A Reliable GPS-Based Routing Protocol for Mobile Ad Hoc Networks

Nen-Chung Wang and Shou-Wen Chang
Department of Computer Science and Information Engineering
Chaoyang University of Technology, Taichung 413, Taiwan, R.O.C.
Email: ncwang@mail.cyut.edu.tw

Abstract*-A mobile ad hoc network (MANET) is a dynamically reconfigurable wireless network with no fixed infrastructures. In this paper, we propose a reliable on-demand routing protocol (RORP) with mobility prediction. In this scheme, we determine the duration of time between two connected mobile nodes by using the global positioning system (GPS), and discover a request region between the source node and the destination node for routing discovery. During routing discovery, we can find many routes from the source node to the destination node. Then we select the routing path that requires the longest duration of time for transmission to increase routing reliability. We also propose a GPS-aided partial reconstruction process that selects a backup path for maintenance. When a link on a routing path will break, the route maintenance process is initiated. We use the information from two disconnected mobile nodes to plan a local request region and execute a local route discovery to find a backup path. Simulation results show that the proposed scheme outperforms the previous scheme.*

Keywords: Mobile ad hoc networks, mobile computing, mobility prediction, on-demand routing, reliable routing.

1. Introduction

A mobile ad hoc network (MANET) is a dynamically reconfigurable wireless network with no fixed infrastructures. In recent years, many routing protocols have been proposed for MANETs. We first review key routing protocols in MANETs in three broad categories: proactive routing protocols, reactive routing protocols, and location-based routing protocols.

In proactive routing protocols [6, 7], each node must maintain path information from each node to the destination node and update its routing table entries periodically. They are also called table driven protocols because routes are available as part of a well-maintained table. Although the optimal path can be found whenever a source node issues a request to communicate with another node, the proactive protocol has a lot of overhead when stale route entries are uploaded. Some typical proactive routing protocols, such as destination sequenced distance vector (DSDV) [7] and wireless routing protocol (WRP) [6], have been proposed.

In reactive routing protocols [1, 4, 8], mobile nodes maintain path information for destinations only when they need to contact the source node or relay packets. The reactive protocol spends more time establishing a routing path than the proactive protocol in MANET. Although the shortest communication path can be constructed, a very large amount of search packets exist to find a routing path. Recent examples of this approach are ad hoc on-demand distance vector (AODV) [8], dynamic source routing (DSR) [4], and relative distance micro-discovery ad hoc routing (RDMAR) [1].

In location-based protocols [2, 3, 5], some routing protocols have been proposed which uses GPS. It is assumed that a mobile node knows its current physical location, and such location information can be exploited to facilitate routing. These protocols that have been proposed are based on a geographic model. Recent examples of this approach are GPS zone routing protocol (GZRP) [2], location-aided routing (LAR) [3], and zone-based hierarchical link state (ZHLS) [5].

In this paper, we propose a reliable on-demand routing protocol (RORP) with mobility prediction. In this scheme, we determine the duration of time between two connected mobile nodes by using the global positioning system (GPS) and discover a request region between the source node and the destination node for routing discovery. We select the routing path with the longest duration of time for transmission to increase routing reliability. In addition, when a link on a routing path is broken, the routing path will be disconnected. We also propose a GPS-aided partial reconstruction process that selects a backup path for route maintenance.

The rest of this paper is organized as follows. Section 2 presents the preliminaries to the system. In Section 3, we propose a reliable on-demand routing protocol with mobility prediction. Section 4 describes the experimental results. Finally, conclusions are given in Section 5.

2. Paper Preliminaries

2.1. Expected Region and Request Region

We will introduce some notations and terminologies that are used in location-aided routing (LAR) [3]. We first need to find an expected region. If the source node *S* wants to find a path to the destination node *D*, we assume that node *S* knows that node *D* was at location *P* at time t_0. Let t_1 be the current time. Because node *S* knows that the information of node *D* includes velocity, node *S* may assume that the expected region is a circular region of radius v (t_1 - t_0). In general, it is also possible to define *v* to be the maximum speed or some other measure of the speed distribution. An example of an expected region is shown in Fig. 1.

