A Topology-Aware Routing Protocol for the MANET with Multiple Sinks

Huahui Wang* Zhiwei Cen† Tongtong Li* Matt Mutka†
*Department of Electrical and Computer Engineering
†Department of Computer Science and Engineering
Michigan State University, East Lansing, Michigan 48824, USA
Email: wanghuah,tongli@egr.msu.edu, cenzhiwe,mutka@cse.msu.edu

***Abstract*— In this paper, a topology-aware routing protocol is developed for the mobile ad-hoc networks (MANET) with multiple fixed base stations (BSes). The protocol aims at delivering packets reliably from the source node to its nearest BS. A topology-aware route recovery method is proposed to resolve route failures, and a novel "majority vote" algorithm is developed to locally update the information about the nearest BS. In addition, as a backup solution to the periodic advertisement, we propose an "on-demand" topology refreshing algorithm, timely updating the network topology at the request of the urgent source nodes. Simulation results are provided to illustrate the effectiveness of the proposed protocol.**

***Index Terms*— Ad-hoc networks, topology-aware routing, majority vote.**

I. Introduction

The idea of forming the ad-hoc network of mobile nodes has been popular for decades. More recently, the interest in the routing solutions of such networks has been growing, e.g. the formation of the MANET [1], [2] working group is focused on developing a solution framework for routing in ad-hoc networks.

Generally the routing protocols for MANET can be classified as either table-driven or demand-driven [3]. In table-driven protocols [4]–[6], each node in the network maintains one or more routing tables and periodically broadcasts its updated information into the network, which incurs substantial signaling traffic. The DSDV [4], for instance, is efficient in throughput yet limited in traffic, as the number of nodes connected to the network, say n, is very limited since the routing overhead grows as $O(n^2)$. Considering the dynamic nature of the wireless network, the demand-driven protocols [7]–[11] are regarded as more efficient routing solutions since they build the routes only when necessary. The AODV [7], for example, is efficient in saving signaling overhead but has higher latencies due to the necessity of route acquisition.

In this paper, we investigate a special MANET whose structure is differentiated from the conventional MANET by having several fixed base stations (BSes) as the sinks. A topology-aware routing protocol is developed for the network, combining the desirable features of both DSDV and AODV. In contrast to the DSDV protocol where each node broadcasts into the network, the broadcasting in the proposed protocol is only initiated by the BSes. As the number of the BSes is typically small, the resultant routing overhead is much less than that of DSDV. To resolve node failures, a demand-driven route recovery mechanism is proposed and a "majority vote" algorithm is developed to locally update the routing table.

The organization of this paper is as follows. Section II depicts the structure of the network and Section III summarizes the operations of the protocol. Details of the protocol are presented in Section IV and simulations are provided in Section V. We conclude in Section VI.

II. Network Structure and the Application

Fig. 1 illustrates a sample model of the network investigated in this paper. The mobile nodes are randomly distributed around the field. These nodes transfer packets to the fixed BSes in an ad-hoc manner by cooperating with each other. The circles shown in the figure indicate the transmission range of each node.

Such structures can easily find their applications in wireless sensor networks. A practical example is the hierarchical wireless sensor network in hospitals. In order for doctors to track and monitor the patients more conveniently, it is desirable to build a small wireless body area sensor network on each patient. Wireless sensors are attached to each patient, performing tasks such as the blood pressure measurement and the heart rate detection, etc. These sensors form a "cluster" and the sensed data is collected by a more powerful transmission unit fixed to the patient. These transmission units are called "cluster heads" and they cooperate with each other to transfer the data to the BSes distributed around the

