

Cleveland State University  
**EngagedScholarship@CSU**



Electrical Engineering & Computer Science Faculty  
Publications

Electrical Engineering & Computer Science  
Department

11-2009

# Adaptive Multicast on Mobile Ad Hoc Networks Using Tree-Based Meshes With Variable Density of Redundant Paths

Sangman Moh

*Chosun University, smmoh@chosun.ac.kr*

Sang Jun Lee

*Chosun University*

Chansu Yu

*Cleveland State University, c.yu91@csuohio.edu*

Follow this and additional works at: [https://engagedscholarship.csuohio.edu/enece\\_facpub](https://engagedscholarship.csuohio.edu/enece_facpub)

 Part of the [Digital Communications and Networking Commons](#)

**How does access to this work benefit you? Let us know!**

*Publisher's Statement*

The final publication is available at Springer via <http://dx.doi.org/10.1007/s11276-008-0100-y>

## Original Citation

Moh, S., Lee, S. J., & Yu, C. (2009). Adaptive multicast on mobile ad hoc networks using tree-based meshes with variable density of redundant paths. *Wireless Networks*, 15(8), 1029 - 1041. doi:10.1007/s11276-008-0100-y

## Repository Citation

Moh, Sangman; Lee, Sang Jun; and Yu, Chansu, "Adaptive Multicast on Mobile Ad Hoc Networks Using Tree-Based Meshes With Variable Density of Redundant Paths" (2009). *Electrical Engineering & Computer Science Faculty Publications*. 44.

[https://engagedscholarship.csuohio.edu/enece\\_facpub/44](https://engagedscholarship.csuohio.edu/enece_facpub/44)

This Article is brought to you for free and open access by the Electrical Engineering & Computer Science Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering & Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact [library.es@csuohio.edu](mailto:library.es@csuohio.edu).

---

# Adaptive multicast on mobile ad hoc networks using tree-based meshes with variable density of redundant paths

Sangman Moh · Sang Jun Lee · Chansu Yu

**Abstract** Multicasting has been extensively studied for mobile ad hoc networks (MANETs) because it is fundamental to many ad hoc network applications requiring close collaboration of multiple nodes in a group. A general approach is to construct an overlay structure such as multicast tree or mesh and to deliver a multicast packet to multiple receivers over the overlay structure. However, it either incurs a lot of overhead (multicast mesh) or performs poorly in terms of delivery ratio (multicast tree). This paper proposes an adaptive multicast scheme, called *tree-based mesh with  $k$ -hop redundant paths* ( $TBM_k$ ), which constructs a multicast tree and adds some additional links/nodes to the multicast structure as needed to support redundancy. It is designed to make a prudent tradeoff between the overhead and the delivery efficiency by adaptively controlling the path redundancy depending on network traffic and mobility. In other words, when the network is unstable with high traffic and high mobility, a large  $k$  is chosen to provide more robust delivery of multicast packets. On the other hand, when the network traffic and the mobility are low, a small  $k$  is chosen to reduce the overhead. It is observed via simulation that  $TBM_k$  improves the *packet delivery ratio* as much as 35% compared to the multicast tree approach. On the other hand, it reduces control overhead by 23–87% depending on the value of  $k$  compared to the multicast mesh approach. In

general,  $TBM_k$  with the small value of  $k$  offers more robust delivery mechanism but demands less overhead than multicast trees and multicast meshes, respectively.

**Keywords** Mobile ad hoc network · Multicast · Distributed algorithm · Adaptive multicast · Tradeoffs

## 1 Introduction

A mobile ad hoc network (MANET) [1–3] is a collection of mobile nodes without any fixed infrastructure or any form of centralized administration. In such a network, each node is a data source/destination and at the same time as a router for multi-hop routing. MANETs and the corresponding principles can be effectively employed in military battlefields, emergency/disaster relief, and other applications including wireless sensor networks and wireless mesh networks.

