A Neighbor Coverage-Based Probabilistic Rebroadcast for Reducing Routing Overhead in Mobile Ad Hoc Networks

Xin Ming Zhang, *Member*, *IEEE*, En Bo Wang, Jing Jing Xia, and Dan Keun Sung, *Senior Member*, *IEEE*

Abstract—Due to high mobility of nodes in mobile ad hoc networks (MANETs), there exist frequent link breakages which lead to frequent path failures and route discoveries. The overhead of a route discovery cannot be neglected. In a route discovery, broadcasting is a fundamental and effective data dissemination mechanism, where a mobile node blindly rebroadcasts the first received route request packets unless it has a route to the destination, and thus it causes the broadcast storm problem. In this paper, we propose a neighbor coverage-based probabilistic rebroadcast protocol for reducing routing overhead in MANETs. In order to effectively exploit the neighbor coverage knowledge, we propose a novel rebroadcast delay to determine the rebroadcast order, and then we can obtain the more accurate additional coverage ratio by sensing neighbor coverage knowledge. We also define a connectivity factor to provide the node density adaptation. By combining the additional coverage ratio and connectivity factor, we set a reasonable rebroadcast probability. 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.

Index Terms—Mobile ad hoc networks, neighbor coverage, network connectivity, probabilistic rebroadcast, routing overhead

1 Introduction

Mobile ad hoc networks (MANETs) consist of a collection of mobile nodes which can move freely. These nodes 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.

- *X.M. Zhang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China, and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. E-mail: xinming@ustc.edu.cn.*
- *E.B. Wang and J.J. Xia are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, P.R. China. E-mail: {wangeb, jjxia}@mail.ustc.edu.cn.*
- *D.K. Sung is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea. E-mail: dksung@ee.kaist.ac.kr.*

Manuscript received 28 Apr. 2011; revised 19 Oct. 2011; accepted 16 Dec. 2011; published online 28 Dec. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-04-0221.
Digital Object Identifier no. 10.1109/TMC.2011.277.

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. Therefore, 1) in order to effectively exploit the neighbor coverage knowledge, we need a novel *rebroadcast delay* to determine the rebroadcast order, and then we can obtain a more accurate *additional coverage ratio*; 2) in order 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. After that, by combining the *additional coverage ratio* and the *connectivity factor*, we introduce a *rebroadcast probability*, which can be used to reduce the number of rebroadcasts of the RREQ packet, to improve the routing performance.

The main contributions of this paper are as follows:

1. 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.
2. We also propose a novel scheme to calculate the *rebroadcast probability*. The scheme considers the information about the uncovered neighbors (UCN), connectivity metric and local node density to calculate the rebroadcast probability. The rebroadcast probability is composed of two parts:

    a. *additional coverage ratio*, which is the ratio of the number of nodes that should be covered by a single broadcast to the total number of neighbors; and
    b. *connectivity factor*, which reflects the relationship of network connectivity and the number of neighbors of a given node.

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 is an effective mechanism for route discovery, but the routing overhead associated with the broadcasting can be quite large, especially in high dynamic networks [9]. 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 broadcasting 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]. Kim et al. [8] proposed a probabilistic broadcasting scheme based on coverage area and neighbor confirmation. This scheme uses the coverage area to set the rebroadcast probability, and uses the neighbor confirmation to guarantee reachability. Peng and Lu [11] proposed a neighbor knowledge scheme named Scalable Broadcast Algorithm (SBA). This scheme determines the rebroadcast of a packet according to the fact whether this rebroadcast would reach additional nodes. Abdulai et al. [12] proposed a Dynamic Probabilistic Route Discovery (DPR) scheme based on neighbor coverage. In this approach, each node determines the forwarding 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. Thus, there is a room of further optimization and extension for the DPR protocol.