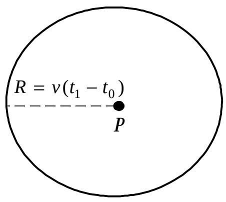

Fig. 1. An example of an expected region.

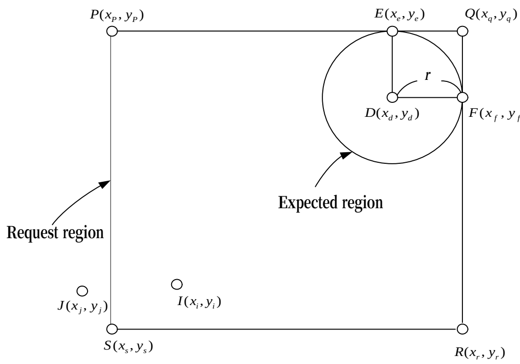

Fig. 2. An example of a request region in location-aided routing.

We must determine the request region after we determine the expected region. We use a request region that is rectangular. As shown in Fig. 2, we assume that node *S* knows the average speed *v* of node *D*. Consequently, node *S* can plan the expected region. We define the request region as a rectangular area that includes the current location of node *S*. The request region is the rectangle whose corners are *S*, *P*, *Q*, and *R*. When the source node *S* wants to search a path to the destination node *D*, node *S* sends a route request (RREQ) packet to its neighboring nodes. When a node receives a RREQ packet, it discards the RREQ packet if that node is not within the rectangle specified by the four corners included in the route request. In this case, the source node *S* wants to send data to the destination node *D*. Node *S* sends a RREQ packet to its neighboring nodes. We assume that node *I* receives a RREQ packet. Because node *I* is located in the rectangular request region, node *I* forwards the packet to its neighbors. On the other hand, if node *J* receives a RREQ packet, node *J* discards the packet because node *J* is not located in the request region.

2.2. Duration of Time

In this section, we introduce the mobility prediction method. This method uses the location information provided by GPS. We assume a free space propagation model [9] in which the signal strength solely depends on the distance to the transmitter. We also assume that all nodes have their clocks synchronized by using the GPS clock. If we know the motion parameters of two nodes, we can calculate the duration of time these two nodes remained connected. These parameters include speed, direction, and radio range and can be obtained from GPS.

We assume that two nodes *A* and *B* are within the same transition range *r* of each other. We let (x_1, y_1) be the coordinate of mobile node *A* and (x_2, y_2) be the coordinate of mobile node *B*. We let v_1 and v_2 be the mobility speeds and q_1 and q_2 ($0 \le q_1, q_2 \le 2p$) be the moving directions. Therefore, we can obtain the duration of time D_t by using the following equation:

$$D_t = \frac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2} \qquad (1)$$

Note that $a = v_1 \cos q_1 - v_2 \cos q_2$, $b = x_1 - x_2$, $c = v_1 \sin q_1 - v_2 \sin q_2$, and $d = y_1 - y_2$. In addition, the equation cannot be applied when $v_1 = v_2$ and $q_1 = q_2$, and when D_t is ∞. In order for the information from the GPS to be utilized, the packets must include extra fields. When a source node sends a RREQ packet, the packet appends its the location, direction, and speed. The next hop neighbor of the source node receives the RREQ packet to predict the duration of time between itself and the source node.

2.3. Flow Oriented Routing Protocol (FORP)

FORP [10] is an on-demand routing protocol. It applies GPS to allow all nodes get their own information. By using the information to calculate the duration of time, this protocol can get a stable path to send data packets, resulting in a decrease of overhead packets. An example is shown in Fig. 3. In Fig. 3(a), when the source node *A* wants to transmit data to the destination node *F*, node *A* sends flow-REQ to destination node *F*. Node *B* receives flow-REQ from node *A* and calculates the link expiration time (LET) between node *A* and node *B*. In this example, the destination node *F* receives two flow-REQ packets.

