http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 4 (2): 176-183, 2005
ISSN 1812-5638

Multicast Routing for Mobile Ad Hoc Network Using Diversity Coding

M.A. Bhagyaveni and S. Shanmugavel
Telematics Laboratory, Department of Electronics and Communication Engineering,
College of Engineering, Anna University, Chennai-600 025, India

Abstract: In this study we propose Coded route diversity technique to improve the packet delivery ratio for MANET. This algorithm uses multiple routes to transmit the data and code packet to all the members of the multicasting group. It is found that the proposed algorithm is most reliable in the sense that it uses only the best neighbors to transmit the data and also that it posses self healing nature that can overcome single packet loss for every two packets using 1 x η diversity code. The proposed algorithm is simulated using GloMoSIM and its performance with respect to mobility, number of senders and number of multicast members in a group are studied and compared with the Node Transistion Probability based Multicast Routing (NTPMR) protocol. The results show improved performance of Packet delivery and end-end delay at the expense of increased control overhead.

Key words: MANET, routing, multicasting, multipath routing, diversity code

INTRODUCTION

Routing in MANET is a non-trivial task, because of frequent topology changes, it requires robust and flexible mechanisms to discover and maintain the routes. In addition, due to the power and bandwidth limitations, a routing protocol in MANET should fairly distribute the routing task to the hosts for the improvement of Quality of Service (QoS). However, the existing multicast routing algorithms such as Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)[1], Adhoc Multicast Routing Protocol (AMRoute)[2], Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol (MAODV)[3], On-Demand Multicast Routing protocol (ODMRP)[4,5] and Node Transistion Probability based Multicast Routing (NTPMR)[6] use either mesh or tree based algorithms to find the path to the destinations. In all these algorithms, only one path is used to route the packets. If that route fails, the packet is lost and the algorithm should start all over again to find a new route. Nasipuri and Das[7] proved that the use of multiple paths could maintain the end-to-end connection for a longer time than a single path. In other words, the frequency of searching for new routes is much lower if a node keeps multiple paths to the destination. In this paper, we propose a new reliable route diversity technique with diversity coding to self heal from failure. The performance of the proposed algorithm is analyzed using NTPMR Protocol, which provides more stable route than other multicast algorithms. The performance of NTPMR algorithm with and without route diversity algorithm is compared. Here, the data and code packets are forwarded using more than one best neighbor at the source node and only through the best neighbor at the intermediate nodes. In addition, the packets are encoded using diversity code[8] to combat single line failure.

CODED ROUTE DIVERSITY BASED MULTICAST ROUTING ALGORITHM (CRDM)

Multicasting is transmitting data from one or many sources to multiple destinations. The multicast transmission minimizes the link bandwidth consumption, sender and router processing, delivery delay when compared to many unicast links. And it also exploits the broadcast nature of the wireless link.

Model initialization: The requirement of a typical MANET Multicast routing protocol are:

- Group membership maintenance mechanism
- Route creation Mechanism
- Route maintenance mechanism
- Loop handling mechanism.

Corresponding Author: Dr. M.A. Bhagyaveni, Telematics Laboratory, Department of Electronics and Communication Engineering, College of Engineering, Anna University, Chennai-600 025, India
E-mail: bhagya@annauniv.edu

Group membership maintenance: In this algorithm the Group membership is maintained by using Hard-state approach i.e., whenever a node decides to join or leave a group, it conveys the corresponding JOIN_INFO or LEAVE_INFO message packet respectively to every node in the network. The nodes that are receiving these messages update their Multicast member table correspondingly.

Route creation mechanism: Route discovery with CRDM is purely on-demand. Each Node maintains a neighbour table (NBR table) and a multicast table for routing the data packets. The computation of these tables were discussed below.

