A Power-Aware Dual-Tree-Based Multicast Routing Protocol for Mobile Ad Hoc Networks

Nen-Chung Wang[1] and Ying-Shiou Chen[2]
[1]Department of Computer Science and Information Engineering
National United University, Miao-Li 360, Taiwan, R.O.C.
Email: ncwang@nuu.edu.tw
[2]Graduate Institute of Networking and Communication Engineering
Chaoyang University of Technology, Taichung 413, Taiwan, R.O.C.
Email: s9330612@cyut.edu.tw

Abstract

A mobile ad hoc network (MANET) is an interconnected system of mobile hosts without a fixed infrastructure. In MANETs, each mobile host has multi-hop transmission capability, and it has to serve as a router. Because of the dynamic topology and limited resources of mobile hosts, the routing scheme in MANETs presents an important challenge. To improve battery power efficiency in MANETs, many routing algorithms have been proposed. Similarly, a stable routing path is equally important for MANETs. In this paper, a power-aware dual-tree-based multicast routing protocol (PDTMRP) for MANETs is proposed. Battery power was used in the route discovery to eliminate unstable paths. To achieve the load balance two multicast trees were constructed. Finally, simulations were conducted to show that the proposed routing scheme outperforms the MAODV routing scheme.

Keywords: Load balance, mobile ad hoc networks, multicast routing, power-aware, tree-based routing.

1. Introduction

A mobile ad hoc network (MANET) is an interconnected system of mobile hosts without a fixed infrastructure. Every node in an MANET must be able to function as a route to forward data to other nodes. When applications must send the same data to more than one destination, multicasting is often used. Multicasting reduces the communication costs for applications that send the same data to multiple recipients. Instead of sending via multiple unicasts, multicasting minimizes the link bandwidth consumption, router processing, and delivery delay. Existing multicast routing protocol for MANETs can be broadly classified into tree-based routing protocols [1, 2, 3, 6, 12, 16, 20] and mesh-based routing protocols [4, 7, 8, 9, 18].

Tree-based routing protocols build a tree structure that connects all multicast members and provide one path between a pair of source and destination nodes. Mesh-based protocols yield a multi-path between the source and the destination nodes. When a link fails, mesh-based multicast protocols do not need to re-compute a mesh. Thus mesh-based multicast protocols have a high packet delivery ratio compared to tree-based protocols, but they incur more control and network overhead by flooding through the mesh.

Many multipath routing protocols have been proposed for load balancing data transmission in MANETs [11, 13, 17, 19]. In [13], Perlman et al. demonstrate the effect of alternate path routing (APR) on load balancing and end-to-end delay in MANETS. In [11], Split Multipath Routing (SMR) builds and maintains maximally disjoint paths to avoid network congestion. In [17], Delay-Sensitive Adaptive Routing Protocol (DSARP) selects several shortest paths, or alternatively selects the shortest path plus the next shortest path. Then, the source node adjusts traffic flow according to the total number of packets on each selected path. In Multiple Tree Video Multicast over Wireless Ad Hoc Networks [19], Wei et al. propose a parallel multiple nearly-disjoint multicast trees routing (Parallel MNTMR) scheme. The Parallel MNTMR establishes two disjoint multicast trees to split the video into two parts and send each part over a different tree.

In this paper, we propose a power-aware dual-tree-based multicast routing protocol (PDTMRP). In this scheme, battery power is used to eliminate the unstable nodes in order to achieve high reliability. Two multicast trees are constructed to achieve load balance.

The rest of this paper is organized as follows. Section 2 presents related work. The proposed scheme is developed in Section 3. Section 4 describes the experimental results. Finally, Section 5 draws the conclusions.

2. Related Work

In this section, two related multicast routing protocols, called MAODV and Parallel MNTMR, are introduced.

2.1. Multicast Ad Hoc On-Demand Distance Vector Routing Protocol (MAODV)

The MAODV [16] is based on Ad-Hoc On-Demand Distance Vector Routing (AODV) [14], and it is an extension of AODV in supporting multicasting. MAODV establishes on-demand multicast tree and uses these for delivery of multicast data. MAODV uses a shared group tree and periodically uses hello messages for link break detection and group leader floods for group information dissemination. When a mobile node wishes to join a multicast tree or has data to send to a multicast group but it does not have a route to that group, the mobile node broadcasts a route request (RREQ) packet. Only members of the desired multicast group can respond to a RREQ.