One packet contains path (*A*, *B*, *C*, *E*, *F*) with link expiration times (LETs) which are equal to (4, 5, 6, 7), and the other packet contains path (*A*, *B*, *D*, *E*, *F*) with LETs = (4, 3, 5, 7). Note that the route expiration time (RET) is equal to the minimum of the set of LETs for the route. Thus, node *F* can get the RET for both routes. The RET of the path (*A*, *B*, *C*, *E*, *F*) is 4, and the RET of the path (*A*, *B*, *D*, *E*, *F*) is 3. In Fig. 3(b), path (*A*, *B*, *C*, *E*, *F*) is more stable since it has a longer RET than the other path. Node *F* replies a flow-SETUP packet from the destination node to the source node. Node *A* sends data by using the path that sends the flow-SETUP packets. When a path is broken, it sends a flow-handoff message form the destination node to the source node to discover the path via flooding in the same way as the flow-REQ. It can discover another path to replace the broken path. FORP defines the critical time T_c = RET - T_d. T_d as the delay experienced by the latest packet which has arrived along the path. Thus, the protocol applies the information that was obtained by GPS to get a more reliable path.

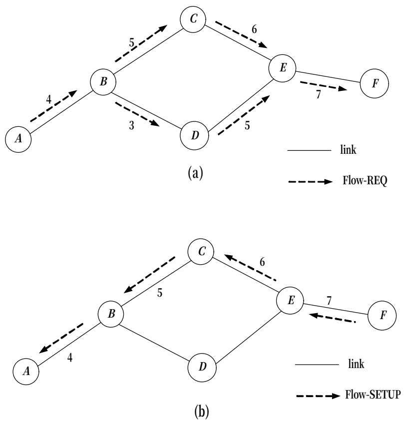

Fig. 3. FORP transmission process. (a) Flow-REQ. (b) Flow-SETUP.

3. Reliable On-Demand Routing Protocol (RORP)

In this section, we propose a reliable on-demand routing protocol (RORP) with mobility prediction. The proposed RORP includes the route discovery process and the route maintenance process.

3.1. Tables for Routing

In order to make routing efficient, we use three kinds of tables in our routing protocol: the neighboring node table, the current path table, and the maintenance table. The details of these tables are described below.

(a) Neighboring node table: A node will record in this table the information of other nodes that are within the transmission range if it can hear the "beacon packet." The format of the table is <node_ID, the location information of node_ID>.

(b) Current path table: This table contains the current path that is used for the transmission of data. The format of the current path table is <seq_number, path_table, lifetime>. Here, the format of the path_table is <node_ID$_i$, Dt$_{i,\ i+1}$>. node_ID$_i$ is a node number in the MANET and Dt$_{i,\ i+1}$ is the duration of time between node_ID$_i$ and node_ID$_{i+1}$ in the routing path.

(c) Maintenance table: This table contains the location information that is used in our maintenance process. The format of the maintenance table is <SID, DID, the location information of the last node and the next node>. We can update the entry by sending a data packet.

3.2. Route Discovery

We first introduce the route discovery process in RORP that searches for a routing path. We need to find the neighboring nodes before we execute the route discovery process. Thus, we send a beacon packet to neighboring nodes. The purpose of the beacon packets is to find the links of the neighboring nodes and to create the neighboring node table for each node. Therefore, a node will record in the neighboring node table the information of other nodes that are within the transmission range if it can hear the beacon packet messages from its neighboring nodes.

We describe here how to execute the route discovery process. We first define two parameters: the link duration time (LDT) and the route duration time (RDT). The LDT represents the duration of time between two nodes, which is calculated by using Equation (1). If we can predict the LDT along each hop on the route, we will be able to predict the RDT. The RDT is equal to the minimum of the set of LDTs for the route. Thus, the RDT is the minimum duration of time along a routing path. If a source node wants to find a path to the destination node, we assume that the source node gets the information of the destination node before the routing process. Each node can know its information by using GPS. Thus, we determine the request region and expected region before we send the search packet. The method is described in Section 2.1.

We must use two packets to search for the routing path. One is the RREQ packet and the other is the RREP packet. When a source node sends information to the destination node, it should first check its routing table to determine if it has a route to the destination node. If a route has been found, it will use the route to send the data packet immediately. Otherwise, a source node will broadcast a RREQ packet. The packet is sent to each node only within the request region. The packet contains a sequence number, source ID, destination ID, the list of node

IDs, and the list of LDT appended by the intermediate nodes when the RREQ packet traversed through the network, and the information measured by GPS. Once a node receives a RREQ packet, it will forward the packet to all of its neighbors if its neighboring nodes are in the request region and if the packet contains a higher sequence number than any of the previously received RREQ packets.