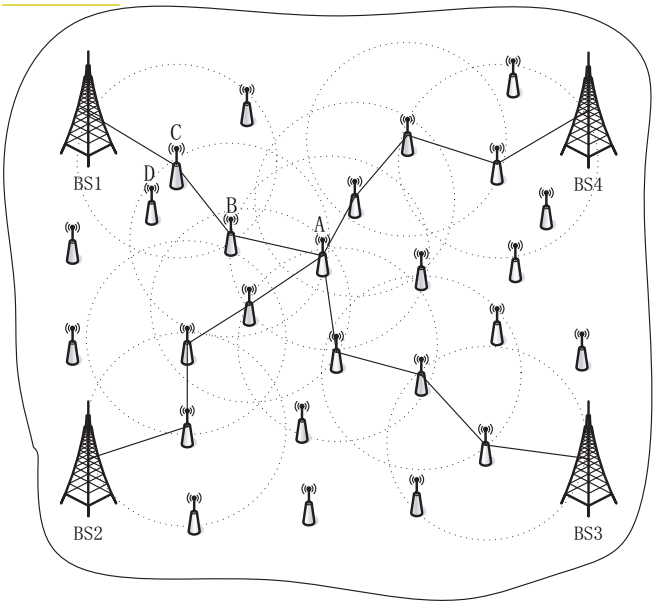

Fig. 1. Structure of the investigated network.

hospital. We can assume that Fig. 1 represents such a system. Only cluster heads and the BSes are shown in the figure, with the assumption that there are numerous sensor nodes, invisible in the figure, distributed around each cluster head.

III. Overview of the protocol

The protocol includes four major procedures.

a) Topology refreshing: The BSes initiate this procedure to set up the relative positions of the nodes in the network. It is reasonable to assume that the BSes are well coordinated and they broadcast advertisements synchronously. During the procedure, a refreshing message "REF" is broadcast into the network by each BS. Flooding or more efficient mechanisms such as SPIN [12] can be used for disseminating REFs.

When a node receives the REF, it establishes a downstream link to the node from which the message is received, and obtains the distance to the originating BS. When multiple REFs from different BSes are received, the node chooses the nearest BS as the destination to forward the data packets.

b) Link repair: Due to the the dynamic network, a node may no longer find its original downstream node to forward the packets. In this case an "RREQ" message is broadcast by the node to request the responses from potential downstream nodes and then re-establish the link. The information contained in the RREQ message includes the destined BS as well as the node's distance to this BS. Only those who are nearer to the destined BS (having a smaller distance metric in the routing table entry, with respect to the destined BS) can respond to the RREQ.

c) Topology update: If a node moves to a region where no nodes can respond to its requesting RREQ, it starts a "topology update" procedure. An "UPDT" message is originated, requiring the neighboring nodes to provide the information about which BS is the nearest and what is the distance to this BS.

The neighbors receiving the UPDT need to find out, in their own routing tables, which BSes are the nearest, and then "vote" these BSes by replying to the originator of the UPDT. In order to make the votes reliable and efficient, only those neighbors who have the valid connectivity with their downstream nodes have the right to vote.

The node originating the UPDT then collects the information from the voting neighbors and selects the BS with the majority of the votes as its nearest BS. The node voting this BS with the smallest metric is selected as the downstream node.

d) Refreshing request: If there are no reliable nodes responding to the UPDT, which may indicate that the network is already in a "chaos" state, the source node may have to wait until after the BSes disseminate the periodic REF messages. This is inconvenient if the node has some important information to send. In the protocol, an "REFREQ" message can then be broadcast by the node and relayed back to the BSes, requiring the BSes to start a new round of topology refreshing. And the protocol starts over again.

IV. Topology-aware Routing

Each node maintains a routing table which contains the following fields:

- IDs of multiple BSes;
- Downstream node to each BS;
- Metric to each BS;
- Refreshing Number (RN);
- Broadcast ID (BID).

The Refreshing Number (RN) is to indicate the freshness of the topology refreshing. It is also maintained by the BSes and initialized to be 0 either for the nodes or for the BSes. Maintaining RN is crucial to avoiding the "over-refreshing" problem addressed in Section IV-D. Besides the RN, each node maintains a Broadcast ID (BID) to distinguish multiple requests, such that the neighbors can respond accordingly.

The details of the protocol are described below.

A. Topology refreshing

The BSes broadcast the REF messages under several circumstances:

- At the start-up stage to initialize the topology refreshing;
- Periodic refreshing required by the system;
- Additional refreshing requested from the network.

The periodic refreshing is required for the purpose of system maintenance. While the additional refreshing is requested by the mobile nodes having urgent information to send and in need of timely topology refreshing.

Specifically, the REF message contains the following fields:

- Message type;
- ID of the BS originating this REF;
- Metric: Hop counts from the originating BS;
- Refreshing Number (RN).

