

MRBL: An Efficient Multicast Routing Protocol with Backup Labeling in MANETs

Po-Jen Chuang and Ting-Yi Chu

*Department of Electrical Engineering
Tamkang University
Tamsui, New Taipei City, Taiwan 25137, R. O. C.
pjchuang@ee.tku.edu.tw*

Abstract

The mobile devices in resource-constrained mobile ad-hoc networks (MANETs) usually communicate with each other without going through base stations. When the communication between mobile devices gets complicated due to fast changing technical developments and increased mobile devices, how to maintain transmission paths efficiently in order to elevate the network performance becomes a critical issue. To this purpose, this paper presents an advanced tree-based multicast routing protocol which uses an improved label mechanism to maintain the topology and multicast groups of a MANET, and uses the built backup paths to secure better transmission stability. Simulation results show that, with slightly more control overhead, the new routing protocol is able to yield constantly higher delivery ratios vs. increased data flow, when compared with other multicast routing protocols.

Keywords: *Mobile ad-hoc networks (MANETs), multicast routing protocols, tree-based, label-based, performance evaluation*

1. Introduction

In a MANET, multicast routing can be mesh-based [1-2] or tree-based [3-4]. Mesh-based routing needs considerable control packets to set up or maintain the topology and routes; it works well in enhancing path stability and repairing invalid paths. Tree-based routing, by contrast, can largely reduce the required control packets as it uses a core node in each group to manage the group members. In face of invalid paths, tree-based routing is less tolerant as it has only one path between the source-destination pair, while mesh-based routing is more tolerant because it allows a node to build one or more paths to the source or destination and can thus adjust a routing path, when necessary, to avoid intercepted data transmission. Mesh-based multicast routing is nevertheless undesirable for a MANET with huge transmission flows (today's practical trend) which needs a lot more control packets to maintain the topology and paths and will thus affect the transmission flows, delivery delay time and delivery success ratios. For a super busy MANET, tree-based multicast routing may turn out a better routing choice if we can bring down the cost for building and maintaining paths lower than that for repairing the broken paths.

There are a number of multicast routing protocols introduced in recent years, *e.g.*, [1-7]. In [5], AMRIS uses a shared tree to manage the multicast groups and transmit data, and gives each node a unique ID number for identification when the link changes. In [6], MAODV adds a unique sequence number to each multicast entry and allows group heads to manage the numbers, to avoid route loops or duplicate usage of intermediate nodes on the path. Among other multicast routing protocols, AMRoute [3], like AMRIS, is a shared-tree multicast

routing protocol, LAM [2] employs group shared trees which need no assistance of timing parameters, while ODMRP [1] and CAMP [4] are both based on the mesh topology. With increased sources and multicast groups, POEM [7], which sets a unique label to each node and a core node to manage multicast groups, can directly build paths between sources and destinations to facilitate data transmission and strengthen the overall network performance.

This paper presents a new tree-based multicast routing protocol which uses an advanced label mechanism to reduce the needed control packets for topology initialization and path construction, and meanwhile introduces a new mechanism able to select a better routing path for transmission. The key value of our new protocol is to improve the path repairing problems (when broken or invalid) of existing tree-based multicast routing protocols and to resultantly raise the data delivery ratios. Extended simulation runs are conducted to evaluate the performance of our multicast routing protocol and related protocols for comparison. The results show that, with its special designs and slightly more control overhead, our new protocol can produce constantly higher delivery ratios and shorter latency under increased data flows than other protocols.

2. The Proposed Routing Protocol

As mentioned, in multicast routing protocols for MANETs, ODMRP [1] is mesh-based while POEM [7] is tree-based. As ODMRP lets source nodes broadcast control packets periodically to maintain the groups, the required control packets will increase when the receivers increase, i.e., when nodes multiply, control and data packets multiply too. In such a situation, despite there are multiple paths between the nodes, the increased control packets can still affect the bandwidth of data transmission and as a result reduce the delivery ratios. The tree-based POEM uses labels to mark the nodes when building the topology, assigning each node a unique label. When nodes change locations or paths become invalid, it has no effective way to handle the situations but to drop the data packets – just like other tree-based multicast routing protocols. To solve the problem, our new tree-based multicast routing protocol employs a more advanced label mechanism which uses labels to maintain the topology and groups and, more importantly, to build the backup paths. These additional backup paths help us attain higher stability than other tree-based protocols. To raise the performance at reasonable cost, we have managed to reduce the number of control packets required for topology/group maintenance and also to fortify the paths in every group with less additional control packets.

