www.ijaret.org Vol. 2, Issue I, Jan. 2014
ISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS…..

Probabilistic Rebroadcast Protocol based on Neighbor Coverage to Reduce Routing Overhead in MANETs

Shakeela Attikeri[1], Preeti Patil[2]

[1]Department of Computer Science and Engineering, Visvesvaraya University
GNDEC, Bidar 585403, India
shakeela.attikeri@gmail.com

[2] Department of Information Science and Engineering, Visvesvaraya University
GNDEC, Bidar 585403, India
preeti.patil6@gmail.com

Abstract: *Broadcasting is a fundamental and effective data dissemination mechanism for route discovery in ad hoc networks. There causes broadcast storm problem due to frequent rebroadcasting. To discover the route better than broadcasting methodology rebroadcast can done with the help of neighbor knowledge methods. In this paper, we propose a neighbor coverage-based probabilistic rebroadcast protocol for reducing routing overhead in MANETs. Our approach combines the advantages of the neighbor coverage knowledge and the probabilistic mechanism, which can significantly decrease the number of retransmissions so as to reduce the routing overhead, and can also improve the routing performance.*

Keywords: *Mobile ad hoc networks, neighbor coverage, network connectivity, probabilistic rebroadcast, routing overhead.*

1. INTRODUCTION

Nodes in MANETs can be dynamically self-organized into arbitrary topology networks without a fixed infrastructure. One of the fundamental challenges of MANETs is the design of dynamic routing protocols with good performance and less overhead. Many routing protocols, such as Ad hoc On-demand Distance Vector Routing (AODV) [1] and Dynamic Source Routing (DSR) [2], have been proposed for MANETs. The above two protocols are on-demand routing protocols, and they could improve the scalability of MANETs by limiting the routing overhead when a new route is requested [3]. However, due to node mobility in MANETs, frequent link breakages may lead to frequent path failures and route discoveries, which could increase the overhead of routing protocols and reduce the packet delivery ratio and increasing the end-to-end delay [4]. Thus, reducing the routing overhead in route discovery is an essential problem. The conventional on-demand routing protocols use flooding to discover a route. They broadcast a Route Request (RREQ) packet to the networks, and the broadcasting induces excessive redundant retransmissions of RREQ packet and causes the broadcast storm problem [5], which leads to a considerable number of packet collisions, especially in dense networks [6].Therefore, it is indispensable to optimize this broadcasting mechanism. Some methods have been proposed to optimize the broadcast problem in MANETs in the past few years. Williams and Camp [7] categorized broadcasting protocols into four classes: “simple flooding, probability-based methods, area-based methods, and neighbor knowledge methods.” For the above four classes of broadcasting protocols, they showed that an increase in the number of nodes in a static network will degrade the performance of the probability-based and area-based methods [7]. Kim et al. [8] indicated that the performance of neighbor knowledge methods is better than that of area-based ones, and the performance of area-based methods is better than that of probability-based ones.

We now obtain the initial motivation of our protocol: Since limiting the number of rebroadcasts can effectively optimize the broadcasting [5], and the neighbor knowledge methods perform better than the area-based ones and the probability-based ones [8], then we propose a neighbor coverage-based probabilistic rebroadcast (NCPR) protocol which helps to keep the network connectivity and reduce the redundant retransmissions, we need a metric named connectivity factor to determine how many neighbors should receive the RREQ packet.

The main contributions of this paper are as follows:

- We propose a novel scheme to calculate the rebroadcast delay. The rebroadcast delay is to determine the forwarding order. The node which has more common neighbors with the previous node has the lower delay. If this node rebroadcasts a packet, then more common neighbors will know this fact. Therefore, this rebroadcast delay enables the information that the nodes have transmitted the packet spread to more neighbors, which is the key to success for the proposed scheme.
- We also propose a novel scheme to calculate their broadcast probability. The scheme considers the information about the uncovered neighbors (UCN), connectivity metric and local node density.

The rest of this paper is organized as follows: Section 2 introduces the related previous work. Section 3 proposes a Neighbor Coverage-based Probabilistic Rebroadcast protocol for reducing routing overhead in route discovery. Section 4 presents simulation parameters and scenarios which are used to investigate the performance of the proposed protocol. Section 5 concludes this paper.