When a node (source) has data to transmit to a group of nodes, it first sets up the neighbour table by initiating beacons. The node (say k) which receives the beacon replies with RREP packet and also initiates beacon transmission as shown in Fig. 1. The source node receiving the RREP records the power level of the beacon and reinitiates the flooding of the beacon till a preset FLOOD number of times. Each time the received power is recorded and a power matrix is computed. The power matrix is a N x n matrix, where N is the number of nodes, n is the number of flooding and each element P_{ij} is the power with which the node i replied to node k at j^{th} flooding. From the power matrix the frequency matrix M x N is computed, where, M is the number of significant power levels, N is the number of nodes and every element n_{ij} is the number of times the node j replied with power level i to node k. The neighbour table is obtained by multiplying the frequency matrix with a weight matrix. The weight matrix is chosen such that the results of recent flood are given more weightage. In the neighbour table the node addresses are arranged in the ascending order of transition probability, the first node in the table is the most reliable neighbour for that node. The step-by-step procedure to obtain the neighbour table is enumerated below.

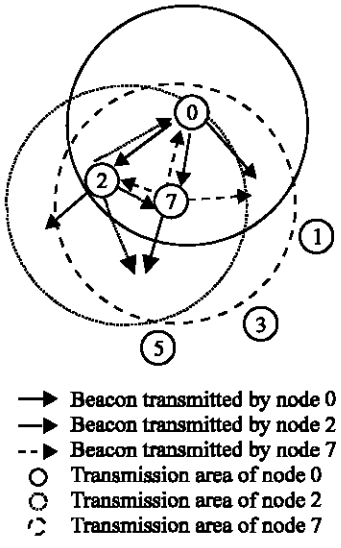

Fig. 1: Beacon transmission for neighbour table setup

The basic idea behind routing in CRDM is to assess the stability of neighbours by initiating beacons and computing the node transition probability matrix. The following are the steps involved in the computation of neighbour table.

Step 1: The first sender initiates the first beacon and the receiving neighbors re-initiate the beacons. The source node records the received power level of the beacons in the power table. This is repeated 'n' number of times. The power table has the dimension of N x n, where, N is the number of nodes and n is the number of flooding. The power table S_k for the k^{th} node is

$$S_k = \begin{bmatrix} p_{11}\ p_{12} \cdots p_{1j} \cdots p_{1n} \\ p_{21}\ p_{22} \cdots p_{2j} \cdots p_{2n} \\ \vdots \\ p_{i1}\ p_{i2} \cdots p_{ij} \cdots p_{in} \\ \vdots \\ p_{N1}\ p_{N2} \cdots p_{Nj} \cdots p_{Nn} \end{bmatrix} Nxn \quad (1)$$

where, p_{ij} is the power with which node i replies to node k during the j^{th} flooding and p_{kj} =0.

Step 2: After waiting for a finite interval of time Tw = 2 * n * t_n, The elements of the power matrix are arranged in the descending orders of power as $p_{1j} > p_{2j} > \ldots > p_{(N-1)j}$ and j =1,2, …n. Here, n is the number of flooding and t_n is the NODE_TRAVERSAL_TIME and is defined as the worst case time required for the packet to reach the node at the boundary.

Step 3: The index matrix X_k for the k^{th} node is formed as:

$$X_k = \begin{bmatrix} X_{11}\ X_{12} \cdots X_{1j} \cdots X_{1n} \\ X_{21}\ X_{22} \cdots X_{2j} \cdots X_{2n} \\ \vdots \\ X_{i1}\ X_{i2} \cdots X_{ij} \cdots X_{in} \\ \vdots \\ X_{N1}\ X_{N2} \cdots X_{Nj} \cdots X_{Mn} \end{bmatrix} Mxn \quad (2)$$

Where, X_{ij} refers to the id of the node replying with i^{th} power level in j^{th} flooding and is an element of the set

Table 1: Power level assignment

Power level	Power received in watts
1	>=3.07645e-07
2	(7.69113e-08,3.07645e-07)
3	(3.41828e-08,7.69113e-07)
4	(1.9227e-08, 3.41828e-08,)
5	(1.23058e-08, 1.9227e-08)
6	(8.5457e-09, 1.23058e-08)
7	(6.27847e-09, 8.5457e-09)
8	(4.80695e-09,6.27847e-09)
9	(3.79809e-09,4.80695e-09)
10	<3.79809e-09

R =1,2,3…k…N. If more than one node replies with the same power level, then the node with smaller id is assigned the higher power level and the next node id is assigned the next power level and so on without any loss of generality. M refers to number of power levels.