The flow of our protocol, briefed as MRBL (Multicast Routing with Backup Labeling), is as follows.

1. Building the topology: Select a node to broadcast the control packet to the other nodes in the topology. Initially label each node that receives the control packet. When receiving other packets later, a node will keep additional labels in the buffer.

2. Data transmission: Upon receiving a request, a source can use the label of the requesting node to set up paths for transmitting data to the destination. When transmitting data, the source will check if it is necessary to use the backup paths.

3. When to use the backup paths: The source will check the labels of all destination nodes to see if there are similar labels in the same group. If no, use the original route; if yes, use the backup paths.

2.1. The Operating Process

2.1.1. Building the Topology: The core node broadcasts the label packet to the network to label each node that receives it. Nodes periodically broadcast control packets to neighbor nodes to update neighbor tables and build the topology.

- (1) Each node receives the label packet and checks its time stamp.
- (2) Each node gets a label (according to the node level in the tree).
- (3) Each node updates neighbor information in its neighbor table.
- (4) After receiving the first label, each node also records the alternative label (according to the alternative level in the tree) for backup use.
- (5) Use the neighbor information to build the initial topology.

2.1.2. Data Transmission: When a source receives a request packet, it will

- (1) check the time stamp in the packet,
- (2) build a path according to the specified destination labels,
- (3) check if there is a better routing path built from the backup labels (by checking the “similarity” – possible parent-child relationship – between destination labels, to be further discussed in Sec. 2.2), and then
- (4) transmit the packet by the original path or the better path.

2.1.3. Route Maintenance:

(1) Node joining: To join a multicast group, a new node needs to send a request packet to the neighbor nodes and update the information in its neighbor table (Figure 1).

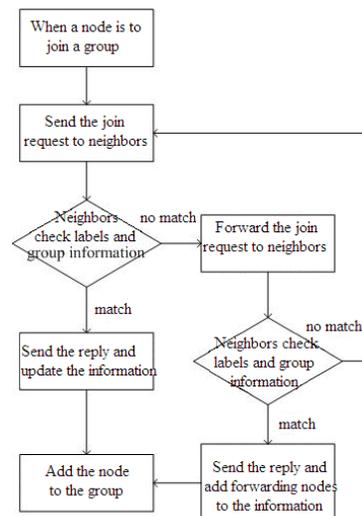


Figure 1. The Flowchart of Node Joining

(2) Route maintenance: In face of a broken path, a node first checks its neighbors for an alternative valid path, and then sends a control packet to update the information in neighbors (Figure 2).

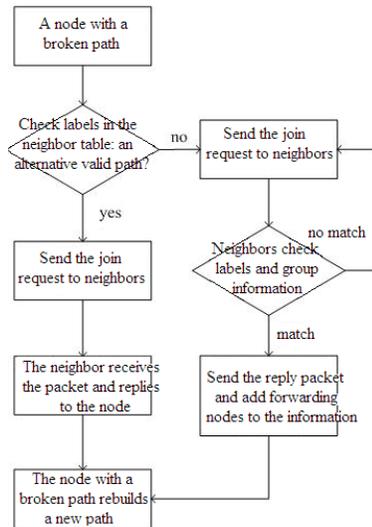


Figure 2. The Flowchart of Route Maintenance

2.2. The Backup Paths

On top of the original POEM routing, our label mechanism allows us to create an additional and available backup path by checking the similarity between node labels, and use such a backup path to carry on transmission in face of link failure. Unlike POEM which lets a node carry only one label, we let each node carry additional backup labels. When link failure breaks data transmission to a specific node, we can repair the routing path by using the alternative backup labels of that node. Figure 3 gives an example to illustrate the forming and use of our backup labels. As we can see, nodes e (01) and f (02) belong to different branches of the tree but are within the communication range of each other. We let e and f respectively keep the alternative labels originating from each other (022 and 013) as backup labels and use them to restore broken paths when necessary. Likewise, nodes i (0121) and j (0211) also keep labels originating from each other (02111 and 01212) as backup labels.