When an intermediate node receives a RREQ packet, the intermediate node rebroadcast the RREQ to its neighbors. A node receiving the RREQ packet may unicast a route reply (RREP) to the source node if it is either the destination or a member of the multicast tree with a corresponding sequence number greater than or equal to that of the RREQ. As nodes along the path to the source receive the RREP, they add both a route table and a multicast routing table entry for the node from which they received the RREP.

After the source node receives the RREPs, it selects the route with the largest sequence number and shortest hop count from the RREPs and sends a multicast activation (MACT) message to select its next hop. The MACT message activates the route. The next hop node receiving the MACT message enables the entry for the source node in its multicast routing table. If the node is a member of the multicast tree, it does not send the MACT message any further. However, if the intermediate node is not a member of the multicast tree, it receives several RREPs from its neighbors. The MACT message ensures that the multicast tree does not have multiple paths to any tree node. The intermediate node forwards data packets only along the activated route.

2.2. Parallel Multiple Nearly-Disjoint Trees Multicast Routing Protocol (Parallel MNTMR)

The Parallel Multiple Nearly-Disjoint Trees Multicast Routing Protocol (Parallel MNTMR) [19] establishes two multicast trees to reduce data retransmission. Parallel MNTMR splits the video into two parts and sends each part over a different tree. Parallel MNTMR first classifies all the nodes randomly into one of two types, group-0 or group-1, based on uniform distribution. Then it can construct tree-0 only from nodes in group-0, and tree-1 only from nodes in group-1. When a mobile node has to join a multicast tree or has data to send to a multicast group but it does not have a route to that group, it broadcasts a join query (JQ) packet. Each intermediate node immediately broadcasts the first JQ packet if the last hop of the JQ packet is the sender or the same group type as the node belongs to. Otherwise, the intermediate node broadcasts the JQ packet after a short delay if the intermediate node has not already broadcast a JQ packet in the same JQ round.

When a receiver receives a pure JQ packet with group-0 and pure JQ packet with group-1, the receiver selects the last hop of this JQ packet as its upstream node for tree-0 and tree-1 if it has not sent out a join relay (JR) packet in JQ round. The objective of Parallel MNTMR is to maximize the disjointness of two trees. If the JQ packet is forwarded by nodes in both a group-0 and group-1, the last hop of the previously received group-0 and group-1 JQ packet is selected as an upstream node. Each node, on receipt of a JR packet, selects an upstream node based on a corresponding type of JR packet.

3. Proposed Power-Aware Dual-Tree-Based Multicast Routing Protocol (PDTMRP)

In this section, a power-aware dual-tree-based multicast routing protocol (PDTMRP) is proposed. The parallel MNTMR routing scheme was extended for load balancing concept in this study. In the proposed scheme, it was assumed that all nodes could be randomly classified into two types, i.e., group-0 or group-1. Then the dual trees (tree-0 for group-0 and tree-1 for group-1) for data transmission were constructed. Each node maintained two routing tables: the neighboring table and the routing table.

The format of the neighboring table was <*node_ID, distance*>. The format of the routing table was <*source_ID*, *destination_ID*, *seq_number*, *route_class*, *next_hop*>. The *source_ID* and *destination_ID* fields contained the unique addresses of the source and the destination node, respectively. The *seq_number* field contained the sequence number of the source node. The *route_class* field recorded the class of route for group-0 or group-1. The *next_hop* field contained the address of the neighbor node to which data packets had to be forwarded.

3.1. Route Discovery Process

In the proposed scheme, power level threshold ($P_{threshold}$) was defined. When the source node wanted to send the packet to the destination nodes, it broadcasted the route request (RREQ) packet to the neighboring nodes in its transmission range, when the source node did not have a path in the routing table. Fig. 1 illustrates the format of the RREQ packet. Each RREQ packet was associated with a unique *Broadcast ID*; together with the *source ID*, each node identified a RREQ packet. *Broadcast ID* and *Source ID* form an identifier pair. After neighboring nodes received the RREQ packet, the neighboring nodes first checked the remaining battery of nodes (P_{remain}). When P_{remain} of nodes was higher than $P_{threshold}$, the neighboring nodes store received the RREQ packet and re-broadcasted the RREQ packet. The neighboring node added its ID to the routing path field of the RREQ packet and the *Class* field of the RREQ packet was assigned a type of neighboring node.