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. [14] proposed two deterministic timer-based broadcast schemes: Dynamic Reflector Broadcast (DRB) and Dynamic Connector-Connector Broadcast (DCCB). They pointed out that their schemes can achieve full reachability over an idealistic lossless MAC layer, and for the situation of node failure and mobility, their schemes are robustness. Stann et al. [15] 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. They presented a new perspective for broadcasting: not to make a single broadcast more efficient but to make a single broadcast more reliable, which means by reducing the frequency of upper layer invoking flooding to improve the overall performance of flooding. 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 Rebroadcast 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:

$$U(n_i) = N(n_i) - [N(n_i) \cap N(s)] - \{s\}, \quad (1)$$

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 (1), 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:

$$\begin{aligned} T_p(n_i) &= 1 - \frac{|N(s) \cap N(n_i)|}{|N(s)|} \\ T_d(n_i) &= MaxDelay \times T_p(n_i), \end{aligned} \quad (2)$$

where $T_p(n_i)$ is the delay ratio of node n_i, and $MaxDelay$ is a small constant delay. $|\cdot|$ is the number of elements in a set.

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 $n_i, i = 1, 2, \ldots, |N(s)|$ receive and process the RREQ packet. We assume that node n_k has the largest number of common neighbors with node s, according to (2), 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. The objective of this rebroadcast delay is not to rebroadcast the RREQ packet to more nodes, but to disseminate the neighbor coverage knowledge more quickly. After determining the rebroadcast delay, the node can set its own timer.

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 from n_j. Then, the $U(n_i)$ can be adjusted as follows:

$$U(n_i) = U(n_i) - [U(n_i) \cap N(n_j)]. \quad (3)$$

After adjusting the $U(n_i)$, the RREQ packet received from n_j is discarded.

We do not need to adjust the rebroadcast delay because the rebroadcast delay is used to determine the order of disseminating neighbor coverage knowledge to the nodes which receive the same RREQ packet from the upstream node. Thus, it is determined by the neighbors of upstream nodes and its own.

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* ($R_a(n_i)$) of node n_i as

$$R_a(n_i) = \frac{|U(n_i)|}{|N(n_i)|}. \quad (4)$$

This metric indicates the ratio of the number of nodes that are additionally covered by this rebroadcast to the total number of neighbors of node n_i. The nodes that are additionally covered need to receive and process the RREQ packet. 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.

Xue and Kumar [16] derived that if each node connects to more than $5.1774 \log n$ of its nearest neighbors, then the probability of the network being connected is approaching 1 as n increases, where n is the number of nodes in the network. Then, we can use $5.1774 \log n$ as the *connectivity metric* of the network. 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: $|N(n_i)| \cdot F_c(n_i) \geq 5.1774 \log n$. Then, we define the minimum $F_c(n_i)$ as a *connectivity factor*, which is

$$F_c(n_i) = \frac{N_c}{|N(n_i)|}, \quad (5)$$

where $N_c = 5.1774 \log n$, and n is the number of nodes in the network.

From (5), 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. And when $|N(n_i)|$ is less than N_c, $F_c(n_i)$ is greater than 1. That means node n_i is in the sparse area of the network, then node n_i should forward the RREQ packet in order to approach network connectivity.

Combining the *additional coverage ratio* and *connectivity factor*, we obtain the *rebroadcast probability* $P_{re}(n_i)$ of node n_i:

$$P_{re}(n_i) = F_c(n_i) \cdot R_a(n_i), \quad (6)$$

where, if the $P_{re}(n_i)$ is greater than 1, we set the $P_{re}(n_i)$ 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.

Note that the calculated rebroadcast probability $P_{re}(n_i)$ may be greater than 1, but it does not impact the behavior of the protocol. It just shows that the local density of the node is so low that the node must forward the RREQ packet. Then, node n_i need to rebroadcast the RREQ packet received from s with probability $P_{re}(n_i)$.

3.3 Algorithm Description

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

Algorithm 1. 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* $T_d(n_i)$:}
  $T_p(n_i) = 1 - \frac{|N(s) \cap N(n_i)|}{|N(s)|}$
  $T_d(n_i) = MaxDelay \times T_p(n_i)$
  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 $U(n_i, R_s.id)$:}
  $U(n_i, R_s.id) = U(n_i, R_s.id) - [U(n_i, R_s.id) \cap N(n_j)]$
  discard($RREQ_j$)
end while

if Timer($n_i, R_s.id$) expires **then**
  {Compute the *rebroadcast probability* $P_{re}(n_i)$:}
  $R_a(n_i) = \frac{|U(n_i, R_s.id)|}{|N(n_i)|}$
  $F_c(n_i) = \frac{N_c}{|N(n_i)|}$
  $P_{re}(n_i) = F_c(n_i) \cdot R_a(n_i)$
  if Random(0,1) $\leq P_{re}(n_i)$ **then**
    broadcast($RREQ_s$)
  else
    discard($RREQ_s$)
  end if
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 [17] 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 *HelloInterval*, the node needs to send a *Hello* packet. The value of *HelloInterval* 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_j, then node n_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_j 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.