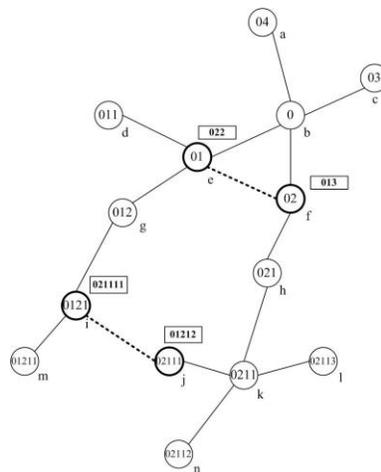


Figure 3. An Example of Keeping the Backup Labels

Employing the backup paths: When a target node needs to transmit a packet, the source node will check if the destination nodes hold similar node labels (to form backup labels). If yes, the source will send a request to those destinations with similar labels for path updating, to attain a better routing path and more desirable transmission performance – as the flowchart in Figure 4 depicts.

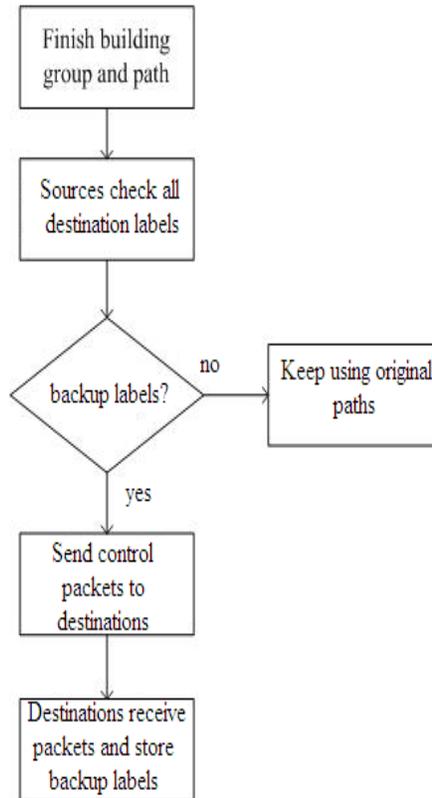


Figure 4. The Flowchart of our Protocol

2.3. An Example of Performing Our Protocol

For better clarity, we illustrate the performance of our protocol in Figure 5 where the thick-circled nodes are members of the same group while the thin-circled nodes are general or forwarding nodes. Assume node d in Figure 5(a) is a source node. When node d needs to transmit data to the group, it must go through two branches d-e-g-i-m and d-e-b-f-h-k-j-l-n to reach all members. By the backup mechanism of our protocol, node d will realize node i is associated with node j, *i.e.*, both nodes are in the transmission range of each other (as the dotted line indicates), and then send a request packet to node i. Upon receiving the request packet, node i will send the packet to node j to build a link between each other (as Figure 5(b) shows). Node d now attains a new and better data transmission path d-e-b-f-h-k-j-l-n by which it is able to reach all group members. Via this new path, data transmission from node d to both nodes i and m can go through node j – without passing nodes e and g, thus reducing the network load and the amount of control packets due to route maintenance.