Type RREQ	$P_{threshold}$	Class	Broadcast ID	Source ID	Destination ID	Routing path

Fig. 1. RREQ packet format.

When the destination node received the first RREQ packet with group-0 and the first RREQ packet with group-1, the destination node selected the last hop of each RREQ packet as its upstream node to be the primary routing paths for tree-0 and tree-1. Then, the destination node sent two route reply (RREP) packets to the source node. Fig. 2 illustrates the format of the RREP packet. The *Class* field of RREP packet was the assigned type for the RREQ packet. When the intermediate node received the RREP packet, it selected the upstream node based on the corresponding type of RREP packet and sent the RREP packets to the source node. The detail of the route discovery process is shown in Algorithm 1.

Type RREP	Class	Destination ID	Source ID	Routing path

Fig. 2. RREP packet format.

Algorithm 1: Route Discovery Process

A network is modeled as graph $G = (N, E)$, where N is the finite set of mobile nodes and E is a set of links. Suppose n is the number of mobile nodes and N is the set of mobile nodes $N = \{N_1, N_2, ..., N_n\}$. Assume that source node N_i wants to find a path to destination node N_j. Node N_i broadcasts a RREQ packet, and node N_k receives the RREQ packet, where N_i, N_j, N_k $\in N$, $1 \le i, j, k \le n$, and $i \ne j$.

```
if (node $N_k$ is the destination node $N_j$) {
    (1) Node $N_k$ selects the first RREQ packet with group-0 and RREQ with group-1 as the upstream node and unicasts a RREP packet to the source node.
    (2) Each node receives the reply RREP packet and writes the entry to the current routing table. Then the node selects an upstream node with a corresponding type of RREP.
}
else if ($P_{remain}$ of node $N_k$ is higher than $P_{threshold}$ ) {
    (1) Node $N_k$ stores the received RREQ packet in its list of upstream nodes.
    (2) Node $N_k$ forwards the RREQ packet to the neighboring nodes.
}
else
    Node $N_k$ discards the request packet.
```

Let us consider the example shown in Fig. 3. It is assumed that the $P_{threshold}$ is 20. In Fig. 3(a), the source node *S* wants to send data to some nodes *D* and *H*. Node *S* broadcasts a route request (RREQ) packet to its neighboring nodes. Nodes *A* and *B* forward the request packet and append their own information, such as their own ID and the type of class when they receive RREQ packet. In this example, P_{remain} of node *C* is 18 with lower than $P_{threshold}$, node *C* will discard the RREQ packet. In Fig. 3(b), destination node *D* selects node *E* and node *G* as upstream node for tree-0 and tree-1, respectively. Destination node *H* selects nodes *E* and *I* as upstream nodes for tree-0 and tree-1, respectively.

Then each destination sent the route reply (RREP) packet back to the upstream nodes. Node *E*, on receipt of RREP packet, selects node *A* as upstream node for the tree with group-0, node *G* selects node *B* as upstream node for tree-1. While receiving the RREP packet, node *I* selects node *F* as upstream node for tree-1. Lastly source node *S* receives two multicast path for tree-0 and tree-1. Finally, as shown in Fig. 3(c), these routing paths are used to construct two multicast trees.

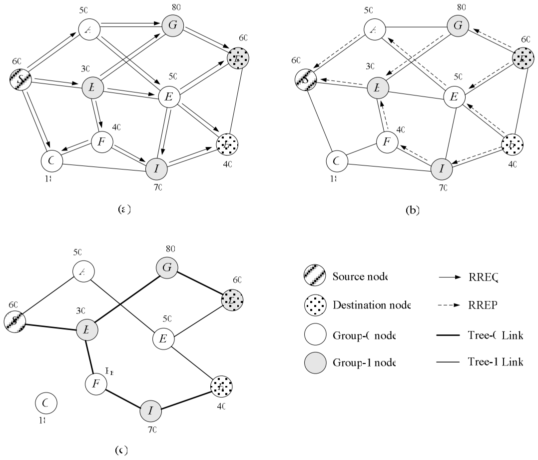

Fig. 3. Routing process. (a) The route discovery process. (b) The route reply process. (c) The multicast tree.