Step 4: The frequency matrix Y_k is formed as

$$Y_k = \begin{bmatrix} n_{11} & n_{12} & \ldots & n_{1j} & \ldots & n_{1N} \\ n_{21} & n_{22} & \ldots & n_{2j} & \ldots & n_{2N} \\ \vdots & & & & & \\ n_{i1} & n_{i2} & \ldots & n_{ij} & \ldots & n_{iN} \\ \vdots & & & & & \\ n_{M1} & n_{M2} & \ldots & n_{Mj} & \ldots & n_{MN} \end{bmatrix} M \times N \quad (3)$$

Where, n_{ij} refers to the number of times the node j has replied to node k with power level i. The power value received is continuous and usually small value that is expressed in an exponent form. Since the power values are very small (Order of 1 x 10^{-8}), it will be difficult to differentiate the weight computed for the two nodes with two different power levels. Hence, we consider power level instead of power values, without any loss of generality. The continuous power can take any value between Pmax (Transmitting Power) and Pmin (Threshold Power). This power range is divided into M power zones and each zone is assigned an integer to be used in calculation of the weight as shown in Table 1 for M =10.

Step 5: The probability matrix is formed after multiplying the frequency matrix by a weight matrix $W = [w_1, w_2, w_3 \ldots w_M]$ as:

$$P_k = \frac{[W][Y_k]}{(\sum w_i\ n)} = (p_1\ p_2\ p_3 \ldots\ p_j \ldots\ p_N), \quad (4)$$

where, w_i is the i^{th} weighted value and n is the number of flooding. In order to give highest probability for the node replying with highest power level, a weight matrix with weights $w_1 > w_2 > w_3 > \ldots > w_M$ is introduced. In our implementation, we have chosen $w_i = 2(M-i)$. Where, i =1,2, …,M. The higher the value of p_i, the larger is the probability that the node i is nearer to node k. In other words, p_i, indicates the nearness probability of each node with respect to the k^{th} node, whereas $q_i = 1-p_i$ indicates the transition probability of node i. The NTP matrix Q_k is then formed as:

$$Q_k = (q_1\ q_2\ q_3 \ldots . q_j \ldots . q_N), \quad (5)$$

Where, $q_j = 1-p_j$ is the probability of j^{th} node being the next hop node. After computing NTP matrix Q, each node deletes the matrix S_k,X_k, Y_k and P and preserves only matrix Q_k.

Step 6: The values q_j in the Q_k matrix indicates the transition probability of the j^{th} node with respect to the k^{th} node. The Q_k matrix is sorted in the ascending order of q_j values and the corresponding index is stored as the neighbour table N_k. The first entry in the table denotes the best neighbour of that node and second entry denotes the second best neighbour and so on.

Step 7: If a node has packet to send, it checks whether the destination is present in the neighbour table. If it is so, then it starts transmitting the packets. Otherwise, a search packet is sent to the best neighbour of the node.

Step 8: If the destination address is found in the neighbour table of the node receiving the search packet, the searching process is terminated. Otherwise, the packet is forwarded to the best neighbour of that intermediate node. This node is entered as the next hop for the particular destination in the Route Table (R_k) and the time at which this hop was found is also recorded. Search packet is not forwarded to the node from which it came. If a node encounters a search packet, which it has already processed, the packet is forwarded to the next best neighbour and the information is entered in the Route table. If TOE is the time of entry in the route table during the route discovery process, then the source starts sending the data at time T =TOE +$2t_n$, where, t_n is the NODE_TRAVERSAL_TIME.

Step 9: If there is a link breakage indicated by the MAC layer and the data packet is dropped, we choose the new route based on 2nd best neighbour using existing neighbour table without flooding the beacons once again. If also the second best neighbour failed, then the entries in the route and neighbour table are deleted and the new neighbour table is computed by initiating beacons and the process is repeated all over again from step 1 through 8 (Table 2).