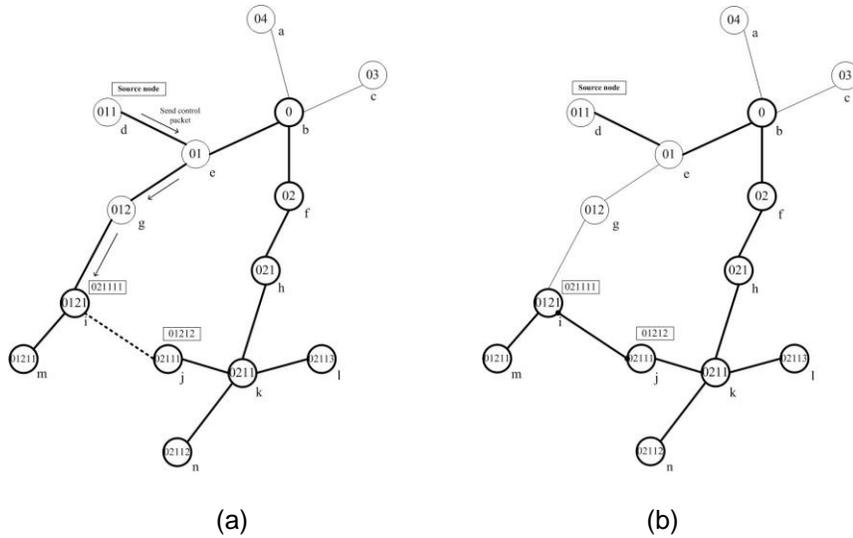


Figure 5. An Example of Performing our MRBL

3. Simulation Results

Experimental evaluation is carried out to compare the performance of MRBL, ODMRP and POEM. Table 1 lists the involved simulation parameters.

Table 1. Parameters in Simulation

Parameters	Value
Transmission Range	250m
Number of Senders	0~20
Number of Receivers	0~100
Speed of nodes	1~10m/s
Simulation duration	300s
Packet Size	256bytes
Mobility Pause Time	50~300ms
Movement model	Random Waypoint
Scenario Size	1000*1000 m ²

Our simulation uses the random way point model in NS-2 (version 2.34 [8]), to set up a 1000m*1000m area with 100 nodes. Node speeds vary from 1 to 10 m/s; each node has pause time with mean 50 to 300ms. The simulation duration is 300 seconds; we run the simulation at least 10 times and average the results to remove the biased topology due to node placement. The performance measures of interest include the control packet overhead, data end-to-end delay, and packet delivery ratios, which are obtained with different lengths of pause time and numbers of source nodes.

3.1. Performance Measures vs. Pause Time

Figures 6-7 exhibit how pause time affects the three performance measures under the same topology and fixed number of groups and nodes.



Figure 6. The Delivery Ratios

We increase the pause time from 50 up to 300 seconds, to show how the three protocols perform in normal cases. It shows that ODMRP needs much more control packets because of its mesh-based topology, while the tree-based MRBL and POEM produce less control overhead. MRBL is shown to produce slightly higher control overhead than POEM because it needs extra control packets to maintain the backup path mechanism (Figure 7).

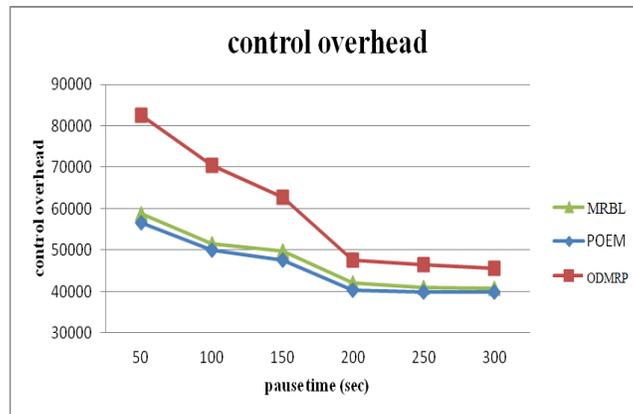


Figure 7. The Control Overhead

We also see that our MRBL yields constantly better packet delivery ratios than the other two protocols (Figure 6) due to its distinct backup feature. This is because, using slightly more control packets, our backup mechanism can work well in route repairing/updating which not only raises the packet delivery ratios but also reduces the cost of path reconstruction and the transmission delay time.

3.2. Performances Measures vs. Numbers of Sources

With more source nodes in the network, the network will be more heavily loaded as each node tends to receive more data packets. Figures 8-9 give the three performance measures obtained under various source numbers for the protocols.



