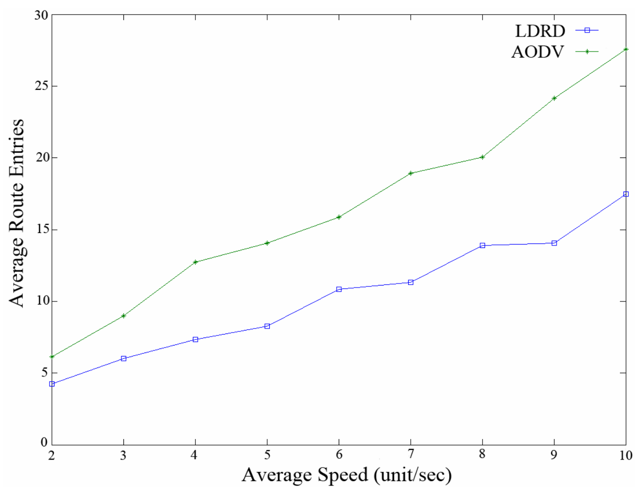
VI. Simulation Results

We have compared our LDRD protocol with AODV [3], one of the representative flooding-based, on-demand route discovery protocols in MANETs. We have simulated our LDRD protocol using ns-2 to evaluate its performance and efficiency in three aspects: network mobility, connectivity and traffic. Our simulation results have shown that our LDRD greatly improves the route discovery efficiency, and only has slightly lower network performance. Due to limited space, only part of simulation results is shown in Fig. 4 and Fig. 5.

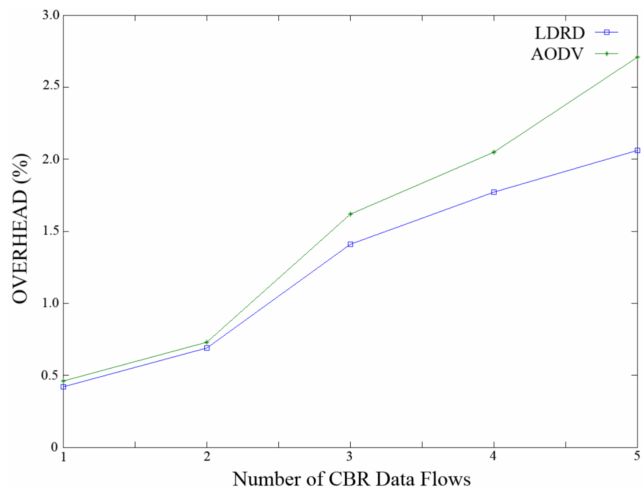
A. Simulation settings

In our simulations, the network has 50 nodes, which are confined to a 1000 unit × 1000 unit area. We assume that the MAC layer has idealized features so that its impact on our simulation results can be neglected. All the mobile nodes move at an average speed v_{avg} on randomized directions, and the actual speed is uniformly distributed in the range between $v_{avg} - \delta$ and $v_{avg} + \delta$, where v_{avg} is changed from 2 units/sec to 10 units/sec, and δ is 10% of v_{avg}. The simulation time is inversely proportional to v_{avg}. Node transmission range changed from 150 units to 400 units for various network connectivity.

Network traffic consists of a number of data flows.

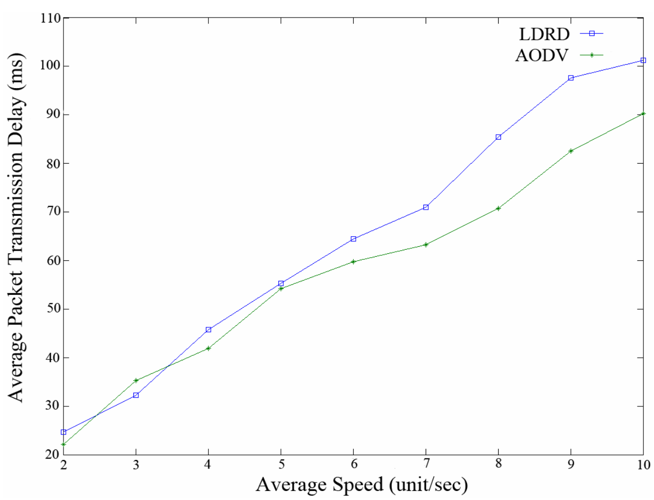


(a) AVG_RE_PER_NODE in different mobility settings

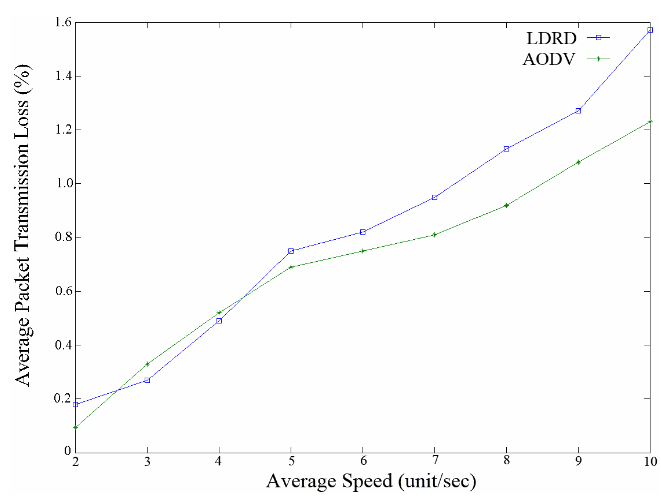


(b) OVERHEAD% in different traffic settings

Fig. 4 Comparison of the efficiency of our LDRD protocol and AODV



(a) Average Packet Transmission Delay



(b) Average Packet Transmission Loss Rate

Fig. 5 Comparison of the performance of our LDRD protocol and AODV

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

Senders and receivers are chosen randomly, and the network traffic was changed by using various number of data flows.