3.2. Route Maintenance Process

3.2.1. Multicast Join Operation

When a new member wants to join the multicast tree, it broadcasts a join request (RREQ_J) packet across the networks. Only a node that is a member of the multicast tree (i.e., a router for the group) may respond, if a node receives a RREQ_J packet for a multicast group of with it is not a member or it does not have a route to that group, it creates a reverse route entry to the prospective node and then broadcasts the RREQ_J packet to its neighbors. Any intermediate node receives the RREQ_J, it re-broadcasts the RREQ_J if the P_{remain} of the node is higher than $P_{threshold}$.

When each member node of the multicast tree receives the RREQ_J packets it sends back the join reply (RREP_J) packet with set class field. When each intermediate node receives the RREP_J packet, the intermediate node selects a downstream node based on the corresponding type of RREP_J packet. The prospective node selects the first RREP_J packet with group-0 and the first join reply with group-1 to join the multicast tree. Fig. 4 shows the node join operation.

In Fig. 4(a), the prospective node *J* broadcasts a RREQ_J packet to its neighbors, node *I* receives the packet, then node *I* puts the information into the RREQ_J packet and rebroadcasts RREQ_J packet until the multicast group member receives the packet. If P_{remain} of node *F* is 15 with lower than $P_{threshold}$, it discards these RREQ packet. The member nodes *B* and *E* reply with RREP_J packet to the prospective node, as shown in Fig. 4(b) The prospective node *J* selects the first RREP_J with group-0 and the first RREP_J with group-1 to join the multicast tree, as shown in Fig. 4(c).

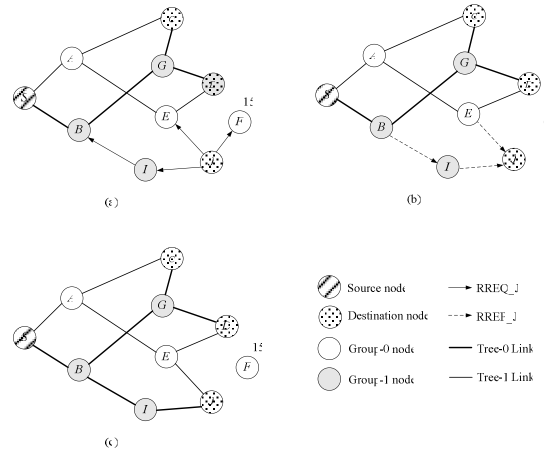

Fig. 4. Multicast join operation. (a) Join request packet propagation. (b) Join reply sent back to source. (c) Multicast tree branch addition.

3.2.2. Node Pruning Operation

When a node wants to move from the multicast tree, the pruning node broadcasts to its upstream node a prune request (RREQ_P) packet. When the upstream node receives the RREQ_P packet, the node removes the corresponding entry from its multicast routing table. If the upstream node becomes a leaf node and it is not the tree receiver, the node can further prune itself from the tree. An example of the pruning operation is shown in Fig. 5.

In Fig. 5(a), node *F* decides to leave the multicast tree, it broadcasts a RREQ_P packet. When nodes *E* and *I* receive the packet, they delete node *F* from its list for next hop. Then node *I* discovers that it is a leaf node and node *I* just a router for the multicast tree and not a multicast member. Node *I* sends a quit request packet to a node when node *B* deletes node *I* from the list of the next hop of node *B*. Fig. 5(b) illustrates the new multicast tree.

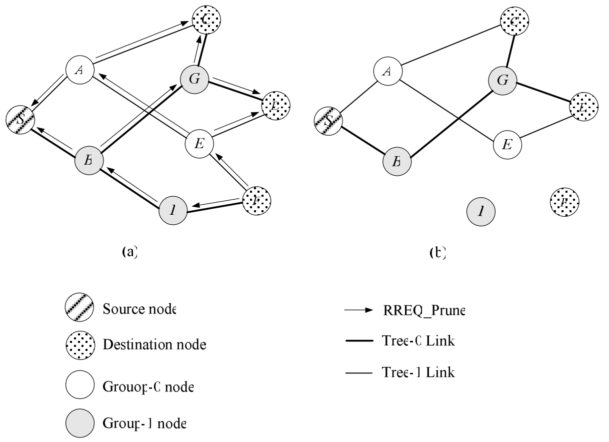

Fig. 5. Node pruning operation. (a) Prune request packet propagation. (b) Multicast tree after pruning.