The nodes which receive the RREQ packet from node n_i can take their actions according to the value of *num_neighbors* in the received RREQ packet:

- if the *num_neighbors* is a positive integer, the node substitutes its neighbor cache of node n_i according to the neighbor list in the received RREQ packet;
- if the *num_neighbors* is a negative integer, the node updates its neighbor cache of node n_i and deletes the deleted neighbors in the received RREQ packet;
- if the *num_neighbors* is 0, the node does nothing.

Because of the two cases 2 and 3, this technique can reduce the overhead of neighbor list listed in the RREQ packet.

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. Broadcasting is a fundamental and effective data dissemination mechanism for many applications in MANETs. 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, we choose the Dynamic Probabilistic Route Discovery [12] protocol which is an optimization scheme for reducing the overhead of RREQ packet incurred in route discovery in the recent literature, and the conventional AODV protocol.

Simulation parameters are as follows: The Distributed Coordination Function (DCF) of the IEEE 802.11 protocol is used as the MAC layer protocol. The radio channel model follows a Lucent's WaveLAN with a bit rate of 2 Mbps, and the transmission range is 250 meters. We consider constant bit rate (CBR) data traffic and randomly choose different source-destination connections. Every source sends four CBR packets whose size is 512 bytes per second. The mobility model is based on the random waypoint model in a field of $1{,}000 \text{ m} \times 1{,}000 \text{ m}$. In this mobility model, each node moves to a random selected destination with a random speed from a uniform distribution [1, max-speed]. After the node reaches its destination, it stops for a pause-time interval and chooses a new destination and speed. In order to reflect the network mobility, we set the max-speed to 5 m/s and set the pause time to 0. The $MaxDelay$ used to determine the rebroadcast delay is set to 0.01 s, which is equal to the upper limit of the random jitter time of sending broadcast packets in the default implementation of AODV in NS-2. Thus, it could not induce extra delay in the route discovery. The simulation time for each simulation scenario is set to 300 seconds. In the results, each data point represents the average of 30 trials of experiments. The confidence level is 95 percent, and the confidence interval is shown as a vertical bar in the figures. The detailed simulation parameters are shown in Table 1.

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.

TABLE 1
Simulation Parameters

Simulation Parameter	Value
Simulator	NS-2 (v2.30)
Topology Size	1000m $\times$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

- **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.

The experiments are divided to three parts, and in each part we evaluate the impact of one of the following parameters on the performance of routing protocols:

- **Number of nodes**. We vary the number of nodes from 50 to 300 in a fixed field to evaluate the impact of different network density. In this part, we set the number of CBR connections to 15, and do not introduce extra packet loss.
- **Number of CBR connections**. We vary the number of randomly chosen CBR connections from 10 to 20 with a fixed packet rate to evaluate the impact of different traffic load. In this part, we set the number of nodes to 150, and also do not introduce extra packet loss.
- **Random packet loss rate**. We use the Error Model provided in the NS-2 simulator to introduce packet loss to evaluate the impact of random packet loss. The packet loss rate is uniformly distributed, whose range is from 0 to 0.1. In this part, we set the number of nodes to 150 and set the number of connections to 15.

In the experiments analysis, when two protocols are compared, we use the following method to calculate the average: we assume that the varied parameter is $(x_1, x_2, \ldots, x_n)$, the performance metric of protocol 1 is $(y_1, y_2, \ldots, y_n)$,

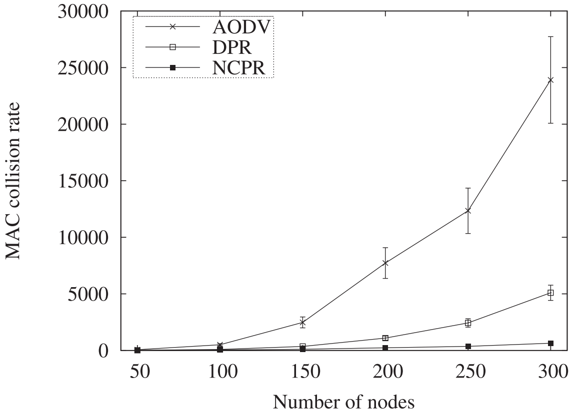

Fig. 1. MAC collision rate with varied number of nodes.