2. RELATED WORK

Broadcasting can be quite large, especially in high dynamic networks [9], [11]. Ni et al. [5] studied the broadcasting protocol analytically and experimentally, and showed that the rebroadcast is very costly and consumes too much network resource. The broadcasting incurs large routing overhead and causes many problems such as redundant retransmissions, contentions, and collisions [5]. Thus, optimizing the broad-casting in route discovery is an effective solution to improve the routing performance. Haas et al. [10] proposed a gossip-based approach, where each node forwards a packet with a probability. They showed that gossip-based approach can save up to 35 percent overhead compared to the flooding. However, when the network density is high or the traffic load is heavy, the improvement of the gossip-based approach is limited [9]. Abdulai et al. [12] proposed a Dynamic Probabilistic Route Discovery (DPR) scheme based on neighbor coverage. In probability according to the number of its neighbors and the set of neighbors which are covered by the previous broadcast. This scheme only considers the coverage ratio by the previous node, and it does not consider the neighbors receiving the duplicate RREQ packet.

Several robust protocols have been proposed in recent years besides the above optimization issues for broadcasting. Chen et al. [13] proposed an AODV protocol with Directional Forward Routing (AODV-DFR) which takes the directional forwarding used in geographic routing into AODV protocol. While a route breaks, this protocol can automatically find the next-hop node for packet forwarding. Keshavarz-Haddad et al. Stann et al. [14] proposed a Robust Broadcast Propagation (RBP) protocol to provide near-perfect reliability for flooding in wireless networks, and this protocol also has a good efficiency. In our protocol, we also set a deterministic rebroadcast delay, but the goal is to make the dissemination of neighbor knowledge much quicker.

3. NEIGHBOR COVERAGE BASED PROBABILISTIC RE-BROADCASTING PROTOCOL

In this section, we calculate the rebroadcast delay and rebroadcast probability of the proposed protocol. We use the upstream coverage ratio of an RREQ packet received from the previous node to calculate the rebroadcast delay, and use the additional coverage ratio of the RREQ packet and the connectivity factor to calculate the rebroadcast probability in our protocol, which requires that each node needs its 1-hop neighborhood information.

3.1 Uncovered Neighbors Set and Rebroadcast Delay

When node n_i receives an RREQ packet from its previous node s, it can use the neighbor list in the RREQ packet to estimate how many its neighbors have not been covered by the RREQ packet from s. If node n_i has more neighbors uncovered by the RREQ packet from s, which means that if node n_i rebroadcasts the RREQ packet, the RREQ packet can reach more additional neighbor nodes. To quantify this, we define the UnCovered Neighbors set $U(n_i)$ of node n_i as follows:

$$U(n_i) = N(n_i) - [N(n_i) \cap N(s)] - \{s\}$$

where $N(s)$ and $N(n_i)$ are the neighbors sets of node s and n_i, respectively. s is the node which sends an RREQ packet to node n_i.According to above eqn we obtain the initial UCN set. Due to broadcast characteristics of an RREQ packet, node n_i can receive the duplicate RREQ packets from its neighbors. Node n_i could further adjust the $U(n_i)$ with the neighbor knowledge. In order to sufficiently exploit the neighbor knowledge and avoid channel collisions, each node should set a rebroadcast delay.

The choice of a proper delay is the key to success for the proposed protocol because the scheme used to determine the delay time affects the dissemination of neighbor coverage knowledge. When a neighbor receives an RREQ packet, it could calculate the rebroadcast delay according to the neighbor list in the RREQ packet and its own neighbor list. The rebroadcast delay $T_d(n_i)$of node n_i is defined as follows:

$$T_p(n_i) = 1 - \frac{|N(s) \cap N(n_i)|}{|N(s)|}$$

$$T_d(n_i) = MaxDealy \times T_p(n_i)$$

where $T_p(n_i)$ is the delay ratio of node n_i, and $MaxDealy$ is a small constant delay.

The above rebroadcast delay is defined with the following reasons: First, the delay time is used to determine the node transmission order. To sufficiently exploit the neighbor coverage knowledge, it should be disseminated as quickly as possible. When node s sends an RREQ packet, all its neighbors $n_i = 1,2, \ldots |N(s)|$ receive and process the RREQ packet. We assume that node n_k has the largest number of common neighbors with node s, according to second eqn, node n_k has the lowest delay. Once node n_k rebroadcasts the RREQ packet, there are more nodes to receive it, because node n_k has the largest number of common neighbors. Then, there are more nodes which can exploit the neighbor knowledge to adjust their UCN sets. Of course, whether node n_k rebroadcasts the RREQ packet depends on its rebroadcast probability calculated in the next section.