*Multicasting* has been extensively studied for MANETs because it is fundamental to above-mentioned ad hoc network applications requiring close collaboration of multiple nodes in a group [4–6]. A multicast packet is delivered to multiple receivers along a network structure such as *multicast tree* [7–9]. It is preferable when network traffic is light and node mobility is low, possibly found in certain ad hoc network scenarios such as wireless sensor networks. However, this network structure is fragile due to node mobility and may not be able to deliver multicast packets to all multicast group members. It may generate a large volume of control messages in order to maintain a valid tree structure, particularly under highly dynamic environment [10]. Alternatively, *multicast mesh* [11–14] has been proposed in the literature. It is more robust to mobility than a multicast tree by virtue of redundant

---

S. Moh (✉) · S. J. Lee  
School of Computer Engineering, Chosun University,  
375 Seoseok-dong, Dong-gu, Gwangju 501-759, Korea  
e-mail: smmoh@chosun.ac.kr

C. Yu  
Department of Electrical and Computer Engineering, Cleveland  
State University, Stilwell Hall 437, Cleveland, OH 44115, USA  
e-mail: c.yu91@csuohio.edu

communication paths between mobile nodes in the mesh [4]. However, when node mobility is low, a rich structure in multicast mesh is not usefully utilized and even hurt the performance by overloading the network with redundant transmissions.

To make the comparison clearer, this paper considers two performance metrics, *control overhead* and *forwarding load* in addition to *packet delivery ratio* (PDR). The former measures the overhead corresponding to construction and maintenance of the multicast structure. The latter measures the quantity of data traffic caused by a single multicast data packet. Multicasting based on network flood incurs zero control overhead but generates a high forwarding load. In general, tree-based multicast incurs a lower control overhead and a lower forwarding load than mesh-based multicast but PDR is lower too. However, this is an oversimplification and performance assessment can vary depending on network status and traffic conditions. For example, a *demand-driven* tree-based scheme [10] could incur a high control overhead than multicast meshes when node mobility is high because of the vast amount of control messages upon link breaks. On the other hand, a mesh-based scheme could exhibit a lower PDR than multicast trees when network traffic is high [10, 15]. This is due to a large volume of redundant transmission or a high forwarding load, which leads to network congestion as discussed earlier. A single network structure cannot be a solution to every possible multicast scenario under a different network environment but a good solution must seek a balance between the overhead (control overhead and forwarding load) and delivery efficiency.

This paper proposes an adaptive multicast scheme, called *Tree-Based Mesh with k-hop redundant paths* ( $TBM_k$ ),<sup>1</sup> which allows a prudent choice between low-overhead multicast tree and highly efficient multicast mesh. More specifically, the proposed  $TBM_k$  constructs a multicast tree and adds some additional links/nodes to the tree structure as needed to support redundancy. Here, the path redundancy, measured in terms of connectivity among the multicast tree nodes, is controlled depending on the network condition such as network traffic, node connectivity and node mobility. In other words,  $TBM_k$  consists of a multicast tree and all  $k$ - or smaller-hop redundant paths between *tree nodes*. By definition, a multicast tree and network-wide flooding can be denoted as  $TBM_0$  and  $TBM_\infty$ , respectively. With high mobility, a large  $k$  is chosen to deliver multicast packets in a more robust way. On the other hand, when node mobility is low, a small  $k$  is chosen to reduce the overhead.

<sup>1</sup> In this paper,  $TBM_k$  refers not only to the proposed multicast algorithm but also to the network structure derived from the algorithm.

Contributions of this paper are two-fold:

- First, it provides new insights on multicast structures and thus helps develop more efficient multicast protocols for MANETs such as the proposed  $TBM_k$  scheme. For example, a multicast mesh is observed as a superposition of multiple multicast trees (see Sect. 2.1 for details). It starts with a tree, which is added by another tree and so on to eventually include all group members and thus to result in a richer structure with redundant paths. It finally converges to network flooding as the number of source nodes increases (Sect. 2.2 has more explanations). This observation motivated us to develop a more direct and efficient way of producing a rich structure by directly adding redundant links/nodes to the tree.
- Second, the redundancy in multicast meshes is in fact not controllable but is given by chance (see Sect. 3.3). In other words, a multicast mesh may not offer redundant paths at all, particularly when the number of sources is small [15]. On the other hand, it is controllable in the proposed  $TBM_k$  by adjusting the value of  $k$ . To the best of authors' knowledge, this is the first study in the literature that discusses the uncontrollable redundancy in most of mesh-based multicast schemes.

According to the simulation study, the proposed  $TBM_k$  improves PDR as much as 35% compared to  $TBM_0$  (multicast tree) within the range of simulation parameters we have tested. Compared to mesh-based multicast,  $TBM_k$  reduces control overhead by 76–87%, 38–73%, and 23–38% when  $k = 1, 2, \text{ and } 3$ , respectively. It also reduces forwarding load by 54–64%, 31–39%, and 23–34% when  $k = 1, 2, \text{ and } 3$ , respectively. In general,  $TBM_k$  with the small value of  $k$  (1, 2, or 3) offers more robust delivery mechanism than multicast trees but demands less control overhead and less forwarding load than multicast meshes.