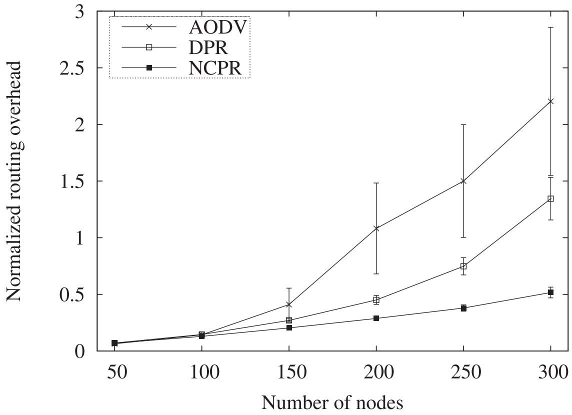

Fig. 2. Normalized routing overhead with varied number of nodes.

and the performance metric of protocol 2 is $(z_1, z_2, \ldots, z_n)$. When protocol 1 compares to protocol 2, the average is defined as: $[(y_1 - z_1)/z_1 + (y_2 - z_2)/z_2 + \cdots + (y_n - z_n)/z_n]/n * 100\%$.

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. The packet drops in MAC layer not only affect the number of retransmissions in MAC layer, but also affect the packet delivery ratio of CBR packets in the application layer. It is very important to reduce the redundant rebroadcast and packet drops caused by collisions to improve the routing performance. Compared with the conventional AODV protocol, the NCPR protocol reduces the MAC collision rate by about 92.8 percent on the average. Under the same network conditions, the MAC collision rate is reduced by about 61.6 percent when the NCPR protocol is compared with the DPR protocol. This is the main reason that the NCPR protocol could improve the routing performance.

Fig. 2 shows the normalized routing overhead with different network density. The NCPR protocol can significantly reduce the routing overhead incurred during the route discovery, especially in dense network. Although the NCPR protocol increases the packet size of RREQ packets, it reduces the number of RREQ packets more significantly. Then, the RREQ traffic is still reduced. In addition, for fairness, the statistics of normalized routing overhead includes *Hello* traffic. Even so, the NCPR protocol still yields the best performance, so that the improvement of normalized routing overhead is considerable. On average, the overhead is reduced by about 45.9 percent in the NCPR protocol compared with the conventional AODV protocol. Under the same network conditions, the overhead is reduced by about 30.8 percent when the NCPR protocol is compared with the DPR protocol. When network is dense, the NCPR protocol reduces overhead by about 74.9 and 49.1 percent when compared with the AODV and DPR protocols, respectively. This result indicates that the NCPR protocol is the most efficient among the three protocols.

Fig. 3 shows the packet delivery ratio with increasing network density. The NCPR protocol can increase the packet delivery ratio because it significantly reduces the number of collisions, which is shown in Fig. 1, so that it reduces the number of packet drops caused by collisions. On average, the packet delivery ratio is improved by about 11.9 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 3.7 percent when compared with the DPR protocol. When network is dense, the NCPR protocol increases the packet delivery ratio about 21.8 and 6.3 percent when compared with the AODV and DPR protocols, respectively.

Fig. 4 measures the average end-to-end delay of CBR packets received at the destinations with increasing network density. The NCPR protocol decreases the average end-to-end delay due to a decrease in the number of redundant rebroadcasting packets. The redundant rebroadcast increases delay because 1) it incurs too many collisions and interference, which not only leads to excessive packet drops, but also increases the number of retransmissions in MAC layer so as to increase the delay; 2) it incurs too many channel contentions, which increases the backoff timer in MAC layer, so as to increase the delay. Thus, reducing the redundant rebroadcast can decrease the delay. On average, the end-to-end delay is reduced by about 60.8 percent in the NCPR protocol when compared with the conventional AODV protocol. Under the same network conditions, the

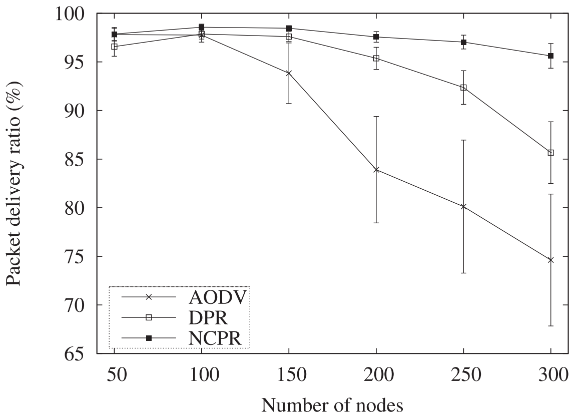

Fig. 3. Packet delivery ratio with varied number of nodes.