3.2.3. Broken Link Maintenance

In PDTMRP, when a node fails to deliver the data packet to the next hop of the route, it considers the link to be broken and sends a route error (RERR) packet to the source node. When the upstream node receives the RERR packet, it removes the corresponding entry form its routing table and forwards the RERR packet to the source node. If only one of the two routes is broken, the source uses the remaining valid route to deliver data packets. When both routes of the path are broken, the source node initiates the route discovery process. The format of the RERR is shown in Fig. 6 and the route maintenance operation is shown in Algorithm 2.

Type RERR	Class	Destination ID	Source ID

Fig. 6. RERR packet format.

Algorithm 2: Route Maintenance Process

A network is modeled as graph $G = (N, E)$, where *N* is the finite set of mobile nodes and *E* is a set of links. Suppose *n* is the number of mobile nodes and *N* is the set of mobile nodes $N = \{N_1, N_2, ..., N_n\}$. Assume that node N_i

wants to send a packet to node N_j, where $N_i, N_j, \in N$, $1 \le i, j \le n$, $i \ne j$, and that the link between node N_i and node N_j breaks.

```
if (the link of node $N_i$ to node $N_j$ breaks) {
    (1) Node $N_i$ saves the current data packet.
    (2) Node $N_i$ broadcasts the RREQ_R packet to node $N_j$,
        counts down $T_{timeout}$ seconds, and waits for the
        RREP_R packet to return.
    if (the RREP_R packet is back in $Ttimeout$ seconds) {
        (1) Node $N_i$ uses the replacement path to replace the
            path that breaks.
        (2) Node $N_i$ continues packet transmission.
    }
    else
        Node $N_i$ sends an RERR packet to the source and
    restarts the routing discovery process.
}
```

4. Performance Evaluation

In this section, the simulation environment is described and the simulation results are presented.

4.1. Simulation Environment

A simulator with NS2 (Network Simulator 2) was designed and implemented to act as a simulation platform. The simulation modeled a network in a 1000 m×1000 m area with 50 mobile nodes. A random waypoint mobility model was used in the simulation. The mobile speed of each node was from 1 m/sec to 25 m/sec. The transmission range was 150 m. The data packet size was 250 bytes. Each simulation was executed for 600 seconds. The source and destination nodes were randomly chosen and each node was randomly assigned an initial energy. The parameters used in the simulations are listed in Table 1.

Table 1. Parameters used in the simulations

Parameter	Values
Examined protocols	PDTMRP, MAODV
Simulation area	1000 m×1000 m
Number of nodes	50
Mobility speed	1-25 m/sec
Mobility model	Random waypoint model
Node transmission range	150 m
Data packet size	250 bytes
Transmission power	2 joules
Receiver power	1 joules

The performance evaluation metrics used in the simulations were:

1. Packet delivery ratio: The data packets delivered divided by the data packets expected to be delivered.
2. Control overhead: The control packets transmitted divided by the data packets delivered.
3. Average remaining battery power: The average node's remaining battery power after data transmission.
4. Network lifetime: The duration of the network operation time until the first node failure due to battery depletion at the node.

4.2. Simulation Results

In the following, the impact of mobility speed on MAODV and the proposed PDTMRP is studied. Some simulations were conducted for packet delivery ratio, control overhead, average remaining battery power, and network lifetime.

Figs. 7, 8, 9, and 10 depict the routing performance of the proposed PDTMRP and MAODV protocols under different mobility speeds. The packet delivery ratio decreased with increasing mobility due to more link breaks. This resulted in more multicast tree partitions for the proposed PDTMRP and MAODV protocols. Notice that the number of packet delivery was high when the nodes had low mobility.

Fig. 7 shows the performance of the packet delivery ratio under various mobility speeds. As shown in Fig. 7, the PDTMRP protocol achieved a much higher packet delivery ratio then the MAODV protocol, because power is evaluated while establishing of two stable routing paths for multicasting. Thus, the packet delivery ratio of the proposed PDTMRP protocol was higher than that of MAODV.