The rest of the paper is organized as follows: Previous studies on multicast trees and multicast meshes are described in Sect. 2. Section 3 presents the proposed adaptive multicast scheme with algorithms and examples. Section 4 shows the effectiveness of the proposed scheme via simulation. Finally, concluding remarks are given in Sect. 5.

## 2 Related work

This section briefly overviews the previous multicast protocols developed for mobile networks. Basically, they construct a network structure to deliver multicast messages. However, to maintain the network structure in the presence of node mobility, multicast group members are supposed to exchange control packets (e.g., JOIN TREE or JOIN MESH messages) periodically to refresh the structure.

A demand-driven tree-based scheme has been proposed to cope with high mobility [9, 16]. Here, the tree structure is repaired whenever a link breakage is detected. On the contrary, multicast meshes are more robust thanks to redundant paths. They can deliver multicast messages in the presence of link breakages without immediately fixing the broken links. For this reason, mesh structures are typically constructed and maintained periodically in a proactive manner using periodic control messages (e.g., JOIN MESH messages).

## 2.1 Multicast trees

A *multicast tree* [4, 8] can be constructed and maintained using periodic JOIN TREE messages. Every member node periodically (e.g., every 3 s in [4]) sends a JOIN TREE message to the predetermined root node that is chosen among the member nodes. Then, the root can construct a multicast tree consisting of the paths that JOIN TREE messages traverse. There is only one path from the root node to each member node of the multicast group. Figure 1 shows an example of a multicast tree of 8-member multicast group constructed on a MANET. Note that every member node is a tree node but not every tree node is a member. For example, non-member nodes such as  $p$ ,  $q$ , and  $s$  are tree nodes because they are intermediate nodes along the paths that JOIN TREE messages traverse.

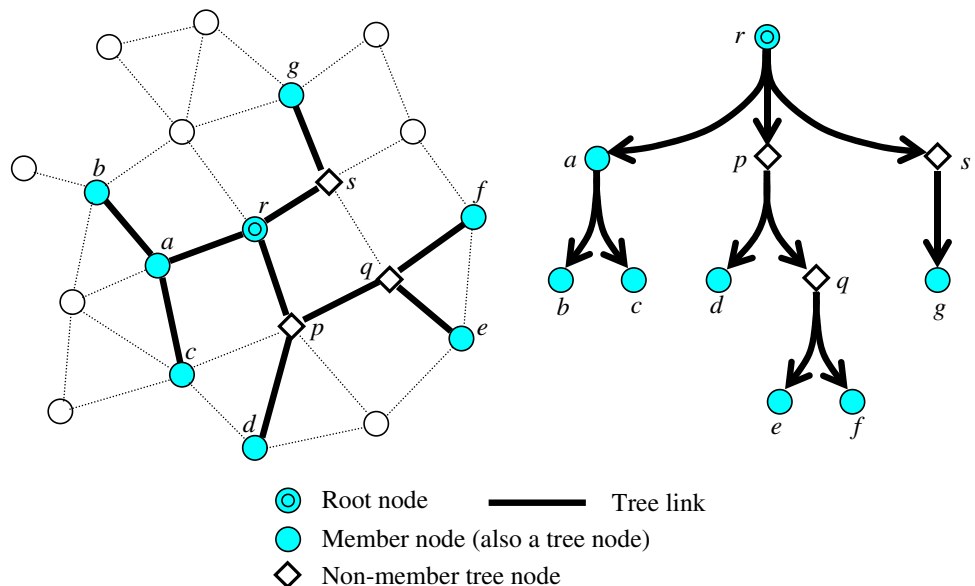
Since a tree is refreshed only every predetermined period, a low packet delivery ratio (PDR) is expected particularly when node mobility is high. Demand-driven multicast trees have recently been studied in order to

address this problem [9, 16]. A JOIN TREE message is sent whenever a downstream node detects a tree link breaks and the tree is repaired locally. For instance, when a tree link ( $a-b$ ) breaks in Fig. 1, the downstream node (node  $b$ ) sends a JOIN TREE message toward any tree node to reconnect to the main tree [9]. However, it requires that every node broadcasts a beacon message once in a while to allow its neighbors to monitor the link condition [9, 17].