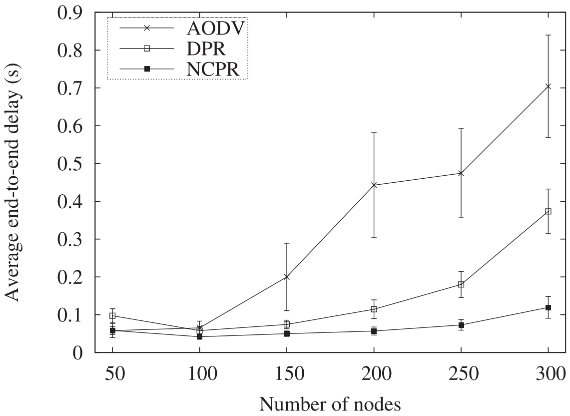

Fig. 4. Average end-to-end delay with varied number of nodes.

delay is reduced by about 46.3 percent when the NCPR protocol is compared with the DPR protocol. When network is dense, the NCPR protocol reduces the average end-to-end delay by about 84.9 and 59.2 percent when compared with the AODV and DPR protocols, respectively. The NCPR protocol uses a rebroadcast delay based on coverage ratio to replace the random delay in the AODV protocol, and the *MaxDelay* in the NCPR protocol is equal to the upper limit random delay in the AODV protocol, so the NCPR protocol does not cause extra delay cost.

4.4 Performance with Varied Number of CBR Connections

Fig. 5 shows the effects of the traffic load on the MAC collision rate. Since the data and control packets share the same physical channel in the IEEE 802.11 protocol, as the number of CBR connections increases, the physical channel will be busier and then the collision of the MAC layer will be more severe. Both the DPR and NCPR protocols do not consider load balance, but they can reduce the redundant rebroadcast and alleviate the channel congestion, so as to reduce the packet drops caused by collisions. Compared with the conventional AODV protocol, the NCPR protocol reduces the MAC collision rate by about 95.2 percent on the average. As shown in Fig. 1, the NCPR protocol reduces more MAC collision rate than the DPR protocol as network density increases. But in the same node density and different traffic load, the NCPR reduce nearly the same

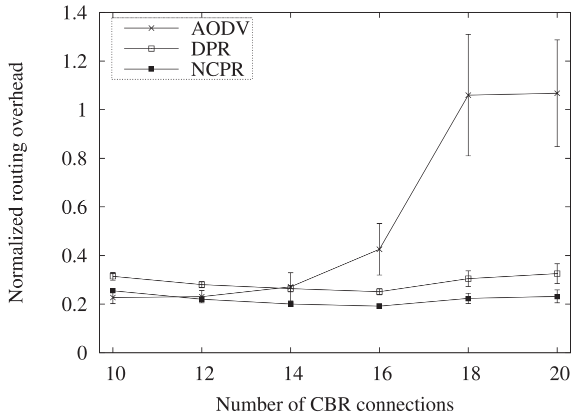

Fig. 6. Normalized routing overhead with varied number of CBR connections.

scale of MAC collision rate than the DPR protocol, that is the MAC collision rate is reduced by about 69.2 percent. Therefore, at different traffic load, the NCPR protocol also can improve the routing performance.

Fig. 6 shows the normalized routing overhead with different traffic load. At very light traffic load (10 CBR connections), both the DPR and NCPR protocols have more routing overhead than the conventional AODV protocol. This is because that the *Hello* packets and neighbor list in the RREQ packet add extra overhead, and the effect of reducing redundant rebroadcast is not significant when traffic load is light. As the traffic load increases, the routing overhead of the conventional AODV protocol significantly increases, but the overhead of the DPR and NCPR protocols are relatively smooth. By contrast, both the DPR and NCPR protocols reduce the routing overhead. On average, the overhead is reduced by about 38.4 percent in the NCPR protocol compared with the conventional AODV protocol. Under the same network conditions, the overhead is reduced by about 23.9 percent when the NCPR protocol is compared with the DPR protocol.