Table 2: Neighbour table

Node	0	1	2	3	4	5	6	7	8	9
Neighbour ID	2,7,4	7,5,6,4,2	7,0,5,1	8,6,5	9,7,1,6,0	1,3,6	8,1,3,5	1,2,4,0	6,3,9	4,6,1

Table 3: Multicast table for node 1

No. of multicast group	Multicast group address	Join flag	Group members	Group size
1	3700561	1	7,3,8	3

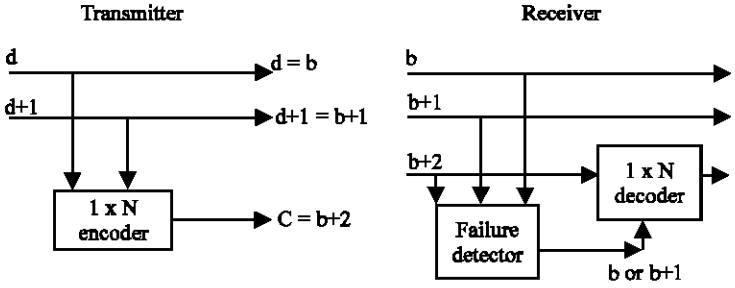

Fig. 2: 1-for-η diversity coding system

Multicast table formation:
The Multicast table contains the following fields:

- Number of Multicast Groups
- Multicast Group Addresses
- JOIN_FLAG
- Group Members
- Group Size

There can be more than one multicast group existing in a network. The first field contains the number of multicast groups in the network. The address corresponding to each group is stored in the second field of the table. The first two fields of the multicast table are updated during the beacon initiation process. If the node is a member of the group, the Join_FLAG is set. Any member can join the group at any instant of time and can remain so for a fixed period set by the source node. The 'Group Members' field contains the node addresses of the neighbouring multicast group members. The size of each group is stored in the Group Size field. The number in this field indicates the group size of the neighbouring node, which can be reached in a single hop from that node. The information about the neighbouring multicast group members is obtained through the beacon packets. The membership information field of the beacon packet contains this information. Each node that supports multicast routing in the network maintains this table even if it is a member of a multicast group. The multicast table is given in Table 3 for multicast source node 1 and 3,7,8 as the member nodes

Diversity coding: Diversity coding is a channel coding approach used for self-healing and fault tolerance. The principle theme of diversity coding is that if there are η information bearing links and μ overhead links for transmission and if out of μxη links atleast μ or more links are available at the reception side, then the original information can be reconstructed at the receiver. The transmitter and receiver block diagram of 1xη diversity coding is shown in Fig. 2.

The 1-for-η diversity coding works as follows: The packets are assigned a packet id d. The source node computes the codeword C as:

$$C = (d)^{th} \text{ packet} \oplus (d+1)^{th} \text{ packet}$$

where, ⊕ represents modulo 2 addition.

A BlockID b is assigned to these packets as b, b+1 and b+2, respectively.

In case of data packet loss at the receiver, the failure detector identifies the lost data packet BlockID as either b or b+1. The 1-for-η decoder recovers the lost data packet as,

Recovered Packet $b = (b+1)^{th}$ packet ⊕ $(b+2)^{th}$ packet, if b^{th} packet is lost
(OR)
Recovered Packet 'b+1' = $(b)^{th}$ packet ⊕ $(b+2)^{th}$ packet, if 'b+1'th packet is lost

Multipath multicast routing: The routing of information is done after computing the multicast table and neighbour table. We have restricted the number of multipaths used to 3 only because using of more paths may lead to unreliable routing and increase packet drop[9].

Any two data packets d, d+1 and its corresponding code packet are transmitted simultaneously through the

