IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 6 (Jul. - Aug. 2013), PP 01-09
www.iosrjournals.org

A Novel Rebroadcast Technique for Reducing Routing Overhead In Mobile Ad Hoc Networks

Vinayak T. Patil [1], Asst.Prof. Padma. S. Dandannavar [2]
Department of Computer Science and EngineeringKLS Gogte Institute of Technology, Belgaum-59008, INDIA.

Abstract- *In mobile ad hoc networks (MANETs), the network topology changes frequently and unpredictably due to the arbitrary mobility of nodes. This feature leads to frequent path failures and route reconstructions, which causes an increase in the routing control overhead. The overhead of a route discovery cannot be neglected. Thus, it is imperative to reduce the overhead of route discovery in the design of routing protocols of MANETs. One of the fundamental challenges of MANETs is the design of dynamic routing protocols with good performance and less overhead. 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. This paper focuses on a probabilistic rebroadcast protocol based on neighbor coverage to reduce the routing overhead in MANETs.*
Keywords *- Mobile Ad Hoc Networks, Neighbor Coverage, and Network Connectivity, Probabilistic Rebroadcast, Routing Overhead, AODV.*

I. Introduction

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 [1, 5]. In our implementation some broadcasting techniques are used to reduce the overhead of Hello packets and neighbor list in the RREQ packet. 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. 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.

II. Literature Review

The routing overhead occurred because of the dissemination of routing control packets such as RREQ packets can be quite huge, especially when the network topology frequently changes. Traditional on-demand routing protocols produce a large amount of routing traffic by blindly flooding the entire network with RREQ packets during route discovery. Recently, the issue of reducing the routing overhead associated with route discovery and maintenance in on demand routing protocols has attracted increasing attention.

Huang [2] proposed a methodology of dynamically adjusting the Hello timer and the Timeout timer according to the conditions of the network. For example, in a high mobility network (with frequent topology changes) it is desirable to use small values for the timers to quickly detect the changes in the network. On the other hand, in a low mobility network where the topology remains stable and with few changes, a large value for the timers is more effective to reduce the overhead. In order to decide whether the mobility of the network is high or low, we use a simple way to approximate in real time of the link change rate. The reduction of the overhead is greatly achieved with the minimal cost of slightly increasing the drop rate in data traffic. While the packet loss increases around 1%, the overhead reduction reaches 40%.

Ould-Khaoua[4]proposed two new probabilistic route discovery method, called Adjusted Probabilistic route discovery(AP) and Enhance Adjusted Probabilistic route discovery (EAP) which addresses the broadcast storm problem in the existing on-demand routing protocols. The forwarding probability is determined by taking into account about the local density of the sending node. In order to reduce the routing overhead without degrading the network throughput in dense networks, the forwarding probability of nodes located in sparse areas is set high while it is set low at nodes located in dense areas. EAP-AODV reduces overhead by 71% while APAODV reduces the overhead by 55%.

Aminu[6] proposed a rebroadcast probability function which takes in to account about the value of the packet counter together with some key simulation parameters(i.e. network topology size, transmission range and number of nodes) to determine the appropriate rebroadcast probability for a given node. The rebroadcast probability of a node is computed based on these parameters. Compared to the other schemes, simulation results have revealed that counter Function achieved superior saved rebroadcast (about 20% better than its closest competitor i.e., counter-based scheme, in dense network) and end-to-end delay (around 26% better than counter-based scheme in dense network) without sacrificing reach ability in medium and dense networks.

III. Neighbor Coverage Based Probabilistic Rebroadcast (NCPR) Protocol

This paper proposes neighbor coverage based probabilistic rebroadcast protocol [1] which combines both neighbor coverage and probabilistic methods. 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. In order to keep the network connectivity and to reduce the redundant retransmissions, we need a metric named connectivity factor to determine how many neighbors should receive the RREQ packet [9]. After that, by combining the additional coverage ratio and the connectivity factor, we introduce rebroadcast probability, which can be used to reduce the number of rebroadcasts of the RREQ packet and to improve the routing performance.

3.1 Rebroadcast Delay

We proposed a 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 [10]. Therefore, this rebroadcast delay enables the information about the nodes which have transmitted the packet to more neighbors, which is the key success for the proposed scheme.