Fig. 8 shows the performance of the control overhead under various mobility speeds. As was expected, the control overhead increased as the mobile nodes became more mobile. The reason was that there were more chances for routes to break when the speed of the mobile nodes was faster. Thus, the number of rebroadcasts increased. The proposed PDTMRP not only eliminated inefficient nodes to decrease the number of control packets, but also structured dual trees to reduce the number of route reconstructions. Therefore, the proposed PDTMRP had a lower control overhead than MAODV.

Fig. 9 shows the performance of the average remaining battery power under various mobility speeds. Due to the mobility of the node making the control overhead increase it consumed more power. Therefore, the average remaining battery power decreased with increasing mobility. As observed in Fig. 9, the average remaining battery power of the proposed PDTMRP was higher than that of MAODV. This is due to the proposed PDTMRP reducing the power consumption by using dual trees for transmission.

Fig. 10 shows the performance of the network lifetime at various mobility speeds. From Fig. 10, the network lifetime of the proposed PDTMRP outperformed that of MAODV. This was because the node residual battery power of the proposed PDTMRP was always higher than that of MAODV.

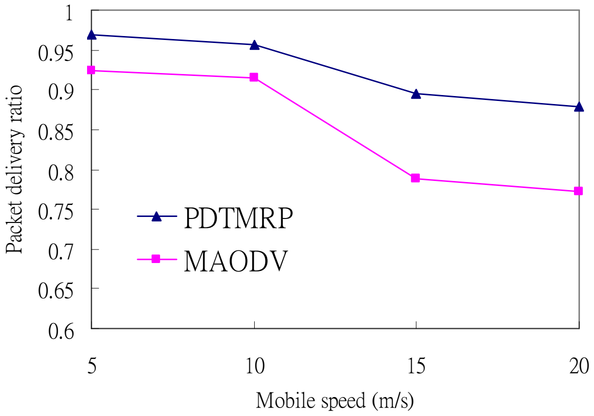

Fig. 7. Packet delivery ratio vs. mobility speed.

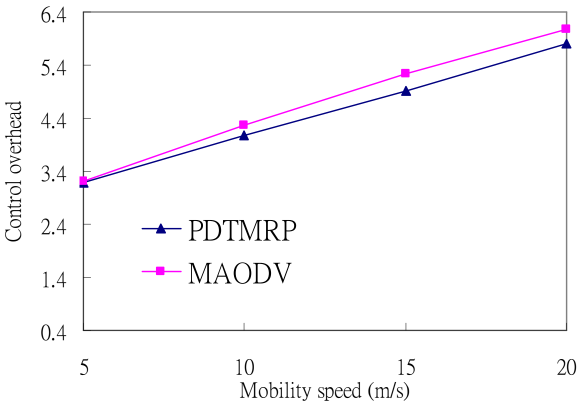

Fig. 8. Control overhead vs. mobility speed.

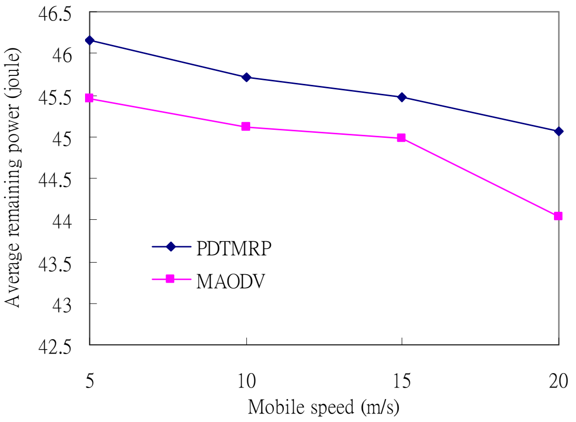

Fig. 9. Average remaining power vs. mobility speed.