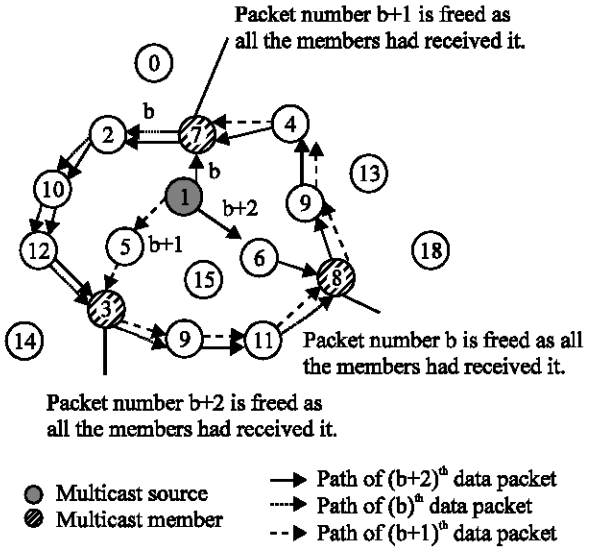

Fig. 3: Multipath routing of data packets in the network

best three neighbours of the source. The node handles the data packet after verifying the Data Seen Table, which contains the source address and the BlockID. Every node compares the source address and the BlockID present in the IPHeader of the received packet with field values of the Data Seen Table. This helps in avoiding duplication of the data packet received by a node. The multicast source transmits data with the destination address field set to multicast address. At each node, comparison is made between the Multicast Member Table and the Neighbour Table. The member field of the table, for the corresponding multicast address, is matched with the neighbour table entries. The first matched member of the group, in the neighbour table, is chosen as the destination of the data packet to be multicast. If there happens to be no members in the neighbour table, the packet is transmitted to the best neighbour. If the best happens to be already traversed node which is checked using the last_addr(2) field of the data packet, then the next best neighbour is chosen. The member nodes, which receive a data packet, record their addresses in the IPHeader of the data packet. This is done in order to avoid unnecessary transmission to a member node, which has already received the data packet. If the member addresses in the IPHeader field are the same as those of the group, then the retransmission process stops. Otherwise, the process of forwarding the data packet continues till all the multicast members receive the packet.

The Fig. 3 shows an example of multipath transmission of data packets by the source node1. The source node selects the member neighbour 7 and its best two neighbour 5 and 6 to transmits the packet numbered b^{th}, $b+1^{th}$ and $b+2^{th}$ packet respectively. The member node 7 after verifying the received packet with the Data seen Table and member address field in the IPHeader, it records its address in the IPHeader field of the packet and transmits the packets to its best neighbour 2. Similarly the nodes 5 and 7 which are not the members of the group, receives the data packets b+1 and b+2, respectively, checks it with the Data seen Table and checks the neighbour table for members. If member is its neighbour it transmits the packet to its member. In this example node 5 transmits the packet to member node 3 and node 6 transmits the packet to member node 8. The member node then verifies whether the member address field in the IPHeader is the same as the group address. if yes, the data packet is freed at that node. Otherwise the processes of forwarding packet is continued till the all the members receives the packet.

At the end of simulation each member node checks the BlockID b of the received packets and increments the number of packets received if its a data packet. The lost data packets are recovered using the corresponding data packet and its code packet.

Route maintenance mechanism: Route maintenance involves handling of link failures between the mobile nodes. Link breakage occurs due to node mobility and route expiration time. The MAC layer senses the link breakage and notifies the network layer of a packet drop. Once a packet drop occurs, the node initiates beacons in order to update its neighbour table. This ensures better stability of the routes and hence minimization of packet loss. The lost packet is recovered at the receiver using the code.

Loop handling mechanism: In Unicasting, looping does not arise when the destination address is present in the neighbour table since the data packet is not forwarded once it reaches the destination. Whereas, in multicasting looping may occur, since the data packet continue to circulate in the network even after reaching one of the member. Hence, the necessity for loop avoidance arises in multicasting. In order to avoid looping, the addresses of last two traversed nodes are stored in the last_addr(2) field of the IPHeader. If the node decides to transmit to one of its neighbors, it checks whether that destination address is present in the last_addr(2) field. If it is present, then it chooses the next best neighbor for transmission. This process continues till the chosen neighbour table is exhausted.

SIMULATION

The simulation package, GloMoSim[10] was used to analyze and evaluate the performance of our routing protocol. The GloMoSim (GLObal MObile information system SIMulator) provides a scalable simulation environment for wireless network systems. We simulated a network of 100 mobile nodes placed uniformly within a 1000x1000 m area. The Radio propagation range of 250 m and channel capacity of 2 Mb/s was chosen for each node. Each simulation was run for 10 min. Multiple runs with different seed values were conducted for each scenario and the collected data was averaged over those runs. Table 4 lists the simulation parameters and environments, which are used as default values unless otherwise specified. The following metrics were used in computing the protocol performance.