When a node first receives the RREQ packet, it adds its own ID, the duration of time that was calculated by using its information, and the information in the RREQ packet to the RREQ packet. When a RREQ packet arrives at the destination node, it records all the nodes along the route it has traversed and the duration of time of each link along the route. When the destination node receives a RREQ packet, it determines the RDT by looking in the packet to find the minimum number of LDTs along the path of the RREQ packet. We suppose that there are many routing paths that can be used. Then the destination node selects the path with the maximum RDT to be the main routing path. The destination node sends a route reply (RREP) packet to the source node along the decided path. We can find a routing path with the longest duration of time to increase routing reliability. Thus, we can use the more stable path to send data packets.

The steps of the route discovery process are described below.

Step 1: Source node *S* broadcasts a route RREQ packet to its neighboring nodes. After they receive the packet, the neighboring nodes calculate the duration of time and decide whether they are within the request region. If a neighboring node is in the request region, the node forwards the packet to its neighboring nodes and adds its ID and the LDT for the last link of the RREQ packet to the packet entry. Otherwise, the packet will be discarded if the node is not in the request region.

Step 2: When the destination node *D* receives a RREQ packet. Node *D* waits for a waiting time to receive other RREQ packets. Node *D* determines the RDT of the routing path by using the minimum of the set of LDTs along the routing path. Then node *D* selects the path with the maximum RDT to be the main routing path. Node *D* sends a RREP packet to the source node *S* along the main routing path.

Let us consider the example shown in Fig. 4. In Fig. 4(a), the source node *A* sends a RREQ packet to the destination node *I*. Nodes *B*, *C*, … , and *H* will forward the RREQ packet and append their own information, such as their own ID and the duration of time, to the RREQ packet. In this example, two RREQ packets arrive at node *I*. One packet contains path (*A*, *B*, *C*, *D*, *E*, *I*) with duration of time of LDTs = (8, 7, 6, 8, 9). The other packet contains the path (*A*, *F*, *G*, *H*, *I*) with LDTs = (5, 8, 7, 4). Node *I* can obtain the RDT evoked from the minimum LDT. In this case, the RDT of the path (*A*, *B*, *C*, *D*, *E*, *I*) is 6 and the RDT of the path (*A*, *F*, *G*, *H*, *I*) is 4. Thus, the RDT of the path (*A*, *B*, *C*, *D*, *E*, *I*) is larger than that of the path (*A*, *F*, *G*, *H*, *I*). Thus, the path (*A*, *B*, *C*, *D*, *E*, *I*) is more stable than the path (*A*, *F*, *G*, *H*, *I*). In Fig. 4(b), we select the path with the longest duration for routing to reverse a RREP packet. The source node A receives a RREP packet, and delivers the data packet along the routing path (*A*, *B*, *C*, *D*, *E*, *I*). Thus, we can get a more reliable routing path. In this case, we can predict that the main routing path will be disconnected after 6 time units. Thus, we can execute route maintenance in advance.

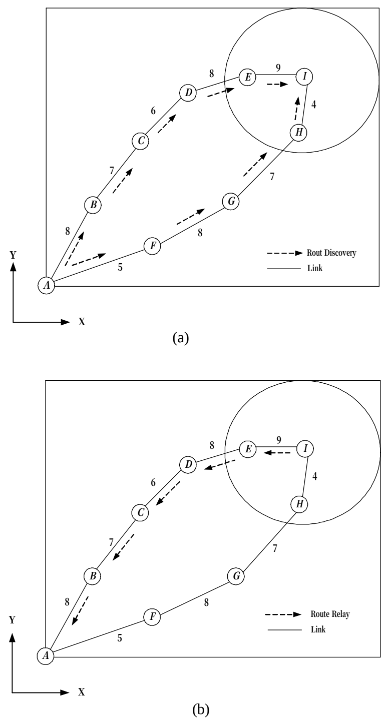

Fig. 4. Routing process. (a) The route discovery process. (b) The route reply process.