When a node ni receives an RREQ packet from its previous node s, node s can use the neighbor list in the RREQ packet to estimate how many its neighbors have not been covered by the RREQ packet . If node ni has more neighbors uncovered by the RREQ packet from s, which means that if node ni rebroadcasts the RREQ packet, the RREQ packet can reach more additional neighbor nodes.

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 ni, i = 1, 2 …receive and process the RREQ packet. We assume that node nk has the largest number of common neighbors with node s, node nk has the lowest delay. Once node nk rebroadcasts the RREQ packet, there are more nodes to receive the RREQ, because node nk has the largest number of common neighbors. Node nk rebroadcasts the RREQ packet depends on its rebroadcast probability calculated in the next subsection. 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 Rebroadcast Probability

We also proposed a novel scheme to calculate the rebroadcast probability. The scheme considers the information about the uncovered neighbors, 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 node which has a larger rebroadcast delay may listen to RREQ packets from the nodes which have lowered one [9]. We do not need to adjust the rebroadcast delay because the rebroadcast delay is used to determine the order of disseminating neighbor coverage knowledge. When the timer of the rebroadcast delay of node ni expires, the node obtains the final uncovered neighbor set. The nodes belonging to the final uncovered neighbor 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 uncovered neighbor set is not changed, which is the initial uncovered neighbor set. Now we study how to use the final uncovered neighbor set to set the rebroadcast probability. The metric Ra indicates the ratio of the number of nodes that are additionally covered by this rebroadcast to the total number of neighbors of node ni. The nodes that are additionally covered need to receive and process the RREQ packet. As Ra 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 [7] 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 ni is Fc(ni).If

the local node density is low, the parameter Fc 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 Fc could further decrease the rebroadcast probability, and then further increases the efficiency of NCPR in the dense area. Thus, the parameter Fc adds density adaptation to the rebroadcast probability.

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.

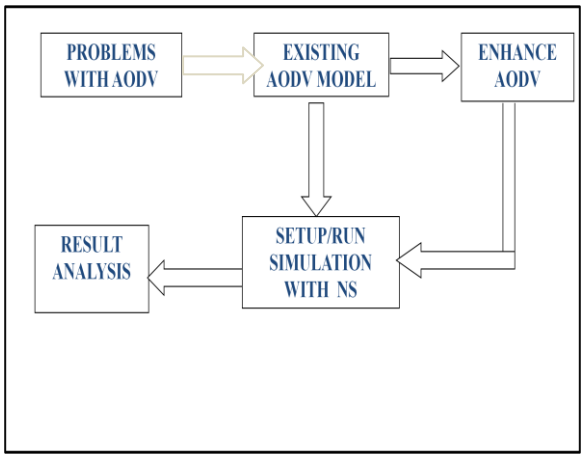

Fig. 3.1 Flow Diagram Of Protocols

Fig 3.1shows the flow diagram of protocols in this model enhancing of AODV protocol at Mac layer will be done.

Algorithm

The formal description of the Neighbor Coverage based Probabilistic Rebroadcast (NCPR) for reducing routing overhead in route discovery is shown in algorithm [12].

Definitions:

RREQv: RREQ packet received from node v.

Rv.id: the unique identifier (id) of RREQv.

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.}

1: if ni receives a new RREQs from s then
2: {Compute initial uncovered neighbors set U(ni,Rs.id) for RREQs:}
3: $U(ni,Rs.id) = N(ni) - [N(ni) \cap N(s)] - \{s\}$
4: {Compute the rebroadcast delay Td(ni):}
5: $Tp(ni) = 1 - \frac{|N(s) \cap N(ni)|}{|N(s)|}$
6: $Td(ni) = MaxDelay \times Tp(ni)$
7: Set a Timer(ni,Rs.id) according to Td(ni)
8: end if 9:
10: while ni receives a duplicate RREQj from nj before Timer(ni,Rs.id) expires do
11: {Adjust U(ni,Rs.id):}
12: $U(ni,Rs.id)=(ni,Rs.id)-[U(ni,Rs.id) \cap N(nj)]$
13: discard(RREQj);
14: end while
15:
16: if Timer(ni,Rs.id) expires then
17: {Compute the rebroadcast probability Pre(ni):}
18: $Ra(ni) = |U(ni,Rs.id)|$

|N(ni)|
19: Fc(ni) = Nc
|N(ni)|
20: Pre(ni) = Fc(ni) · Ra(ni)
21: if Random(0,1) ≤ Pre(ni) then
22: broadcast(RREQs)
23: else
24: discard(RREQs)
25: end if
26: end if

IV. Protocol Implementation and Performance Evaluation

4.1 Protocol Implementation

We enhance the source code of AODV at MAC in NS-2 to implement our proposed protocol. 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. 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. For sending or forwarding of RREQ packets, the neighbor table of any node ni has the following 3 cases:

1)If the neighbor table of node ni adds at least one new neighbor nj , then node ni 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.