Figure 8. The Delivery Ratios

It is obvious that ODMRP which lets sources transmit data packets and broadcast control packets needs more control packets than other protocols when the sources increase. It significantly increases the network load and delay time, and consequently decreases the delivery ratios. MRBL and POEM, both using the label-based mechanisms to assist route building and maintenance, can reduce the usage of control packets even when source nodes increase. MRBL is shown to take a bit more control overhead than POEM due to its backup mechanism, which nevertheless helps secure constantly higher delivery ratios by its alternative route selection.

When the source number grows larger, the backup mechanism of our protocol helps ensure even better performance. This is because when source nodes and data packets increase, our protocol can shorten the transmission latency while strengthen the delivery ratios by reducing the path lengths and the usage of control/data packets.

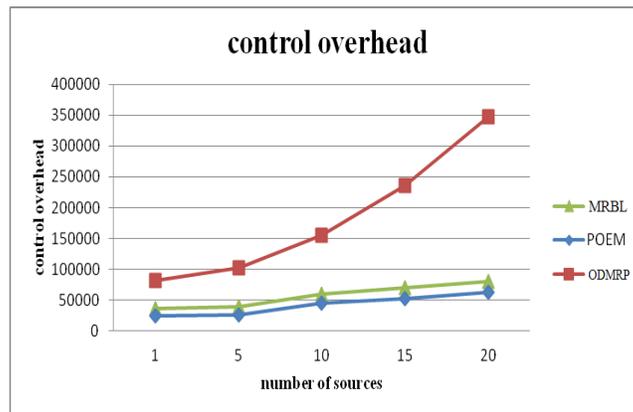


Figure 9. The Control Overhead

3.3. Performance Measures vs. Numbers of Groups

To create more practical situations, we build up to five multicast groups in the simulated network. By doing so, we produce more complexity in the network topology and more difficulty in maintaining the inter-group relations. The performance of the protocols under different multicast group numbers is collected in Figures 10 and 11. As we can see, ODMRP performs well when the network has fewer groups. When the multicast groups grow beyond

three, its performance drops remarkably as it needs tremendously increased control packets to maintain the groups. For ODMRP, the large amount of control packets will cause network overload, slow down data transmission and as a result increase the transmission delay time. By contrast, our MRBL needs stably less control packets regardless of the group numbers, thanks to its backup path mechanism which helps attain better utilization of the control packets and as a result successfully contain the control overhead. The result reassures the core value of our backup path mechanism: Even when multicast groups grow in the network, it can constantly locate a more desirable transmission path with less control overhead (in contrast to the original path) but better performance (because the consumption of control packets will directly affect packet delivery ratios and delay time).