3.3. Route Maintenance

Because of the high mobility of mobile nodes, a routing path may be broken. Thus, we need to maintain the routing path. Route maintenance is usually classified into full reconstruction and partial reconstruction. In full reconstruction, one node will know a path is broken when it does not receive a RREP packet. Thus, the node sends a route error (RERR) packet to the source node. When the source node receives an RERR packet, it will reconstruct a new path from the source node to the destination node. Full reconstruction requires more control overhead to rebuild a routing path. In partial reconstruction, when a node finds that a path is broken, the node finds a backup path to replace the broken path. Because the replacement node is closer to the destination node than to the original source node, routing overhead can be reduced. Therefore, partial reconstruction spends less on overhead than full reconstruction.

We propose a GPS-aided partial reconstruction maintenance process that uses GPS to find a backup path. When a main routing path has been found after a discovery packet is sent, our proposed route maintenance process will be executed. Each of the nodes on the main path will construct a maintenance table. The maintenance table records the source ID, destination ID, last node information, next node information, and replacement node ID. When a node sends a message to the destination node, it also sends its own information as measured by GPS to the next node. The node broadcasts the packet to neighboring nodes, so a node can get the information of the last node and the next node. An example is shown in Fig. 5(a). Node *A* sends information to node *B*. The information of node *A* includes the position, velocity, and direction. Then node *B* sends its information to node *C*. As shown in Fig. 5(b), node *A* also receives the information from node *B*. For the same reason, node *B* can receive the information from node *C*. Thus, node *B* can get the information of the last node *A* and the information of the next node *C*. Thus, each node along the main path can get the information of the last node and the next node.

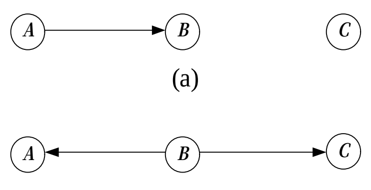

Fig. 5. Data transmission process. (a) Node *A* sends its information to *B*. (b) Node *B* sends its information to *A* and *C*.

We then use the information to do partial reconstruction. When a node finds that a path will break (the duration of time will expire), the node will find a backup path in advance. We use the information of the next node to determine a local request region. We send a local RREQ packet in the local request region to determine if there is another path that can be used to deliver the data packet.

The steps of the route maintenance are described below.

Step 1: If a path will be broken, the GPS-based partial reconstruction process is initiated to avoid the loss of a message. Each of the nodes will construct a maintenance table. A node can know the information of the last node and the next node by broadcasting packets. When a node knows that the link will break, it uses its own information and the information of the next node to determine a local request region.

Step 2: A node sends a local RREQ packet in the local request region. When a node finds a backup path, it sends data along the backup path. If it does not find a replacement node, it sends a RERR packet to the source node and restarts the route discovery process.

An example of route maintenance is shown in Fig. 6. The source node *A* sends a data packet to the destination node *E*. We assume that the link between node *B* and node *C* will break. When node *B* finds that the link between node *B* and node *C* will break, node *B* uses its own information and the information of the next node *C* to determine a local request region. Then node *B* sends a local RREQ packet to discover a backup path. In this example, node *B* sends a local RREQ packet to discover the routing path. It finds a replacement node *D* in the local request region. Then node *B* delivers data along the backup path (*B*, *D*, *C*) to the destination node *E*. If we cannot find a replacement node in the local request region, we use the last node of the broken path to send the RERR packet to the source node. We utilize the RERR packet to restart the route discovery process. Because we use GPS-aided partial reconstruction maintenance process to find a backup path, this method costs less control packet than that use a full reconstruction maintenance process. GPS-aided partial reconstruction maintenance process can decrease the control overhead and the amount of delay time in the routing process.

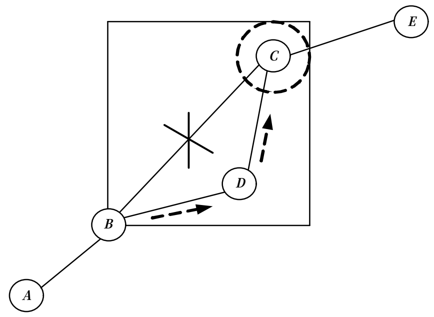

Fig. 6. An example of route maintenance.