2) If the neighbor table of node ni deletes some neighbors, then node ni 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;

3) If the neighbor table of node ni does not vary, node ni does not need to list its neighbors, and set the num neighbors to 0.The nodes which receive the RREQ packet from node ni can take their actions according to the value of num neighbors in the received RREQ packet:

1) If the num neighbors is a positive integer, the node substitutes its neighbor cache of node ni according to the neighbor list in the received RREQ packet;

2) If the num neighbors is a negative integer, the node updates its neighbor cache of node ni and deletes the deleted neighbors in the received RREQ packet;

3) If the num neighbor 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.

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

- **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 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.

4.2 Simulation parameters

Simulation parameters and scenarios which are used to investigate the performance of the proposed protocol.

Table 4.1

Simulation Parameter	Value
Simulator	NS-2 (v2.31)
Topology Size	1000m × 1000m
Number of Nodes	10,20,30
Transmission Range	250m
Bandwidth	2Mbps
Interface Queue Length	40
Traffic Type	CBR
Number of CBR Connections	10, 12, ..., 15,
Packet Size	512 bytes
Packet Rate	4 packets/sec
Min Speed	1 m/s
Max Speed	5 m/s

V. Results

1. Performance with Varied Number of Nodes

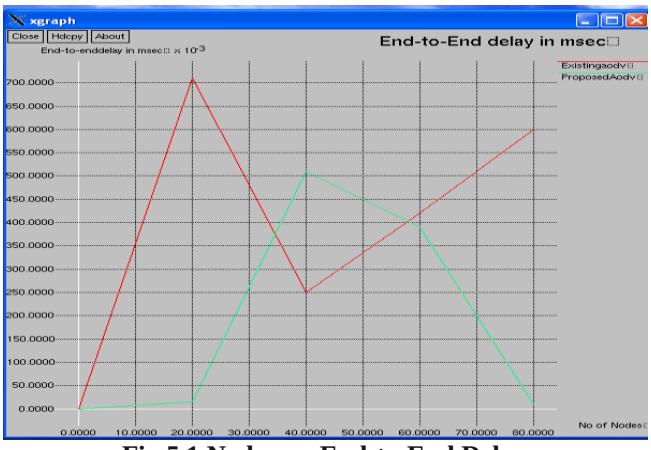

Fig.5.1 Nodes vs. End-to-End Delay

Fig. 5.1 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 back off timer in MAC layer, so as to increase the delay.

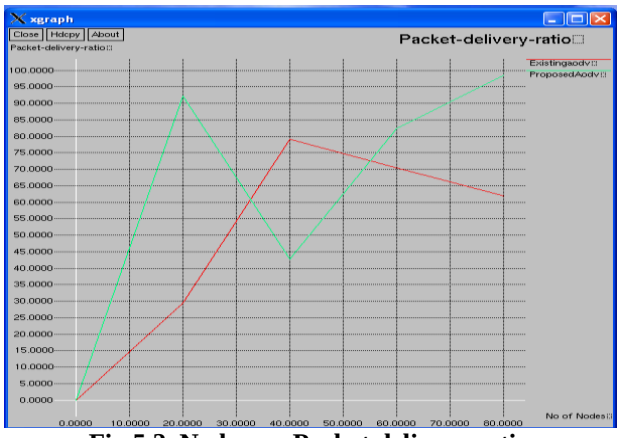

Fig.5.2. Nodes vs. Packet delivery ratio

Fig. 5.2 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, so that it reduces the number of packet drops caused by collisions.