Each time when a BS broadcasts an REF, it increments its maintained RN by 1 and copies the number into the REF message. Two REFs are regarded as duplicate if they come from the same BS and have the same RN, regardless how many hops they have been relayed. The upcoming duplicate REFs are discarded by the receiving node.

When a node receives a new REF, it compares its maintained RN with the received RN in the REF. If the received RN is smaller, the REF is regarded as a stale one and it is discarded. Otherwise, the node keeps the new RN and updates its routing table: The hop count to the BS (the one originating the REF) is updated to be 1-increment of the *Metric* in the received REF, and the node relaying the REF is saved as the corresponding downstream node.

B. Link recovery

When a node loses the contact with its original downstream node, it broadcasts to its neighbors a route request message RREQ, with time to live (TTL) value of 1. The fields of the RREQ include:

- Message type;
- Flag (F);
- Destined Nearest BS;
- Broadcast ID (BID);
- Originator Address;
- Metric.

In the protocol, there are two types of RREQ distinguished by the *Flag* field of the message. When a node cannot find its original downstream node, it broadcasts an RREQ message with flag 0 ($F = 0$) asking the potential downstream nodes to reply. The other type of the RREQ with $F = 1$, is addressed in Section IV-C. As mentioned before, besides the RN number, each node maintains a BID as well. Immediately before an RREQ with flag 0 is broadcast, the node's BID is incremented by 1 and copied into the corresponding field of the RREQ message. The *Originator Address* is the address of the node originating the RREQ.

Note that when a node originates the RREQ, it includes its distance metric with respect to the destined BS in the message. A neighbor with a smaller metric determines it is nearer to the BS after receiving this RREQ, and sends a replying message "RREP" to the originator. The format of the RREP is the same as that of the RREQ. The *Metric* in the RREP is the replying node's own metric with respect to the destined BS and the contents of other fields, except for the *Message type*, are the copies of the corresponding contents from the received RREQ.

When the originator receives an RREP with $F = 0$, it establishes a new link to the replying node and then forwards the buffered packets. The new metric to the destined BS is the 1-increment of the metric in the received RREP. The upcoming RREPs with the same *Originator Address* and BID are regarded as duplicate messages and are discarded.

As the illustration of the procedure, we refer to Fig. 1. Suppose node C, which is the original downstream node of B with respect to BS1, has moved out of the transmission range of B. B broadcasts an RREQ with flag 0 to all its neighbors requiring a new downstream node to BS1. The time to live (TTL) value of the broadcasting is only 1, i.e. the RREQ will not be forwarded by the neighbors. The metric from B to BS1, ν_{b1}, is included in the RREQ message. When the neighboring nodes receive the message, they compare their own metric (number of hops to BS1) with ν_{b1}. If a neighbor, say D, has a smaller metric ν_{d1}, it responds to B by sending its metric ν_{d1}. B then updates its routing table by setting D as its downstream node to BS1 and the corresponding metric to be $\nu_{d1} + 1$.

C. Topology update

If a node moves to a region where its downstream nodes are no longer available, it will not receive any response to the RREQ. In this case, after a request timer expires, the node starts a new procedure called "topology update", requiring the neighbors to provide some information about the nearest BS.

An "UPDT" message is originated by the node which is referred to as the *source node* in the sequel, requiring neighboring nodes to respond. The UPDT contains the address of the source node as well as its BID (incremented by 1 before UPDT is sent). After the UPDT is

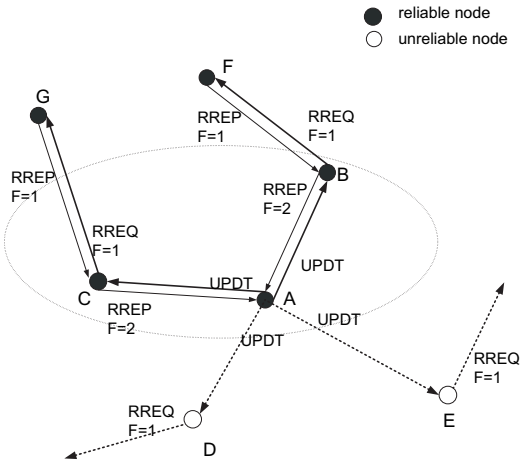

Fig. 2. Local topology update based on majority vote

received, the neighbor searches in its routing table for the nearest BS. And the searched BS is "voted" to the source node by the neighbor *after it confirms that it still has a valid link with its downstream node with respect to this BS*. In other words, a neighbor has the right to vote only when it has at least one downstream node. The nodes losing contacts with their downstream nodes are prohibited from voting, hence these nodes no longer provide misleading votes and the reliability of the votes can be greatly improved.