4. Experimental Results

We conducted some simulation experiments for two routing protocols were simulated: FORP and RORP. The simulations evaluate the average routing overhead with different number of mobile nodes and mobility speed. We generate mobile nodes in a 600m×600m network. Each mobile node moves in a random direction. We assumed that the number of nodes was between 40 and 120. We send 2M data to destination node. The packet length for data packets is 2kbits. The total control overheads include RREQ packet, RREP packet, and RERR packet overhead.

The transmission range was assumed to be 150m. The routing overhead with different numbers of mobile nodes and mobility speeds is shown in Figs. 7(a) and 7(b), respectively.

Fig. 7(a) shows the routing overhead with different number of hosts. The mobility speed is assumed to be 20 km/hr. In general, increasing number of the routing overhead of FORP is faster than that of RORP when the number of nodes increased. In this case, RORP has the lower control overhead than FORP. The simulation result shows that our protocol decreases the control overhead than other protocol.

Fig. 7(b) shows the routing overhead with different mobility speeds. We assumed that the number of mobile nodes was 80. The mobility speed varied from 15km/hr to 55km/hr. When the speed of the mobile node increased, the routing path was more unreliable. The reason is that there were more chances for routes to break when the speed of the mobile node was faster. Thus, the number of rebroadcasts increased. Because our protocol can maintain a routing path in advance by using the proposed route maintenance process, the control overhead of RORP protocol was lower than that of FORP.

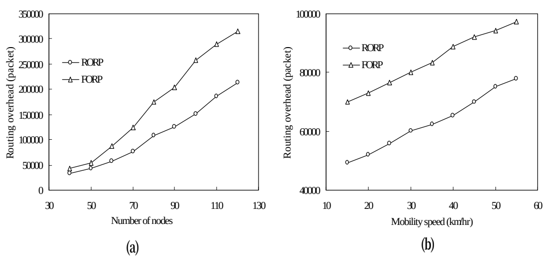

Fig. 7. Performance of routing overhead within a 150m-transmission range. (a) Routing overhead vs. number of mobile nodes with a mobility speed of 20km/hr. (b) Routing overhead vs. mobility speed with 80 mobile nodes.

5. Conclusions

In this paper, we proposed a reliable on-demand routing protocol (RORP) with mobility prediction in MANETs. We can use GPS to get the information for each node in advance and utilize this information to improve routing efficiency. Reactive routing protocols find a route using the flooding technique, which causes a waste of bandwidth and affects network performance. Thus, we utilize the information obtained by GPS to plan a request region. These packets are sent in the request region. We also calculate the duration of time between two nodes and select the path with the longest duration of time for routing. This can decrease the number of routing overhead. We also proposed a GPS-aided partial reconstruction process that selects a backup path for maintenance before a link breaks. Simulation results showed that the proposed scheme outperforms the previous scheme.

Acknowledgements

This work was supported by the National Science Council of Republic of China under grant NSC-92-2213-E-324-025.

References

[1] G. Aggelou, R. Tafazolli, "RDMAR: A Bandwidth-Efficient Routing Protocol for Mobile Ad Hoc Networks," *Proceedings of The Second ACM International Workshop on Wireless Mobile Multimedia* (WoWMoM), August 1999.

[2] A. Boukerche and S. Rogers, "GPS Query Optimization in Mobile," *Proceedings of Sixth IEEE Symposium*, pp 198-203, July 2001.

[3] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing in Mobile Ad Hoc Networks," *ACM Wireless Networks*, Vol. 6, No. 4, pp. 307-321, July 2000.

[4] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Kluwer Academic Publishers, 1996.

[5] J. N. Mario and I. T. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Netwoks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1415-1425, August 1999.

[6] S. Murthy and J. J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks," *Proceedings of ACM First International Conference on Mobile Computing and Networking*, Berkeley, CA, USA, pp. 86-95, November 1995.

[7] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proceedings of the ACM Special Interest Group on Data Communication*, London, UK, pp. 234-244, September 1994.

[8] C. E. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proceedings of the Second IEEE Workshop on Mobile Computing System and Application*, New Orleans, LA, USA, pp. 90-100, February 1999.

[9] T. S. Rappaport, "Wireless Communications: Principles and Practice," *Prentice-Hal*l, Upper Saddle River, NJ, 1995.

[10] S. William and M. Gerla, "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction," *Global Bridge Communications* (GLOBECOM), pp. 271-275, December 1999.