Packet delivery ratio: The number of data packets actually received by multicast members over the number of data packets supposed to be received by multicast members.

$$PDR = \frac{\left[\sum_{j=1}^{Ng}\sum_{i=1}^{Nj} R_{ij}\right]}{\left[\sum_{j=1}^{Ng} N_j.S_j\right]}$$

where, R_{ij} is the number of data packets actually delivered to the i^{th} destination in the j^{th} group, N_j is the number of Multicast members in the j^{th} group, N_g is the number of multicast groups and S_j is the number of packets sent by the multicast source to the j^{th} group. This number presents the effectiveness of a protocol.

End-to-end delay: The time elapsed between the instant when the source has data packet to send and the instant when the destination receives the data. Note that if no neighbour is available, the time spent in building a neighbour table is included in the end-to-end delay.

Control overhead: Is the ratio of total control bytes transmitted to data bytes delivered.

$$CBTD = \frac{(|N_B| + |N_j| + |N_D|)}{\sum_{j=1}^{Ng}\sum_{i=1}^{Nj} R_{ij}}$$

Where, the symbol | | represents size of and N_B is the total number of beacon packets, N_j is the number of join request packets and N_D is the bytes overhead in data packets in the networks.

Table 4: Simulation parameters

Number of nodes	100
Terrain range	1000X1000 square
Receiver power threshold	-81.0 dBm
Simulation time	10 min
Node placement	Uniform
Mobility model	Random way point model
Speed	0-80 Kmph
Propagation model	Two ray model
Channel bandwidth	2 Mbps
MAC	802.11
Multicast group size	10

The performance of the algorithm is studied comparatively by implementing multicasting using single path and multiple path with diversity coding.

RESULTS AND DISCUSSION

Many scenarios are configured using GloMoSim in order to investigate the performance of the proposed protocol by varying the network parameters such as mobile speed, number of multicast senders and multicast group size.

Mobile speed: In this simulation, each node moves at a predefined speed in a randomly selected direction. When the mobile reaches the simulation terrain boundary or the destination point it stays there for a defined pause time period then selects another random destination and direction and continues to move. The simulation parameter for the mobile environment is shown in Table 5 and the packet delivery ratio performance with respect to mobility is plotted in Fig. 4. From the results shown in Fig. 4, it is very clear that the Packet delivery ratio performance of CRDM algorithm is superior to that of NTPMR and it is due to the self-healing nature of the proposed CRDM algorithm. This improvement is achieved at the expanse of excess control overhead as seen in Fig. 5 and the increase in control overhead is due to the

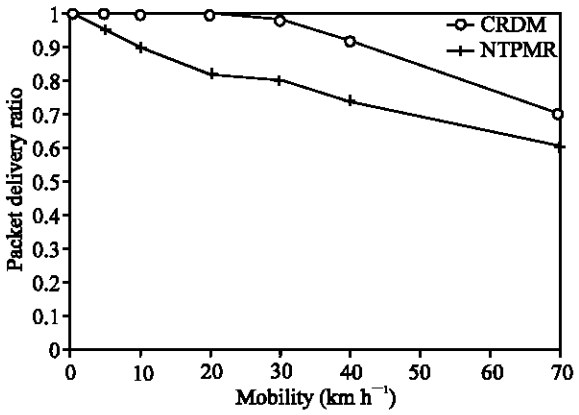

Fig. 4: Effect of mobility on PDR

Table 5: Mobile environment

Mobility model	Random waypoint
Mobile speed	0-70 Km/h
Multicast member	10
Data rate	2 pkts/sec
No. of sources	5

Table 6: Simulation environment for nnmber of senders

Mobile speed	1 m/s
Multicast members	10
Data rate	2 pkts/sec
No. of senders	1,2,5,10,20

Table 7: Simulation parameters for group size analysis

Mobile speed	1m/s
Multicast members	5,10,15,20
Data rate	2pkts/sec
No. of senders	5