The procedure is as below. When a neighbor receives the UPDT from the source node, it buffers the address of the source node as well as its BID. After finding out its nearest BS in the routing table, the neighbor broadcasts an RREQ message with flag 1 ($F = 1$) to require the connectivity confirmation from the potential downstream nodes. Other than the flag field, this RREQ is distinguished from the aforementioned RREQ message ($F = 0$) in the fields of the *Originator Address* and *BID*. In the RREQ with $F = 1$, the *Originator Address* and *BID* are the address and BID of the *source node* originating the *UPDT*, instead of those of the node originating the *RREQ*. This is the major difference between the RREQ with $F = 1$ and $F = 0$. As illustrated in Fig. 2, the source node A broadcasts an UPDT with TTL=1 which is received by its neighbors B, C, D, E. These neighbors search in their routing tables the nearest BSes and broadcast the RREQ with $F = 1$. The *Originator Address* and *BID* of these RREQs are those of source node A.

The downstream nodes of B and C in Fig. 2, F, G respectively, respond to the RREQ by sending an RREP with $F = 1$, copying the *Originator Address* and *BID* of the RREQ into the corresponding fields of the RREP. After receiving the RREP, B and C claim themselves as "reliable" neighbors of A and send an RREP with $F = 2$ to A as the reply to the UPDT. In this type of RREP, e.g., the RREP with $F = 2$ from B, the field of the *Destined Nearest BS* is the nearest BS of B and the *Metric* field is B's updated hop counts after it receives the confirmation from the downstream node F.

The neighbors of the source node may receive the RREQ from each other, e.g., B may receive the RREQ from C and vice versa. Suppose F and G are no longer the downstream nodes of B and C but C happens to be the downstream node of B. When A originates an UPDT, chances are that B still has the right to vote since it has a downstream node C. A route of $A \rightarrow B \rightarrow C$ is then established. But C has no downstream node and it will in turn start a new "topology update" procedure serving as the source node. By then, A and B are its neighbors and A has the downstream node B, hence a route of $C \rightarrow A \rightarrow B$ is to be established. Such a procedure will continue on and an "isolated loop" is formed between A, B and C, as circulated in Fig. 2.

To avoid this "bounce-back" effect, when the neighbors receive the UPDT from the source, they buffer the address of the source node as well as its BID. If they receive the RREQ with $F = 1$ afterwards, they compare the buffered information with the *Originator Address* and *BID* fields in the received RREQ. When the information is matched, which means the RREQ is from another neighbor receiving the UPDT, the RREQ is then discarded. In short, the neighbors of the source node are prohibited to accept RREQs from each other during the topology update procedure.

When the source node broadcasts the UPDT, it starts a timer and then collects the "votes" from its "reliable" neighbors. After the timer expires, the node ignores the upcoming RREPs and processes the collected information. The BS voted by the majority of the neighbors is determined as the source node's nearest BS. Correspondingly, the neighbor who votes this BS and has the smallest metric is saved as the source node's downstream node.

D. Refreshing request

When the network is in a "chaos" state, a node may not have any reliable neighbors responding to the UPDT message. As shown in Fig. 2, if F and G are no longer the downstream nodes of B and C, none of A's neighbors votes at all. And A may have to buffer its packets until the network is reordered through the system's periodic refreshing. This is not satisfying if A has some urgent information to send. We provide here an "on-demand" solution, allowing the nodes with urgent messages to adaptively request the topology refreshing.

In Fig. 2, when A receives no votes at all, a refreshing request REFREQ is broadcast into the network and relayed to the BSes, asking them to start a new round of topology refreshing. The RN of node A is included in the REFREQ. If, around the same time, there are various nodes in the network broadcasting REFREQs, these REFREQs are "identical" since they contain the same RN. Even if there is only one node broadcasting the REFREQ, due to the path diversity, multiple copies of the same REFREQ can be received by the BSes. If a BS broadcasts an REF whenever it receives an REFREQ, those identical REFREQs will result in the "over-refreshing" problem, making the BS broadcast the REFs again and again.