Tree-based multicast can be further classified as either *per-source tree multicast* or *shared tree multicast* [18]. In the per-source tree approach, each source has to construct a separate multicast tree rooted at itself. Therefore, there will be as many trees as the number of sources and a significant amount of overhead is required when the number of sources is large. On the other hand, shared tree multicast has a lower overhead because it maintains only a single tree shared by all sources [8]. A multicast packet is (unicast) delivered to the root node first and then (multicast) delivered to all group members along the tree structure. The path is not necessarily optimal, and the root node is easily overloaded due to the sharing of the single tree.

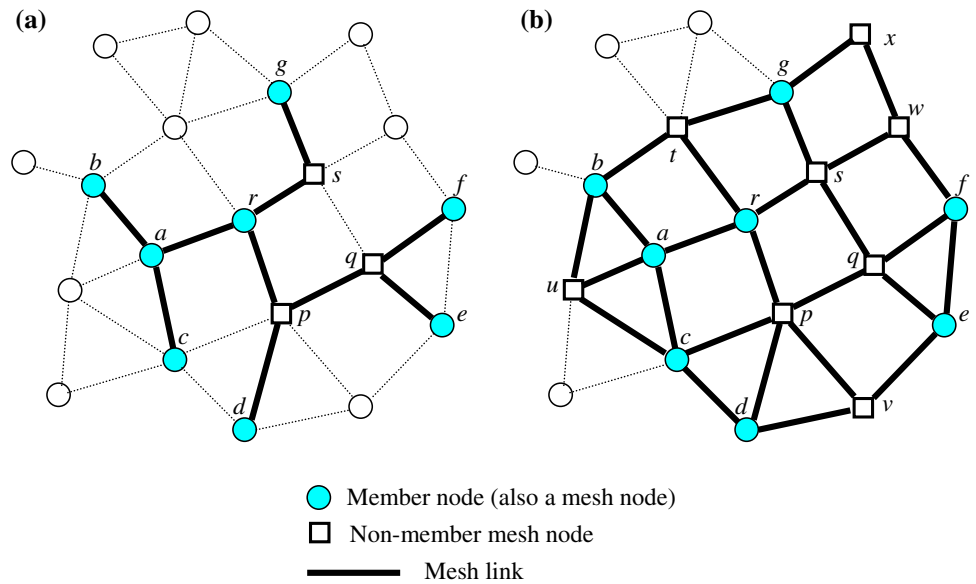
*Associativity-Based Multicast Routing Protocol* (ABAM) [19] and *Multicast Routing Protocol based on Zone Routing* (MZR) [7] are per-source type multicast protocols. *Ad hoc Multicast Routing* (AM-Route) [20], *Multicast Ad-hoc On-demand Distance Vector Protocol* (MAODV) [9], *Ad hoc Multicast Routing protocol utilizing Increasing id-numberS* (AMRIS) [21], *Lightweight Adaptive Multicast* (LAM) [22], and *Adaptive Demand-driven Multicast Routing* (ADMR) [16] are based on shared trees. The proposed  $TBM_k$  uses a shared tree as a fundamental structure to derive the proposed tree-based mesh.

**Fig. 1** An example of a multicast tree of 8 members





**Fig. 2** An example of a multicast mesh of 8 members. (a) Minimal multicast mesh (same as tree) (b) Maximal multicast mesh



## 2.2 Multicast meshes

The aforementioned tree-based protocols may not perform well under high node mobility because a multicast tree is fragile and needs to be reconfigured frequently as its connectivity changes. A *multicast mesh* [11–14] is different from a tree since each node in a mesh has multiple parents in terms of packet delivery functionality. A multicast mesh is constructed using JOIN MESH and JOIN REQUEST messages. For example, in *On-Demand Multicast Routing Protocol* (ODMRP) [11], a source node floods a JOIN REQUEST message periodically (e.g., every 3 s). Upon receiving a JOIN REQUEST message, every member node replies with a JOIN MESH message back to the source node. Non-member nodes recognize themselves as mesh nodes for the particular multicast group when they receive a JOIN MESH message.<sup>2</sup> Therefore, a multicast mesh is a superposition of multiple per-source trees as discussed in Introduction. While these per-source trees are constructed individually, a mesh-based multicast scheme utilizes them collectively to offer path redundancy.