3.2 Neighbor Knowledge and Rebroadcast Probability

The node which has a larger rebroadcast delay may listen to RREQ packets from the nodes which have lower one. For example, if node n_i receives a duplicate RREQ packet from its neighbor n_j, it knows that how many its neighbors have been covered by the RREQ packet from n_j. Thus, node n_i could further adjust its UCN set according to the neighbor list in the RREQ packet fromn_j. Then, the$U(n_i)$ can be adjusted as follows:

$$U(n_i) = U(n_i) -| U(n_i) \cap U(n_j)$$

After adjusting the$U(n_i)$, the RREQ packet received from n_j is discarded. When the timer of the rebroadcast delay of node n_i expires, the node obtains the final UCN set. The nodes belonging to the final UCN set are the nodes that need to receive and process the RREQ packet. Note that, if a node does not sense any duplicate RREQ packets from its neighborhood, its UCN set is not changed, which is the initial UCN set. Now, we study how to use the final UCN set to set the rebroadcast probability. We define the additional coverage ratio node n_i as

$$(R_a(n_i)) = \frac{|U(n_i)|}{|N(n_i)|}$$

This metric indicates the ratio of the number of nodes that are additionally covered by this rebroadcast to the total number of neighbors of noden_i . As R_a becomes bigger, more nodes will be covered by this rebroadcast, and more nodes need to receive and process the RREQ packet, and, thus, the rebroadcast probability should be set to be higher.

We assume the ratio of the number of nodes that need to receive the RREQ packet to the total number of neighbors of node n_i is$F_c(n_i)$. In order to keep the probability of network connectivity approaching 1, we have a heuristic formula: $N(n_i).F_c(n_i) \geq 5.1774 \log n$ Then, we define the minimum $F_c(n_i)$ as a connectivity factor, which is

$$F_c(n_i) = \frac{N_c}{|N(n_i)|}$$

where N_c= 5.1774 log n, and n is the number of nodes in the network.From above eqn, we can observe that when$||N(n_i)|$ is greater than N_c, $F_c(n_i)$is less than 1. That means node n_i is in the dense area of the network, then only part of neighbors of node n_i forwarded the RREQ packet could keep the network connectivity.

Combining the additional coverage ratio and connectivity factor, we obtain the rebroadcast probability$pre(n_i)$ of noden_i :

$$pre(n_i) = F_c(n_i) . (R_a(n_i))$$

where, if the $pre(n_i)$ is greater than 1, we set the$pre(n_i)$ to 1.

The above rebroadcast probability is defined with the following reason. Although the parameter R_a reflects how many next-hop nodes should receive and process the RREQ packet, it does not consider the relationship of the local node density and the overall network connectivity. The parameter$F_c$is inversely proportional to the local node density. That means if the local node density is low, the parameter F_c increases the rebroadcast probability, and then increases the reliability of the NCPR in the sparse area. If the local node density is high, the parameter $F_c$could further decrease the rebroadcast probability, and then further increases the efficiency of NCPR in the dense area. Thus, the parameter F_c adds density adaptation to the rebroadcast probability.

The formal description of the Neighbor Coverage-based Probabilistic Rebroadcast for reducing routing overhead in route discovery is shown in Algorithm 1.

Algorithm1: NCPR

Definitions:

$RREQ_v$: RREQ packet received from node v.

R_v.id: the unique identifier (id) of $RREQ_v$.

$N(u)$: Neighbor set of node u.

$U(u,x)$: Uncovered neighbors set of node u for RREQ whose id is x.