To resolve the problem, after the BS receives the the REFREQ, it compares its maintained RN with the received RN in the REFREQ. If the received RN in the REFREQ is smaller than the RN maintained by the BS, the received REFREQ is discarded and the BS remains silent; otherwise, the BS increments the received RN by 1 and saves this updated number as its new RN. And an REF with the new RN is broadcast. In the paper, we suppose all BSes are well coordinated hence whenever a BS updates its RN, other BSes change their RNs to the same number. Such a consideration is based on the fact that if the BSes do not synchronize their RN number, then the REF from the BS with a smaller RN will not be accepted by the nodes in the network.

V. Simulations

To evaluate the performance of the proposed protocol, simulations are carried out in this section using the network simulator NS2 [13]. Due to the speciality of the network we investigate, no completely fair comparison can be made between the existing protocols and the proposed one. Furthermore, to the best knowledge of the authors, there is no similar work addressing the topic of this paper. Hence in our simulations, we present only the results of the proposed protocol, investigating the performances for different scenarios.

A. Simulation Setup

Four fixed BSes are placed at each corner of a $150m \times 150m$ rectangular area to collect data. Networks with size of 70 and 90 mobile nodes are simulated. Each node assumes a radio range of $20m$. The movements of the mobile nodes follow the Random Waypoint model. In the simulations, each node moves to a random spot within the area, rests for 30 seconds and then heads for another random spot. The speed of the movement is varied over a wide range in the simulation experiments.

A simple datagram workload model is used. All data packets are 1000 bytes long, and the mean of the interarrival times is $500ms$. A total of 7 source nodes are assumed in the networks. There is no flow or congestion control in the workload model. The mobile nodes and the BSes are simulated to work with Orinoco 802.11b network card with a data rate of 11Mbit/sec. All experiments are run for 200 simulated seconds, and each point in the following performance plots represents an average of 50 simulation runs.

B. Simulation Results

In Fig. 3, the fraction of packets dropped, which is measured as the ratio of the number of data packets dropped to the number of data packets sent by the source node, is presented with the varying mobility of the nodes in the network. As it is illustrated, even for relatively highly dynamic network, the fraction of packets dropped is no more than 4% for the network with 70 nodes, and the number is even smaller for the network with 90 nodes. This indicates the reliability of the protocol.

Generally, the packets loss increases with the nodes mobility. And it can be seen from Fig. 3, the network with 90 nodes can deliver more packets than the network with 70 nodes. The underlying reason is that, under the proposed protocol, a larger size of network provides more diversity for the node to find the downstream nodes.

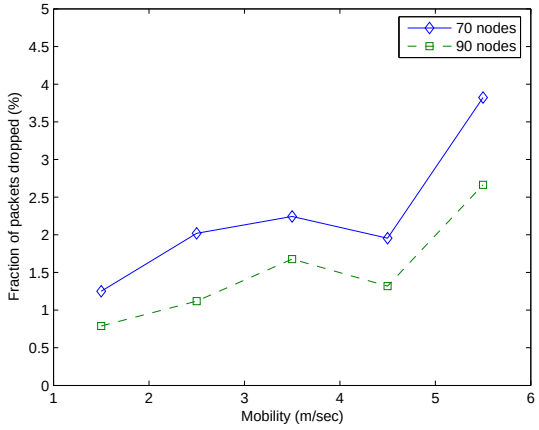

Fig. 3. Fraction of packets dropped vs. average mobility for networks of 70 nodes and 90 nodes.

Although nodes diversity is good to keep a desirable packet loss rate, it results in a little more latency in packets delivery, as can be seen in Fig. 4. With the proposed protocol, a larger network can provide more opportunities for a node to find reliable neighbors, and more opportunities for the node to establish new links to

the voted nearest BS. However, extra time is cost during these procedures, which leads to the delay difference of the two networks in Fig. 4.