When we performed evaluation of one aspect, the parameters of the other two aspects are set to default values: 2 unit/sec for average speed, 250 units for transmission range, and one for the number of data flow.

B. Efficiency evaluation

In most network applications, because a majority of the route discovery overhead is caused by sending and receiving route requests, and storing route entries, we neglect the computational overhead caused by the execution of local LDRD algorithms in our efficiency evaluation.

The following metrics were used for efficiency evaluation:

(a) Average number of route entries in route table per node. By restricting the influence range of route requests, LDRD greatly reduces the number of irrelevant route entries, and hence the average size of route tables. Fig. 4(a) shows that LDRD reduced this size up to 40% in various mobility settings. This reduction is even greater in high-mobility environments. Similar reductions are also achieved in various network connectivity and traffic settings.

(b) Percentage of protocol communication overhead (OVERHEAD%) in overall network traffic. The communication overhead comes from route discovery and NLS-LC in LDRD. Fig. 4(b) shows that LDRD reduces the overhead in various network traffic settings. Similar reduction is achieved in different mobility and connectivity settings.

C. Performance evaluation

In our simulations, we use the average packet transmission delay and packet loss rate to show the network performance in packet delivery. Similar to the case in efficiency evaluation, because a majority of packet transmission delay is caused by network topology, mobility and traffic, we neglect the execution time delay of local LDRD algorithms.

Restricting the influence range of route requests will lead to slightly longer average delay. Fig. 5(a) shows that the average delay was at the same level as that of AODV in low mobility, and suffered up to 15% loss in high mobility.

Furthermore, Fig. 5(b) shows that LDRD is able to keep the similar performance in packet transmission loss rate in low mobility, and the performance deviation was controlled up to 20% in high mobility. Similar performance is also achieved in various network connectivity and traffic settings.

VII. Conclusion

In this paper, we have presented a location-based route discovery protocol LDRD with node location service NLS-LC in MANETs, which reduces route discovery overhead using directional route requests. Comparing to other location-based route discovery protocols, LDRD has better efficiency, adaptability and applicability because it iteratively refines the destination area and respondable area during the route discovery process, and is insensitive to the accuracy of location information. Future work in this area includes the improvement of LDRD by analyzing the impact of location errors and the parameters of NLS-LC. More simulations will be conducted to compare the performance and efficiency of our protocol with other protocols. We will also expand the contents of route requests and replies to incorporate service discovery capability.

References

[1] A. Crespo, and H. Garcia-Molina. "Routing Indices for Peer-to-peer Systems", *Proc. IEEE ICDCS*, 2002, pp. 23-34.

[2] F. Banaei-Kashani, et al., "WSPDS: Web Services Peer-to-peer Discovery Service", *Proc. ISWS*, 2004.

[3] C. E. Perkins, et al., "Ad-Hoc On-Demand Distance Vector Routing", *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications,* 1999, pp.90-100.

[4] D. Johnson, et al., "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Mobile Computing*, Kluwer Academic Publishers, 1996, pp. 153-181.

[5] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Proc. ACM SIGCOMM 1994*, pp. 234-244.

[6] M. Mauve, et al., "A Survey on Position-Based Routing in Mobile Ad-hoc Networks", *IEEE Network,* vol. 15(6), 2001, pp. 30-39.

[7] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Network," *Proc. ACM MOBICOM*, 2000, pp. 243-254.

[8] D. Niculescu, et al., "Trajectory Based Forwarding and Its Applications", *Proc. ACM MOBICOM*, 2003, pp. 260-272

[9] S. Basagni, et al., "A Distance Routing Effect Algorithm for Mobility", *Proc. ACM MOBICOM*, 1998, pp. 76-84.

[10] Y. B. Ko, and N. H. Vaida, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Wireless Networks,* vol. 6(4), 2000, pp. 307-321

[11] N. Patwari, et al., "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks", *IEEE Signal Processing Magazine*, vol. 22(4), 2005, pp. 54-69.

[12] S. Meguerdichian, et al., "Localized Algorithms in Mobile Ad-hoc Networks: Location Discovery and Sensor Exposure", *Proc. ACM MOBIHOC*, 2001, pp. 106-116.

[13] S. Capkun, et al., "GPS-free Positioning in Mobile Ad-Hoc Networks", *Cluster Comp.*, vol. 5(2), 2002, pp. 157-167.