Fig. 7 shows the packet delivery ratio with increasing traffic load. As the traffic load increases, the packet drops of the conventional AODV protocol without any optimization for redundant rebroadcast are more severe. Both the DPR and NCPR protocols increase the packet delivery ratio compared to the conventional AODV protocol, because

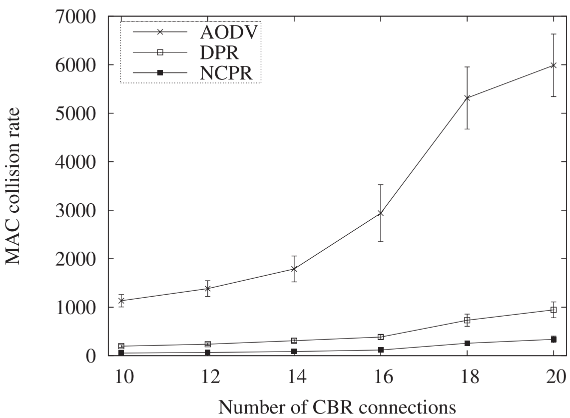

Fig. 5. MAC collision rate with varied number of CBR connections.

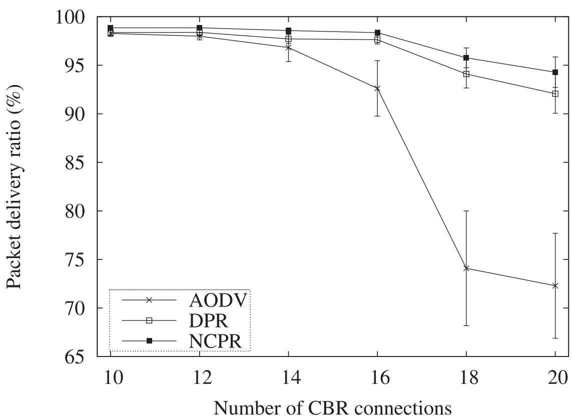

Fig. 7. Packet delivery ratio with varied number of CBR connections.

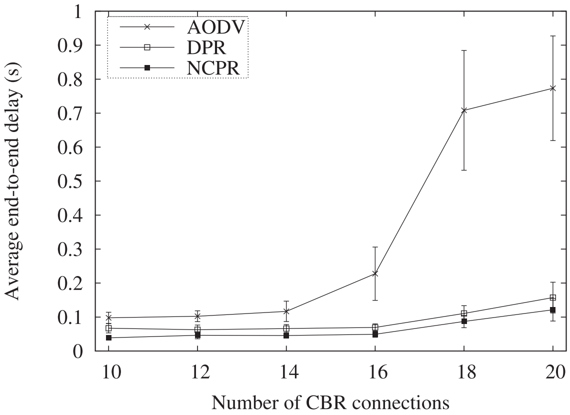

Fig. 8. Average end-to-end delay with varied number of CBR connections.

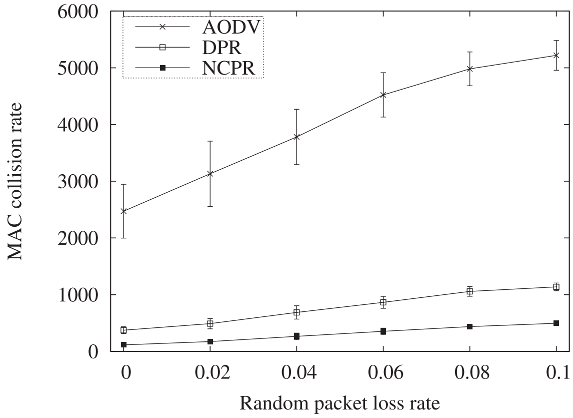

Fig. 9. MAC collision rate with varied random packet loss rate.

both of them significantly reduce the number of collisions and then reduce the number of packet drops caused by collisions. On average, the packet delivery ratio is improved by about 11.5 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 1.1 percent when compared with the DPR protocol.