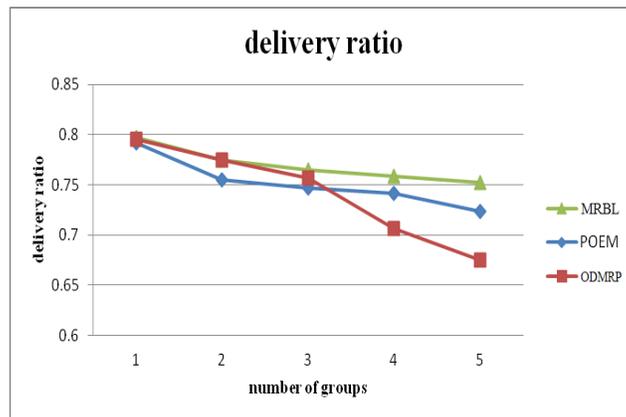


Figure 10. The Delivery Ratios

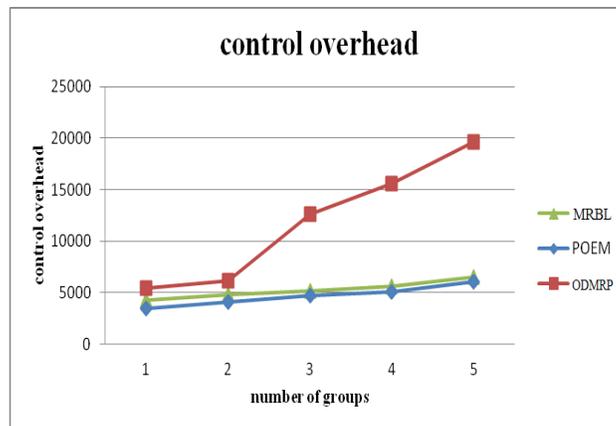


Figure 11. The Control Overhead

3.4. Performances Measures vs. Numbers of Group Members

In Figures 12-13, we set up the group with different numbers of members to check the how the protocols work in route building and maintenance when the group members grow. It shows that the usage of control packets for POEM and our protocol is less than that for ODMRP, and the gap widens with the increase of group members. When group members reach 60 nodes, *i.e.*, when topologies diversify, our protocol outperforms POEM. This is because our mechanism can handle the situation better and therefore yield lower transmission delay time. With the members of a group increase to nearly the entire network nodes, the

benefit of our protocol will shrink, making the delivery ratio close to that of POEM. As for ODMRP, its control packet consumption significantly rises with the increase of group members. Thus, when group members increase, ODMRP tends to build up network overload and transmission delay. Under the same bandwidth, when control packets increase, data packets will decrease.



Figure 12. The Delivery Ratios

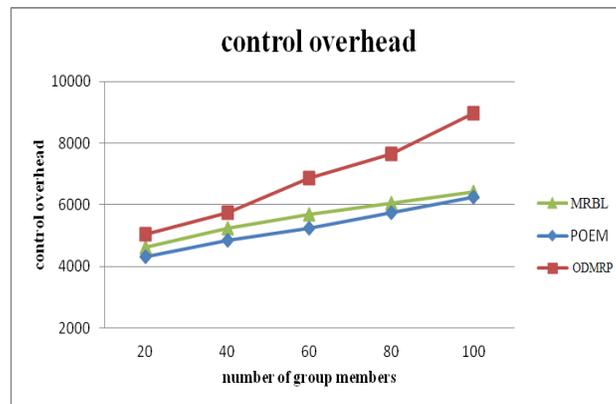


Figure 13. The Control Overhead

3.5. Performances Measures vs. Node Speeds

Figures 14-15 demonstrate how the node speed (or fast changing topologies) affects the control packet consumption and packet delivery ratio of the protocols. The results show a general performance trend, *i.e.*, the control overhead grows while the delivery ratio drops with the increase of the node speed. When node speed = 10m/s, we can see the three protocols yield almost the same delivery ratio, but the control packets and delay time are still problems for ODMRP. At node speed = 20m/s (when the topology is under rapid changes), ODMRP apparently needs more control packets to maintain the topology. It thus increases the network overload and causes packet dropping/loss during congestion or path rebuilding. In contrast to ODMRP, our MRBL exhibits a minor performance decrease at higher node speeds because we use the label mechanism to reduce the required control packets for path maintenance and the backup path mechanism to rapidly update/repair the link or path between nodes. Moreover, with slightly more control overhead, our protocol outperforms even POEM in packet delivery ratios, regardless of the node speed.