A main difference between multicast trees and multicast meshes is that a node is supposed to receive a multicast message only from its parent in multicast trees while it can receive the message from any member or non-member mesh node in multicast meshes. Therefore, in a mesh, when a link is broken due to node mobility, other links can be immediately available improving the delivery capability of multicast messages. Correspondingly, it reduces the

overhead to reconstruct and maintain the network structure frequently. Note that network-wide flooding is an extreme case of multicast meshes, where all the nodes in a MANET participate in forwarding multicast messages. No reconfiguration is required and yet the delivery ratio is high. However, forwarding load, explained in Sect. 1, would be high enough to make this scheme impractical in most of MANET scenarios.

Figure 2 shows two multicast meshes for the same 8-member multicast group as in Fig. 1. It is assumed that every member node is a source. In other words, every member node has multicast messages to send to every other member node, which is typically the case in many multicast scenarios requiring collaboration among the nodes in the group. It is important to note that JOIN REQUEST is a broadcast message and traverses unreliably and unpredictably. Therefore, the final mesh can be in between the two extreme configurations in Fig. 2(a), (b). Note also that a multicast mesh is a superposition of eight source-rooted trees, or equivalently, eight member-rooted trees in this example. Figure 2(a) is the minimal mesh when the eight source-rooted trees are identical. On the other hand, when the eight source-rooted trees are constructed as disjoint as possible, it results in the maximal multicast mesh as shown in Fig. 2(b).

Note that every member node is a mesh node but not every mesh node is a member. Intermediate nodes  $p$ ,  $q$ , and  $s$  in Fig. 2(a), and  $p$ ,  $q$ ,  $s$ ,  $t$ ,  $u$ ,  $v$ ,  $w$ , and  $x$  in Fig. 2(b) are included in the two meshes, respectively, and participate in forwarding multicast messages. The minimal mesh in Fig. 2(a) may not be desirable because it does not offer many redundant paths. On the other hand, the maximal mesh in Fig. 2(b) may not be desirable either because

<sup>2</sup> In ODMRP [11], the JOIN MESH message and the mesh node are called JOIN TABLE and *forwarding node*, respectively. And, the set of mesh nodes is called *forwarding group*.

redundant links are not always useful while they incur high forwarding load. Therefore, it is important to be capable of adjusting the level of redundancy depending on the network status. However, most of mesh-based multicast schemes construct meshes by chance and there exists no control over the level of redundancy. This is in fact the main theme of this paper.

Protocols such as ODMRP [11], *Neighbor-Supporting Multicast Protocol* (NSMP) [12], *Core-Assisted Multicast Protocol* (CAMP) [23], *Multicast Core-Extraction Distribution Ad hoc Routing* (MCEDAR) [24], and *Clustered Group Multicast* (CGM) [25] are multicast mesh schemes.

### 3 Tree-based meshes with $k$ -hop redundant paths

This section proposes a multicast scheme, called *Tree-Based Mesh with  $k$ -hop redundant paths* ( $TBM_k$ ), which can adaptively adjust the level of redundancy to strike the balance between the overhead and the delivery efficiency as well as between the control overhead and forwarding load. Section 3.1 summarizes terminologies used in this paper and provides examples of  $TBM_k$ . Section 3.2 presents the  $TBM_k$  algorithm. Section 3.3 offers a qualitative analysis of  $TBM_k$  in comparison to tree-based and mesh-based multicast algorithms.

#### 3.1 Tree-based meshes

A multicast tree with  $n$  nodes has exactly  $n - 1$  links, which essentially means that there exists only one communication link for each tree node to receive from. A link breakage directly translates to a communication failure to the corresponding tree node as well as all its offspring nodes. Adding one or more links to a multicast tree results in cycles in the graph derived from the tree and this corresponds to the creation of redundant paths among the tree

nodes. We call this network structure as *tree-based multicast mesh*, which is defined as follows.