$Timer(u,x)$: Timer of node u for RREQ packet whose id is x. {Note that, in the actual implementation of NCPR protocol, every different RREQ needs a UCN set and a Timer}.

```
if n_i receives a new $RREQ_s$ from s then
{Compute initial uncovered neighbors set $U(n_i, R_s$.id) for $RREQ_s$:}
$U(n_i, R_s$.id)= $N(n_i) - [N(n_i) \cap N(s)] - \{s\}$
{Compute the rebroadcast delay $T_d(n_i)$}
Calculate $T_p(n_i)$
$T_d(n_i) = MaxDealy \times T_p(n_i)$
Set a $Timer(n_i, R_s$.id) according to $T_d(n_i)$
end if
while $n_i$ receives a duplicate $RREQ_j$ from $n_j$ before $Timer(n_i, R_s$.id)expires do
{Adjust $U(n_i, R_s$.id)}
$U(n_i, R_s$.id)= $U(n_i, R_s$.id)-[ $U(n_i, R_s$.id) $\cap N(n_i)$
discard(RREQ_j)
end while
15: if $Timer(n_i, R_s$.id) expires then
16. Compute the rebroadcast prob $pre(n_i)$:}
17. Compute $(R_a(n_i))$
Compute $F_c(n_i)$
19. Compute $pre(n_i) = F_c(n_i) . (R_a(n_i))$
if Random(0,1) $\leq pre(n_i)$ then
broadcast ($RREQ_s$)
else
discard($RREQ_s$)
End if
```

4. PROTOCOL IMPLEMENTATION AND PERFORMANCE EVALUATION

4.1 Protocol Implementation

We modify the source code of AODV in NS-2 (v2.30) to implement our proposed protocol. Note that the proposed NCPR protocol needs Hello packets to obtain the neighbor information, and also needs to carry the neighbor list in the RREQ packet. Therefore, in our implementation, some techniques are used to reduce the overhead of Hello packets and neighbor list in the RREQ packet, which are described as follows:

In order to reduce the overhead of Hello packets, we do not use periodical Hello mechanism. Since a node sending any broadcasting packets can inform its neighbors of its existence, the broadcasting packets such as RREQ and route error (RERR) can play a role of Hello packets. We use the following mechanism to reduce the overhead of Hello packets: Only when the time elapsed from the last broadcasting packet (RREQ, RERR, or some other broadcasting packets) is greater than the value of Hello Interval, the node needs to send a Hello packet. The value of Hello Interval is equal to that of the original AODV.

In order to reduce the overhead of neighbor list in the

RREQ packet, each node needs to monitor the variation of its neighbor table and maintain a cache of the neighbor list in the received RREQ packet. We modify the RREQ header of AODV, and add a fixed field num_neighbors which represents the size of neighbor list in the RREQ packet and following the num_neighbors is the dynamic neighbor list. In the interval of two close followed sending or forwarding of RREQ packets, the neighbor table of any node n_i has the following three cases:

- if the neighbor table of node n_i adds at least one new neighbor n_i , then noden_i sets the num_-neighbors to a positive integer, which is the number of listed neighbors, and then fills its complete neighbor list after the num_neighbors field in the RREQ packet. It is because that node n_j may not have cached the neighbor information of node n_i, and, thus, node n_i needs the complete neighbor list of node n_i;
- if the neighbor table of node n_i deletes some neighbors, then node n_i sets the num_neighbors to a negative integer, which is the opposite number of the number of deleted neighbors, and then only needs to fill the deleted neighbors after the num_neighbors field in the RREQ packet;
- if the neighbor table of node n_i does not vary, node n_i does not need to list its neighbors, and set the num_neighbors to 0.

4.2 Simulation Environment

In order to evaluate the performance of the proposed NCPR protocol, we compare it with some other protocols using the NS-2 simulator. In this paper, we just study one of the applications: route request in route discovery. In order to compare the routing performance of the proposed NCPR protocol

Simulation parameters are as follows:

Simulation Parameter	Value
Simulator	NS-2 (v2.30)
Topology Size	1000m × 1000m
Number of Nodes	50, 100, 150, ... , 300
Transmission Range	250m
Bandwidth	2Mbps
Interface Queue Length	50
Traffic Type	CBR
Number of CBR Connections	10, 12, 14, ..., 20
Packet Size	512 bytes
Packet Rate	4 packets/sec
Pause Time	0s
Min Speed	1 m/s
Max Speed	5 m/s

We evaluate the performance of routing protocols using the following performance metrics:

MAC collision rate: the average number of packets (including RREQ, route reply (RREP), RERR, and CBR data packets) dropped resulting from the collisions at the MAC layer per second.

Normalized routing overhead: the ratio of the total packet size of control packets (include RREQ, RREP, RERR, and Hello) to the total packet size of data packets delivered to the destinations. For the control packets sent over multiple hops, each single hop is counted as one transmission. To preserve fairness, we use the size of RREQ packets instead of the number of RREQ packets, because the DPR and NCPR protocols include a neighbor list in the RREQ packet and its size is bigger than that of the original AODV.