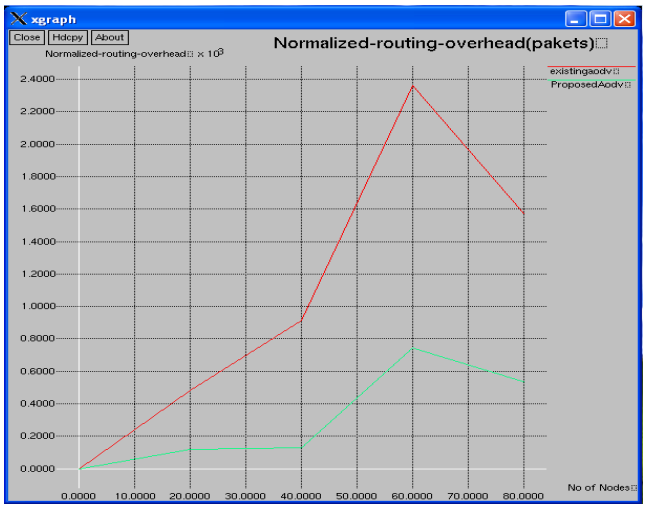

Fig.5.3 Nodes vs. Routing Overhead

Fig. 5.3 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.

2. Performance with Varied Number of CBR Connections

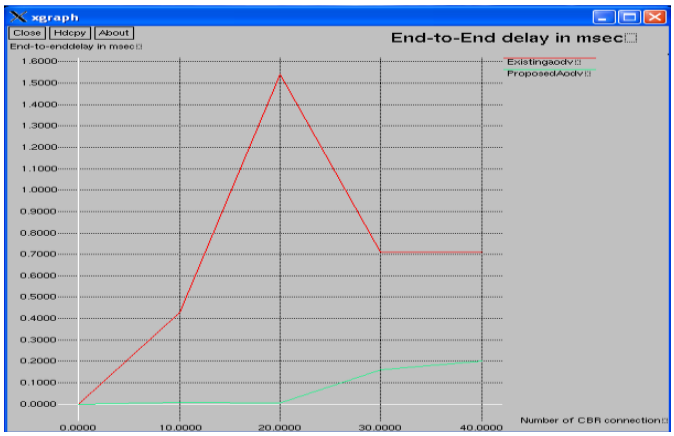

Fig.5.4 Number of CBR connection vs. End-to-End Delay

Fig. 5.4 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, NCPR protocols alleviate the channel congestion and reduce the retransmissions at MAC layer, thus, to reduce the end-to-end delay.

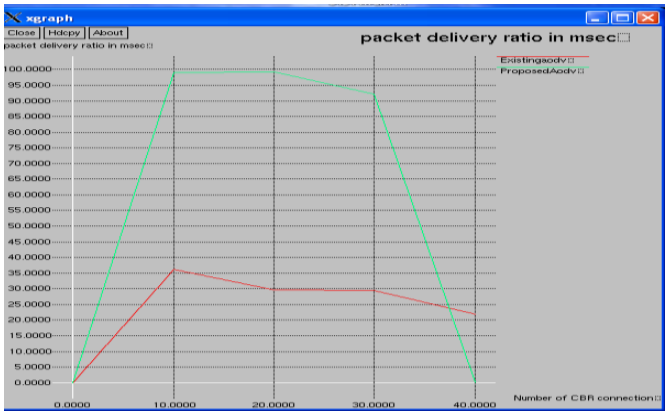

Fig.5.5 Number of CBR connection vs. Packet delivery ratio

Fig. 5.5 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. NCPR protocols increase the packet delivery ratio compared to the conventional AODV protocol, because of the significantly reduce the number of collisions and then reduce the number of packet drops caused by collisions.

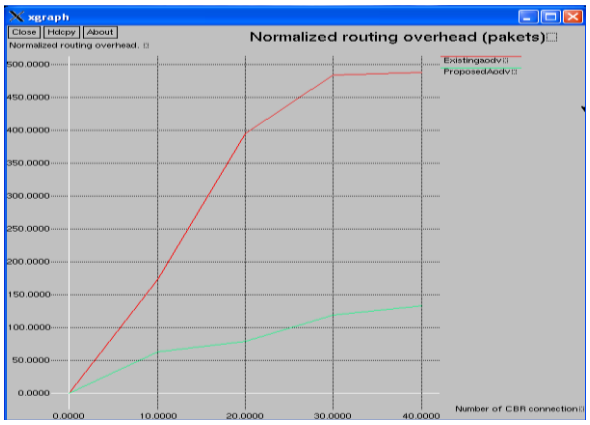

Fig.5.6 Number of CBR connection vs. Routing overhead

Fig. 5.6 shows the normalized routing overhead with different traffic load. At very light traffic load (10 CBR connections), 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 D NCPR protocols is relatively smooth. By contrast, NCPR protocols reduce the routing overhead.