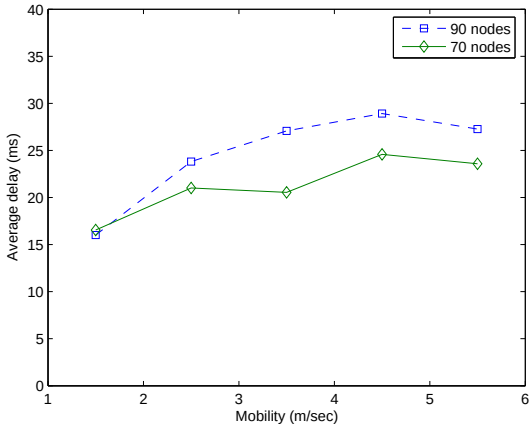

Fig. 4. Average delay vs. average mobility for networks of 70 nodes and 90 nodes.

In our simulations, the BSes initialize the topology refreshing at the beginning of the simulations, and respond afterwards to the REFREQs at the request of the nodes in the network. Here we assume all source nodes in the networks have urgent data to report. Fig. 5 demonstrates the average topology refreshing intervals versus the network dynamic. It is shown that, the relatively sparse network (70 nodes) has a general larger refreshing interval, i.e., the 70-node network refreshes the topology less frequently than the denser network with 90 nodes and hence it is more bandwidth efficient. However, as already illustrated in Fig. 3, such efficiency is at the expense of more packets loss. Hence in network design, tradeoffs should be made in view of all the aspects involving the delivery reliability, latency and bandwidth efficiency, etc.

VI. Conclusions

In this paper, we have presented a routing protocol that is well suited for the ad-hoc network with multiple sinks. By exploiting the topology information, the nodes in the network can perform dynamic route recovery and transfer data to their nearest BSes. A topology update approach based on "majority vote" is proposed in the protocol, as well as the adaptive topology refreshing algorithm. Simulation results demonstrate the effectiveness of the protocol.

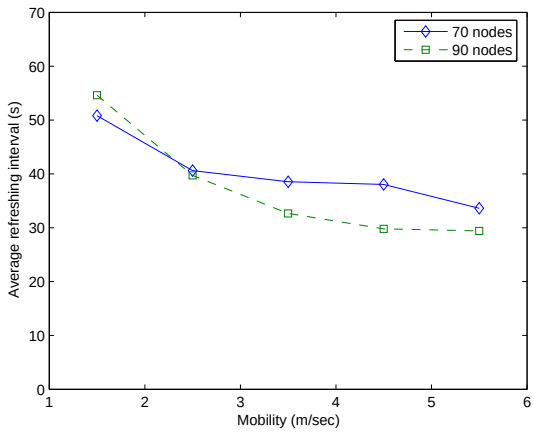

Fig. 5. Average refreshing interval vs. average mobility for networks of 70 nodes and 90 nodes.

References

[1] S. Corson, J. Macker, and S. Batsell, "Architectual considerations for mobile mesh networking," *http://tonnant.itd.nrl.navy.mil/nmnet/mmnetRFC.txt*, 1996.

[2] J. Macker and S. C. (chairs), "Mobile ad-hoc networks (manet)," *http://www.ietf.org/html.charters/manet-charter.html*, 1997.

[3] Elizabeth M. Royer and C-K Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Personal Communications*, 1999.

[4] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Computer Communications Review*, pp. 234–244, 1994.

[5] C.-C. Chiang, H.K.Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proceedings of IEEE SICON'97*, pp. 197–211, 1997.

[6] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks*, pp. 183–197, 1997.

[7] C. E. Perkins, E. Belding-Royer, and S. R Das, "RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF RFC*, 2003.

[8] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Mobile Computing*, pp. 153–181, 1996.

[9] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of INFOCOM '97*, 1997.

[10] C-K. Toh, "A Novel Distributed Routing Protocol To Support Ad-Hoc Mobile Computing," *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communication*, pp. 480–486, 1996.

[11] R. Dube, C. D. Rais, K.-Y. Wang, and S.K. Tripathi, "Signal Stability based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks," *IEEE Personal Communications*, pp. 36–45, 1997.

[12] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," *Proc. 5th ACM/IEEE Mobicom Conference (MobiCom'99)*, pp. 174–185, Aug 1999.

[13] NS, "VINT Project U.C. Berkeley/LBNL, NS2 Network Simulator," *http://www-mash.cs.berkeley,edu/ns/*.