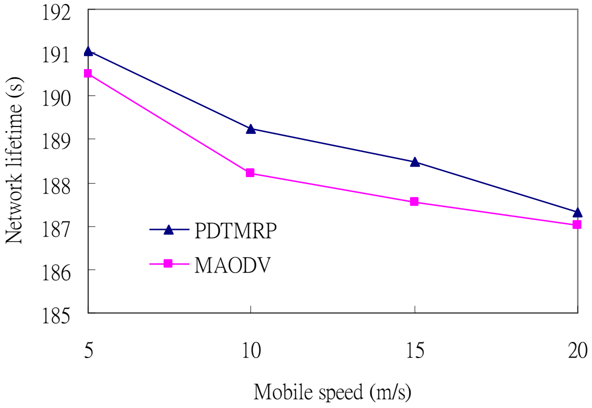

Fig. 10. Network lifetime vs. mobility speed.

5. Conclusions

In this paper, we propose a power-aware dual-tree-based multicast routing protocol (PDTMRP) for MANETs. In this scheme, load balance is used to improve the lifetime of a network. In the route discovery, this scheme not only solves the stability routing problem, but also achieves the load balance of data transmission. Therefore, the control overhead for route construction and the number of route reconstructions can be decreased. Simulation results show that the packet delivery ratio and the control overhead of the proposed scheme outperform that of MAODV. Moreover, the traffic load can be balanced and the network lifetime can be prolonged.

References

[1] A. Ballaradie, J. Crowcroft, and P. Francis, "Core Based Tree (CBT) - An Architecture for Scalable Inter-Domain Routing Protocol," *Proceedings of the 1993 ACM SIGCOM*, pp. 85-89, October 1993.

[2] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Ad Hoc Multicast Routing Protocol," *Internet Draft*, August 1998.

[3] K. L. Calvert and E. W. Zegura, "Core selection methods for multicast routing," *Technical Report*, GIT-CC-95/15, 1995.

[4] C. C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," *Cluster Computing*, Special Issue on Mobile Computing, Vol. 1, No. 2, pp. 187-196, December 1998.

[5] T. Clausen and P. Jacquet, "Optimized link state routing protocol. (OLSR)," *IETF Network Working*

Group Internet draft, *RFC 3626*, October. 2003.

[6] S. K. Das, B. S. Manoj, and C. S. R Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad hoc Wireless Network," *Proceeding of the third ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 24-35, June 2002.

[7] V. Devarapalli and D. Sidhu, "MZR: A Multicast Protocol for Mobile Ad Hoc Networks," *IETF MANET Working Group Internet Draft*, November 2000.

[8] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal of Selected Areas in Communications*, pp. 1380-1394, August 1999.

[9] C. Gui and P. Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," *Wireless Communication and Networking Conference*, 2003.

[10] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Kluwer Academic Publishers*, 1996.

[11] S. J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," *Proceedings of IEEE International conference on communications*, pp. 3201-3205, June 2001.

[12] J. P. Macker and M. S. Corson, "Mobile Ad Hoc Networking and the IETF," *ACM SIGMOBILE Mobile Computing and Communications Reviews*, Vol. 2, No. 2, pp. 9-14, January 1998.

[13] M. R. Pearlman, Z. J. Haas, P. Sholander and S. S. Tabrizi, “On the Impact of Alternate Routing for Load Balancing in Mobile Ad Hoc Networks,” *First Annual Workshop on Mobile and Ad Hoc Networking and Computing*, pp. 3-10, August 2000.

[14] C. E. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proceedings of the Second IEEE Workshop on Mobile Computing System and Application*, New Orleans, LA, USA, pp. 90-100, February 1999.

[15] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, 1995.

[16] E. M. Royer and C. E. Perkins, “Multicast Operation of the Ad Hoc On-Demand Distance Vector Routing Protocol,” *ACM MOBICOM*, pp. 207-18, August 1999.

[17] M. Sheng, H. Li and Y. Shi, “Delay Sensitive Adaptive Pouting Protocol for ad hoc network,” *Proceedings of the 17th International Conference on Advanced Information Networking and Applications*, pp. 731-736, March 2003.

[18] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core Extraction Distributed Ad-Hoc Routing," *Proceedings of the Wireless Communications and Networking Conference*, pp. 1313-1317, September 1999.

[19] W. Wei and A. Zakhor, “Multiple Tree Video Multicast Over Wireless Ad Hoc Networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 2-15, January. 2007.

[20] C. W. Wu, Y. C. Tay, and C. K. Toh, "Ad Hoc Multicast Routing Protocol Utilizing Increasing ID-Numbers (AMRIS) Functional Specification," *Internet Draft IETF*, November 1998.