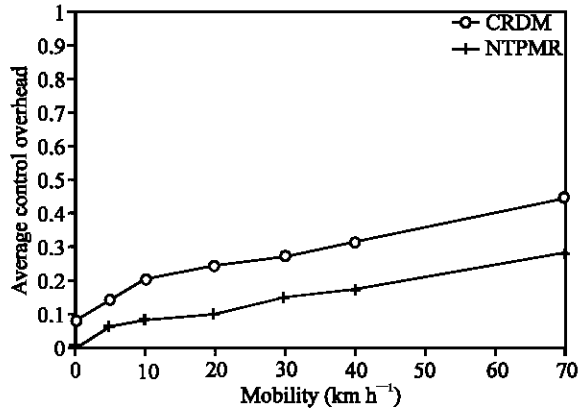

Fig. 5: Effect of mobility on control overhead

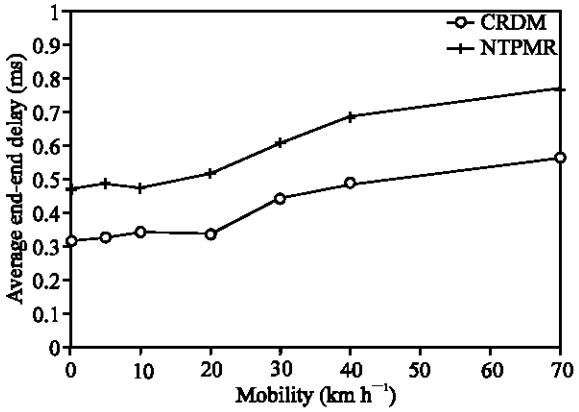

Fig. 6: Effect of mobility on average end-end delay

fact that it involves comparatively more nodes than NTPMR to transmit the data and the code packets.

The effect of mobility on average end-to-end delay is given in Fig. 6. The CRDM algorithm improves the delay performance of NTPMR by about 23%, which is due to the use of multiple paths simultaneously to transmit the data packets.

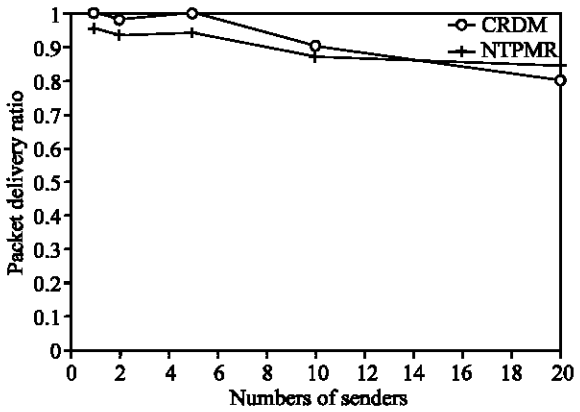

Fig. 7: Effect of number of senders on packet delivery ratio

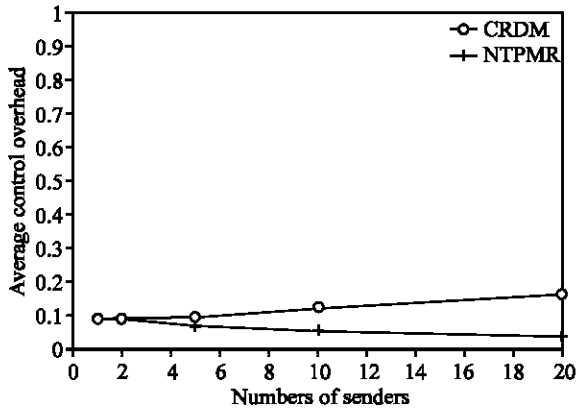

Fig. 8: Effect of number of senders on control overhead

Number of senders: The packet delivery ratio as a function of number of senders is shown in Fig. 7. The simulation parameters shown in Table 6 are used for this study. The results show that the CRDM protocol improves the PDR performance of NTPMR only for lesser number of senders. As number of senders becomes more than 10 the performance become worse than NTPMR this is because of increased number of collisions in the network. This is the area where improvement should be made to reduce the number of collisions.

The Fig. 8 shows the average control overhead as the number of senders is increased. The CRDM has higher control overhead than NTPMR and also the control packets increase as the number of senders are increased. This is because of increased number of collisions in the network.