Fig. 8 measures the average end-to-end delay of CBR packets received at the destinations with increasing traffic load. The end-to-end delay of the conventional AODV protocol significantly increases with the increase of traffic load, which is the same as the MAC collision rate and routing overhead. When the traffic load is heavy, by reducing the redundant rebroadcast, both the DPR and NCPR protocols alleviate the channel congestion and reduce the retransmissions at MAC layer, thus, both of them reduce the end-to-end delay. On average, the end-to-end delay is reduced by about 71.0 percent in the NCPR protocol when compared with the conventional AODV protocol. Under the same network conditions, the delay is reduced by about 28.7 percent when the NCPR protocol is compared with the DPR protocol.

4.5 Performance with Varied Random Packet Loss Rate

Fig. 9 shows the effects of the packet loss rate on the MAC collision rate. In our simulation parameters, we use both the *IncomingErrrProc* and *OutgoingErrProc* 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. Both 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. In the same network density and traffic load but in different packet loss rate, the MAC collision rate is reduced by about 61.6 percent when the NCPR protocol is compared with the DPR protocol.

Fig. 10 shows the normalized routing overhead with different packet loss rate. As the packet loss increases, there will be more link breakages and route discoveries, and then there will be more routing overhead (such as RREQ packets and RERR packets). On the other hand, the CBR connection using UDP protocol does not have any retransmissions mechanism; thus, the CBR connections will drop more packets as packet loss rate increases. By reducing redundant rebroadcast of RREQ packets, both the DPR and NCPR protocols incur less routing overhead than the conventional AODV protocol. On average, the overhead is reduced by about 59.4 percent in the NCPR protocol compared with the conventional AODV protocol. Under the same network conditions, the overhead is reduced by about 22.3 percent when the NCPR protocol is compared with the DPR protocol.

Fig. 11 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. 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

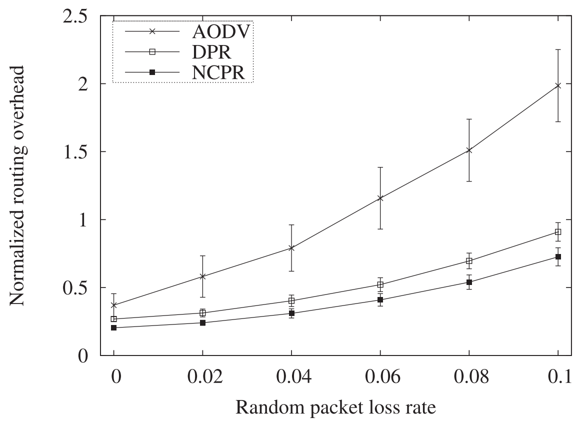

Fig. 10. Normalized routing overhead with varied random packet loss rate.

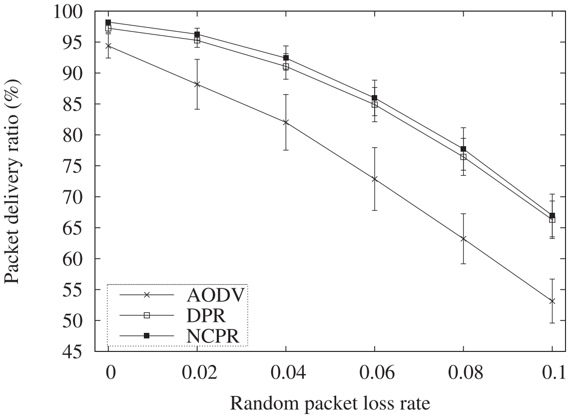

Fig. 11. Packet delivery ratio with varied random packet loss rate.

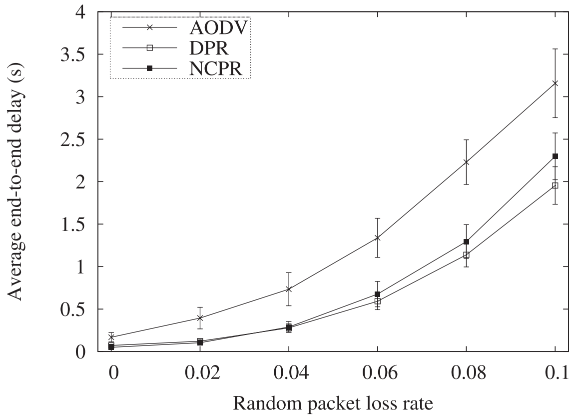

Fig. 12. Average end-to-end delay with varied random packet loss rate.