3. Performance with Varied Random Packet Loss Rate

Fig. 5.7 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. NCPR protocols alleviate the channel congestion and reduce the retransmissions caused by collision at MAC layer, thus, to have a lower end-to-end delay than the conventional AODV protocol.

Fig.5.7 Random packet loss vs. End-to-End Delay

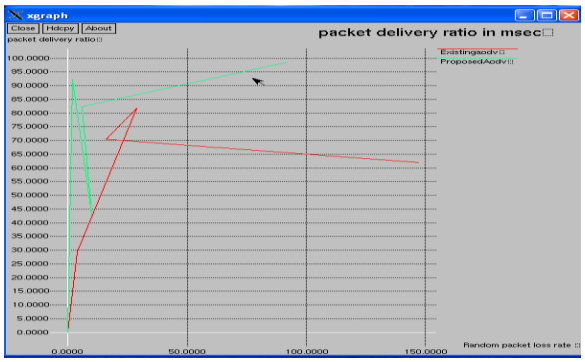

Fig.5.8 Random packet loss vs. Packet delivery ratio

Fig. 5.8 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. NCPR protocols do not exploit any robustness mechanism for packet loss, but it can reduce the redundant rebroadcast, so as to reduce the packet drops caused by collision.

Fig. 5.9 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.

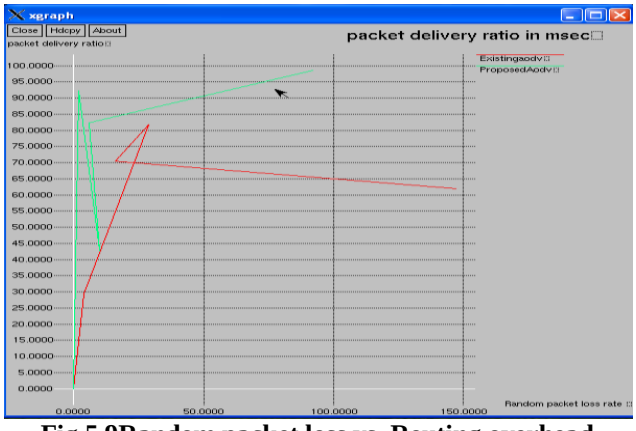

Fig.5.9Random packet loss vs. Routing overhead

By reducing redundant rebroadcast of RREQ packets, both NCPR protocols incur less routing overhead than the conventional AODV protocol.

I. Conclusion

In this paper we proposed a Novel Rebroadcast Technique for Reducing Routing Overhead in Mobile Ad Hoc Networks. 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 Existing protocol. 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.

In future, we can calculate the result for another performance matrix i.e. MAC collision rate. The NCPR algorithm can be apply to DSR and results comparison can be done with AODV protocol.

References

[1] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, 2003.
[2] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR) for IPv4," RFC 4728, 2007.
[3] H. AlAamri, M. Abolhasan, and T. Wysocki, "On Optimising Route Discovery in Absence of Previous Route Information in MANETs," Proc. of IEEE VTC 2009-Spring, 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. of IEEE 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. of ACM/IEEE MobiCom'99, pp. 151-162, 1999.
[6] Mohammed, M. Ould-Khaoua, L.M. Mackenzie, C. Perkins, and J. D. Abdulai, "Probabilistic Counter-Based Route Discovery for Mobile Ad Hoc Networks," Proc. of WCMC'09, pp. 1335-1339, 2009.
[7] Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," Proc. ACM MobiHoc'02, 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. of IEEE GLOBECOM'04, 2004.
[9] J. D. Abdulai, M. Ould-Khaoua, and L. M. Mackenzie, "Improving Probabilistic Route Discovery in Mobile Ad Hoc Networks," Proc. Of IEEE Conference on Local Computer Networks, pp. 739-746, 2007.
[10] Network Simulator - ns-2 http://www.isi.edu/nsnam/ns/.
[11] 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. of SPECTS'08, pp. 165-172, 2008
[12] F. Xue and P. R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," Wireless Networks, vol. 10, issue 2, pp. 169-181, 2004.