Packet delivery ratio: the ratio of the number of data packets successfully received by the CBR destinations to the number of data packets generated by the CBR sources.

Average end-to-end delay: the average delay of successfully delivered CBR packets from source to destination node. It includes all possible delays from the CBR sources to destinations.

4.3 Performance with varied number of nodes.

Fig. 1 shows the effects of network density on the MAC collision rate. In the IEEE 802.11 protocol, the data and control packets share the same physical channel. In the conventional AODV protocol, the massive redundant rebroadcast incurs many collisions and interference, which leads to excessive packets drop. This phenomenon will be more severe with an increase in the number of nodes. Compared with the conventional AODV protocol, the NCPR protocol reduces the MAC collision rate by about 92.8 percent on the average.

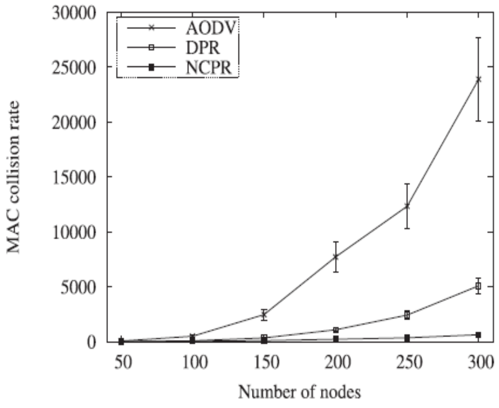

Figure 1: MAC collision rate with varied no. of nodes

4.4 Performance with varied Random packet Loss Rate

Fig. 2 shows the effects of the packet loss rate on the MAC collision rate. In our simulation parameters, we use both the Incoming ErrrProc and Outgoing ErrProc options at the same time; thus, the packet error will be more often and the retransmissions caused by random packet loss at MAC layer will be more. Therefore, the MAC collision rate of all the three routing protocols increases as the packet loss rate increases. The DPR and NCPR protocols do not consider robustness for packet loss, but they can reduce the redundant rebroadcast and alleviate the channel congestion, thus, both of them have the lower packet drops caused by collisions than the conventional AODV protocol. Compared with the conventional AODV protocol, the NCPR protocol reduces the MAC collision rate by about 92.8 percent on the average.

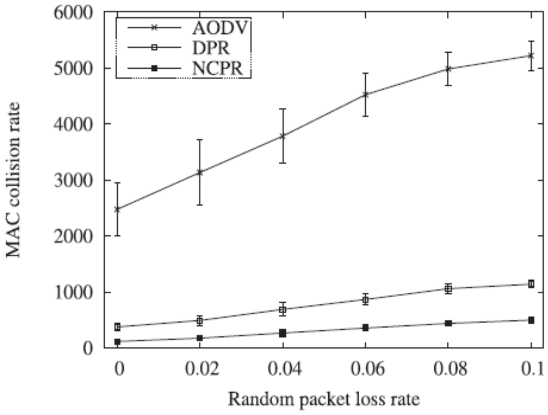

Figure 2: MAC collision rate with varied random packet loss rate

Fig. 3 shows the packet delivery ratio with increasing packet loss rate. As the packet loss rate increases, the packet drops of all the three routing protocols will increase. Therefore, all the packet delivery ratios of the three protocols increase as packet loss rate increases.