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.

Fig. 12 measures the average end-to-end delay of CBR packets received at the destinations with increasing packet loss rate. Due to the increase of packet loss, the retransmissions caused by random packet loss at MAC layer will increase so as to increase the end-to-end delay. Both the DPR and NCPR protocols alleviate the channel congestion and reduce the retransmissions caused by collision at MAC layer, thus, both of them have a lower end-to-end delay than the conventional AODV protocol. On average, the end-to-end delay is reduced by about 53.9 percent in the NCPR protocol when compared with the conventional AODV protocol. Under the same network conditions, the delay is increased by about 0.6 percent when the NCPR protocol is compared with the DPR protocol.

5 Conclusion

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 forwarding order and more effectively exploit the neighbor coverage 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.

Acknowledgments

The authors would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grants 61073185, 60673171, and 61232018 and by the open research fund of National Mobile Communications Research Laboratory, Southeast University (No. 2010D15).

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, "Routing 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 Techniques 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. Mohammed, "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] A. Keshavarz-Haddady, V. Ribeirox, and R. Riedi, "DRB and DCCB: Efficient and Robust Dynamic Broadcast for Ad Hoc and Sensor Networks," *Proc. IEEE Comm. Soc. Conf. Sensor, Mesh, and Ad Hoc Comm. and Networks (SECON '07),* pp. 253-262, 2007.

[15] 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.
[16] 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.
[17] X.M. Zhang, E.B. Wang, J.J. Xia, and D.K. Sung, "An Estimated Distance Based Routing Protocol for Mobile Ad Hoc Networks," *IEEE Trans. Vehicular Technology,* vol. 60, no. 7, pp. 3473-3484, Sept. 2011.

Xin Ming Zhang received the BE and ME degrees in electrical engineering from the China University of Mining and Technology, Xuzhou, in 1985 and 1988, respectively, and the PhD degree in computer science and technology from the University of Science and Technology of China, Hefei, in 2001. Since 2002, he has been with the faculty of the University of Science and Technology of China, where he is currently an associate professor in the School of Computer Science and Technology. He is also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. From September 2005 to August 2006, he was a visiting professor in the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon. His research interest is wireless networks. He has published more than 60 papers in wireless ad hoc and sensor networks. He is a member of the IEEE.

En Bo Wang received the BS degree in computer science and technology from Jilin University, Changchun, China, in 2007 and the ME degree in computer science and technology from the University of Science and Technology of China, Hefei, in 2010. His research interests include wireless networks.

Jing Jing Xia received the BE degree in computer science and technology from the Heifei University of Technology, China, in 2008. She is currently working toward her master's degree in the School of Computer Science and Technology, University of Science and Technology of China, Hefei. Her research interests include wireless networks.

Dan Keun Sung received the BS degree in electronics engineering from Seoul National University, Korea, in 1975 and the MS and PhD degrees in electrical and computer engineering from the University of Texas at Austin in 1982 and 1986, respectively. Since 1986, he has been with the faculty of the Korea Advanced Institute of Science and Technology (KAIST), Daejon, where he is currently a professor in the Department of Electrical Engineering. From 1996 to 1999, he was the director of the Satellite Technology Research Center, KAIST. He was the division editor of the *Journal of Communications and Networks*. His research interests include mobile communication systems and networks, with special interest in resource management, machine-to-machine communications, smart grid networks, wireless local area networks, wireless personal area networks, high-speed networks, next-generation internet protocol-based networks, traffic control in wireless and wireline networks, signaling networks, intelligent networks, performance and reliability of communication systems, and microsatellites. He is a member of the National Academy of Engineering of Korea. He is the editor of the *IEEE Communications Magazine*. He was the recipient of the 1992 National Order of Merits, the Dongbaek Medal, the 1997 Research Achievement Award, the 1997 MoMuc Paper Award, the 2000 Academic Excellence Award, the Best Paper Award at the 2000 Asia-Pacific Conference on Communications, the 2004 This Month's Scientist Award from the Ministry of Science and Technology and the Korea Science and Engineering Foundation, and the Best Paper Award at the 2005 Next-Generation PC International Conference. He is a senior member of the IEEE.