Multicast group size: The scalability of the protocol is studied by varying the group size. The simulation parameters shown in Table 7 are used for this study and the results are shown in Fig. 9.

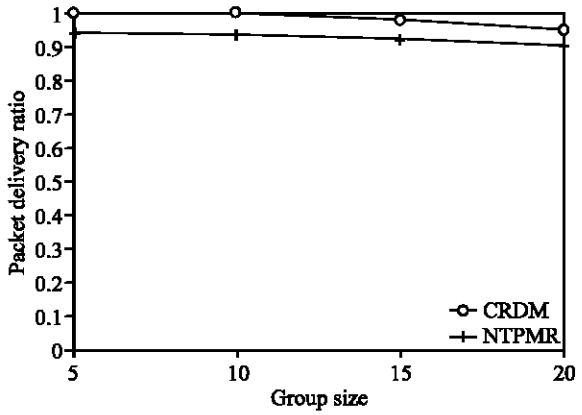

Fig. 9: Effect of group size on packet delivery ratio

The Fig. 9 shows the packet delivery performance of the algorithm with respect to group size. The CRDM protocol is sensitive to group size and as the number of members in a group increases the packet delivery ratio drops. This may be because of the fact that the multiple paths used are not strictly disjoint. But, the diversity code helps in achieving the improved throughput compared to NTPMR.

CONCLUSIONS

The coded route diversity technique is proposed for improving the performance of multicast routing protocol, NTPMR. This protocol utilizes three paths for transmitting the data and the code packets. The code packet is generated for every two data packets and transmitted in the same way as the data packets. The performance of the protocol is studied using the GloMoSim simulator and the results show improved performance of Packet delivery and end-end delay at the expense of increased control overhead.

The main advantage of CRDM is the use of more stable route with diversity code and hence achieves better packet delivery and end-end delay. The results of this work are not specific to the NTPMR protocol. The Multipath and Diversity coding technique are protocol independent and can be incorporated into virtually any on-demand protocol to improve that protocol's performance.

REFERENCES

1. Wu, C.W. and Y.C. Tay, 1999. AMRIS: A multicast protocol for ad hoc wireless networks. Proceedings of the IEEE Military Communications Conference (MILCOM), Atlantic City, NJ, pp: 25-29.
2. Bommaiah, E., M. Liu, A. McAuley and R. Talpade, 1998. AMRoute: Ad hoc multicast routing protocol, internet-draft, draft-talpade-manet-amroute-00.txt. (Work in Progress).
3. Royer, E.M. and C.E. Perkins, 1999. Multicast operation of the ad-hoc on-demand distance vector routing protocol. Proceedings of ACM/IEEE MOBICOM'99, Seattle,WA., pp: 207-218.
4. Sung-Ju Lee, Mario Gerla and Ching-Chung Chiang, 1999. On-Demand Multicast Routing protocol, Proceedings of IEEE WCNC'99, New Orleans, LA., pp: 1298-1302.
5. Bae, S.H., S.J. Lee, W. Su and M. Gerla, 2000. The design, implementation and performance evaluation of the on-demand multicast routing protocol in multipath wireless network. IEEE Network, 14: 70-77.
6. Radha, S. and S. Shanmugavel, 2002. Node transition probalility based multicast routing algorithm for mobile Ad hoc Networks. Proceedings of ICCC, 15th International Conference on Computer Communication. Nov. 11-14.
7. Nasipuri, A. and S.R. Das, 1999. On-Demand Multipath Routing for Mobile Ad Hoc Networks, Proceedings of the 8th International Conference. On Computer Communications and Networks (IC3N), Boston.
8. Ender, A., I. Chih-Lin, R.D. Gitlin and J.E. Mazo, 1993. Diversity coding for transparent self-healing and fault-tolerant communication networks. IEEE Trans. On Commn., 41: 1677-1684.
9. Bhagyaveni, M.A. and S. Shanmugavel, 2004. Multipath Routing with Self-Healing technique for Quality of Service (QoS) Support in Mobile Ad hoc Networks (MANET), IETE J. Education, 45: 1-5.
10. Glomosim User Manual, http://pcl.cs.ucla.edu/projects/glomosim