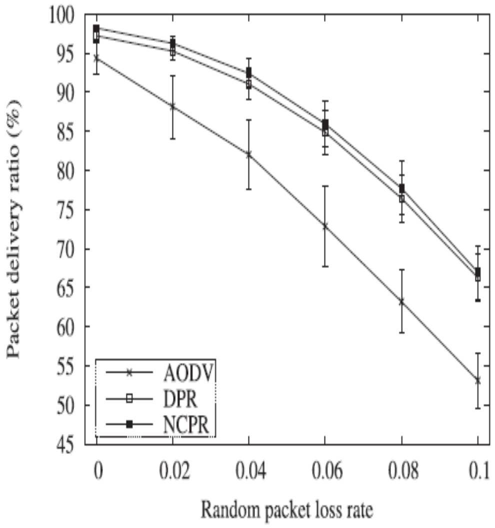

Figure 3: Packet delivery ratio with varied random packet loss rate

Both the DPR and NCPR protocols do not exploit any robustness mechanism for packet loss, but both of them can reduce the redundant rebroadcast, so as to reduce the packet drops caused by collision. Therefore, both the DPR and NCPR protocols have a higher packet delivery ratio than the conventional AODV protocol. On average, the packet delivery ratio is improved by about 15.5 percent in the NCPR protocol when compared with the conventional AODV protocol. And in the same situation, the NCPR protocol improves the packet delivery ratio by about 1.3 percent when compared with the DPR protocol.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a probabilistic rebroadcast protocol based on neighbor coverage to reduce the routing overhead in MANETs. This neighbor coverage knowledge includes additional coverage ratio and connectivity factor. We proposed a new scheme to dynamically calculate the rebroadcast delay, which is used to determine the forward-ing order and more effectively exploit the neighbor cover-age knowledge. Simulation results show that the proposed protocol generates less rebroadcast traffic than the flooding and some other optimized scheme in literatures. Because of less redundant rebroadcast, the proposed protocol mitigates the network collision and contention, so as to increase the packet delivery ratio and decrease the average end-to-end delay. The simulation results also show that the proposed protocol has good performance when the network is in high density or the traffic is in heavy load.

REFERENCES

[1] C. Perkins, E. Belding-Royer, and S. Das, Ad Hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, 2003.

[2] D. Johnson, Y. Hu, and D. Maltz, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) for IPv4, IETF RFC 4728, vol. 15, pp. 153-181, 2007.

[3] H. AlAamri, M. Abolhasan, and T. Wysocki, "On Optimising Route Discovery in Absence of Previous Route Information in MANETs," Proc. IEEE Vehicular Technology Conf. (VTC), pp. 1-5, 2009.

[4] X. Wu, H.R. Sadjadpour, and J.J. Garcia-Luna-Aceves, "Rout-ing Overhead as a Function of Node Mobility: Modeling Framework and Implications on Proactive Routing," Proc. IEEE Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS '07), pp. 1-9, 2007.

[5] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," Proc. ACM/IEEE MobiCom, pp. 151-162, 1999.

[6] A. Mohammed, M. Ould-Khaoua, L.M. Mackenzie, C. Perkins, and J.D. Abdulai, "Probabilistic Counter-Based Route Discovery for Mobile Ad Hoc Networks," Proc. Int'l Conf. Wireless Comm. and Mobile Computing: Connecting the World Wirelessly (IWCMC '09), pp. 1335-1339, 2009.

[7] B. Williams and T. Camp, "Comparison of Broadcasting Techni-ques for Mobile Ad Hoc Networks," Proc. ACM MobiHoc, pp. 194-205, 2002.

[8] J. Kim, Q. Zhang, and D.P. Agrawal, "Probabilistic Broadcasting Based on Coverage Area and Neighbor Confirmation in Mobile Ad Hoc Networks," Proc. IEEE GlobeCom, 2004.

[9] J.D. Abdulai, M. Ould-Khaoua, and L.M. Mackenzie, "Improving Probabilistic Route Discovery in Mobile Ad Hoc Networks," Proc. IEEE Conf. Local Computer Networks, pp. 739-746, 2007.

[10] Z. Haas, J.Y. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing," Proc. IEEE INFOCOM, vol. 21,

pp. 1707-1716, 2002.

[11] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," Proc. ACM MobiHoc, pp. 129-130, 2000.

[12] J.D. Abdulai, M. Ould-Khaoua, L.M. Mackenzie, and A. Mo-hammed, "Neighbour Coverage: A Dynamic Probabilistic Route Discovery for Mobile Ad Hoc Networks," Proc. Int'l Symp. Performance Evaluation of Computer and Telecomm. Systems (SPECTS '08), pp. 165-172, 2008.

[13] J. Chen, Y.Z. Lee, H. Zhou, M. Gerla, and Y. Shu, "Robust Ad Hoc Routing for Lossy Wireless Environment," Proc. IEEE Conf. Military Comm. (MILCOM '06), pp. 1-7, 2006. \

[14] F. Stann, J. Heidemann, R. Shroff, and M.Z. Murtaza, "RBP: Robust Broadcast Propagation in Wireless Networks," Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys '06), pp. 85-98, 2006.

[15] F. Xue and P.R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," Wireless Networks, vol. 10, no. 2, pp. 169-181, 2004.