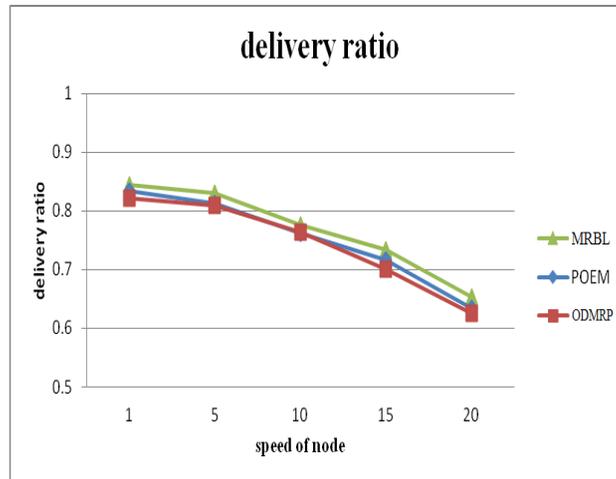


Figure 14. The Delivery Ratios

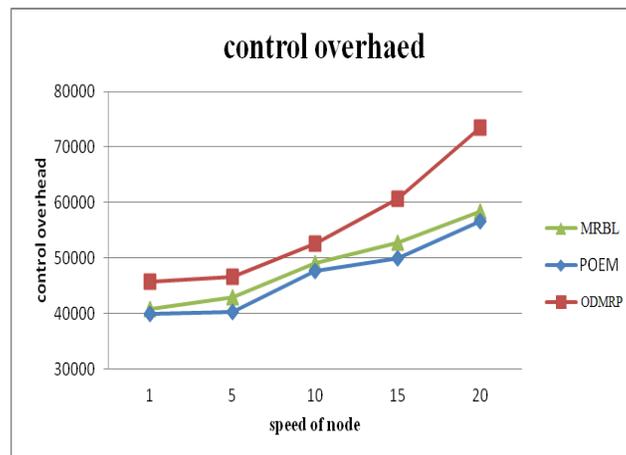


Figure 15. The Control Overhead

4. Conclusions

More control packets can lead to more stable multicast groups and better data delivery ratios in a MANET. When facing link failures due to mobile nodes, the network will need control packets and multicast groups for maintenance. But, more control packets in a bandwidth-constrained MANET may overload the network, reduce the data flow and cause packets to drop or lose. In this paper, we propose a tree-based multicast routing protocol whose advanced label mechanism helps reduce the needed control packets for initializing the topology and building routing paths and also helps to select an optimal routing path for transmission. By shortening the length of paths used to transmit data packets and control packets, our protocol is able to reduce the network load and repair link failures due to the high speed of nodes. Simulation results prove that, with its special labeling design and slightly more control overhead, our MRBL outperforms related protocols in terms of *data delivery ratios* and *transmission latency* in most cases.

References

- [1] M. Gerla, S. J. Lee and W. Su, "On-demand Multicast Routing Protocol (ODMRP) for Ad-hoc Networks", Internet draft, draft-ietf-manet-odmrp-02.txt, (2000).
- [2] L. Ji and M. S. Corson, "A Lightweight Adaptive Multicast Algorithm", Proceedings of the IEEE GLOBECOM, (1998), pp. 1036-1042.
- [3] J. Xie, R. Rajesh and A. McAuley, "AMRoute: Ad-hoc Multicast Routing Protocol", Mobile Networks and Applications, Multipoint Communication in Wireless Mobile Networks, vol. 7, no. 6, December (2002), pp. 429-439.
- [4] J. J. Garcia-Luna-Aceves and E. L. Madruga, "Core-assisted Mesh Protocol", IEEE J. Select. Areas Commun., vol. 17, no. 8, (1999), pp. 1380-1394.
- [5] C. W. Wu and Y. C. Tay, "AMRIS: A Multicast Protocol for Ad-hoc Wireless Networks", Proceedings of the IEEE MILCOM, vol. 1, (1999), pp. 25-29.
- [6] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc on Demand Distance Vector Routing Protocol", Proceedings of the ACM MOBICOM, (1999) August, pp. 207-218.
- [7] J. J. Garcia-Luna-Aceves and D. Sampath, "Efficient Multicast Routing in MANETs Using Prefix Labels", Proceedings of the IEEE International Conference on Computer Communication and Networks, (2009) August, pp. 1-8.
- [8] NS2, <http://www.isi.edu/nsnam/ns/>.

Authors



Po-Jen Chuang, received the B.S. degree from National Chiao Tung University, Taiwan, R.O.C., in 1978, the M.S. degree in Computer Science from the University of Missouri at Columbia, U.S.A., in 1988, and the Ph.D. degree in Computer Science from the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, U.S.A. (now the University of Louisiana at Lafayette), in 1992. Since 1992, he has been with the Department of Electrical Engineering, Tamkang University, Taiwan, where he is currently a Professor. He was the department chairman from 1996 to 2000. His main areas of interest include parallel and distributed processing, fault-tolerant computing, computer architecture, mobile computing, network security and cloud computing.



Ting-Yi Chu, received his B.S. and M.S. degrees in Electrical Engineering in 2009 and 2013 from Tamkang University, Taiwan. His research interests include network management and routing protocols in MANET.