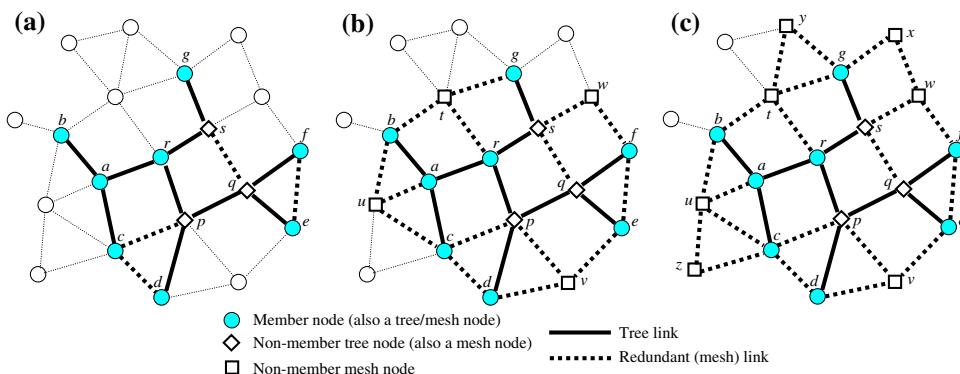
**Definition 1** For a given multicast tree, a  *$k$ -hop redundant path* is a path of length  $k$  ( $k$ -hop path), two end nodes of which are tree nodes and other  $k - 1$  nodes are not.

**Definition 2** For a given multicast tree, a *tree-based mesh with  $k$ -hop redundant paths* ( $TBM_k$ ) is a combination of a multicast tree and all its 1-, 2-, ..., and  $k$ -hop redundant paths.

For the same 8-member multicast group in Figs. 1 and 2, Fig. 3 shows the corresponding  $TBM_k$ . Figure 3(a) shows  $TBM_1$ . Compared to  $TBM_0$  (which is the same as the multicast tree in Fig. 1), it includes four 1-hop redundant paths,  $(s, q)$ ,  $(p, c)$ ,  $(c, d)$ , and  $(e, f)$ , but no additional mesh nodes. Considering the broadcast nature of wireless communication, one may wonder how  $TBM_1$  differs from  $TBM_0$  and improves the delivery capability. However, consider the case when link  $(p, q)$  breaks. In  $TBM_0$ , node  $q$  is supposed to receive the multicast message from node  $p$  and thus node  $q$  as well as node  $e$  and  $f$  are unable to receive the message. On the other hand, in  $TBM_1$ , node  $q$  is supposed to receive the message from any tree or mesh nodes in  $TBM_1$  and therefore, it can still receive the message from node  $s$  or node  $r$  directly if node  $q$  moves closer to it. In other words, the aforementioned four redundant paths can be effectively used in case tree links are broken.

Figure 3(b) shows  $TBM_2$ . In addition to four 1-hop redundant paths of  $TBM_1$ ,  $TBM_2$  includes ten 2-hop redundant paths such as  $(s, w, f)$ ,  $(r, t, b)$ ,  $(r, t, g)$ ,  $(b, t, g)$ ,  $(a, u, b)$ ,  $(a, u, c)$ ,  $(b, u, c)$ ,  $(p, v, d)$ ,  $(p, v, e)$ , and  $(d, v, e)$ , resulting in more robust and reliable delivery. It has four more nodes ( $t, u, v$ , and  $w$ ) participating in the mesh compared to  $TBM_1$ . Figure 3(c) shows  $TBM_3$ . In addition to four 1-hop and ten 2-hop redundant paths, there are six 3-hop redundant paths such as  $(g, x, w, s)$ ,  $(g, x, w, f)$ ,  $(b, t, y, g)$ ,  $(r, t, y, g)$ ,  $(b, u, z, c)$ , and  $(a, u, z, c)$ , which allow

**Fig. 3** A tree-based mesh with 1-, 2-, and 3-hop redundant paths ( $TBM_k$ ). (a)  $TBM_1$ , (b)  $TBM_2$ , (c)  $TBM_3$



even more robust and reliable delivery of multicast packets compared to  $TBM_2$ . It includes three more mesh nodes ( $x$ ,  $y$ , and  $z$ ) compared to  $TBM_2$ . It is not difficult to deduce that  $TBM_4$  has one more mesh node (node  $A$ ) than  $TBM_3$  and  $TBM_k = TBM_4$  for all  $k > 4$ .

A main difference between  $TBM_k$  in Fig. 3 and the multicast mesh in Fig. 2 is that the path redundancy in  $TBM_k$  is controllable by choosing an appropriate value of  $k$  but that in multicast mesh is not. As discussed earlier in Sect. 2.2, the configuration of the mesh is determined by chance in between the minimal and the maximal mesh drawn in Fig. 2(a), (b), respectively.

### 3.2 Algorithm for $TBM_k$ construction

Construction of  $TBM_k$  follows the following 3-step process: (i) Construction of a multicast tree using a JOIN TREE message, (ii) choice of  $k$ , and (iii) construction of  $k$ -hop redundant paths using JOIN  $k$ -HOP messages.

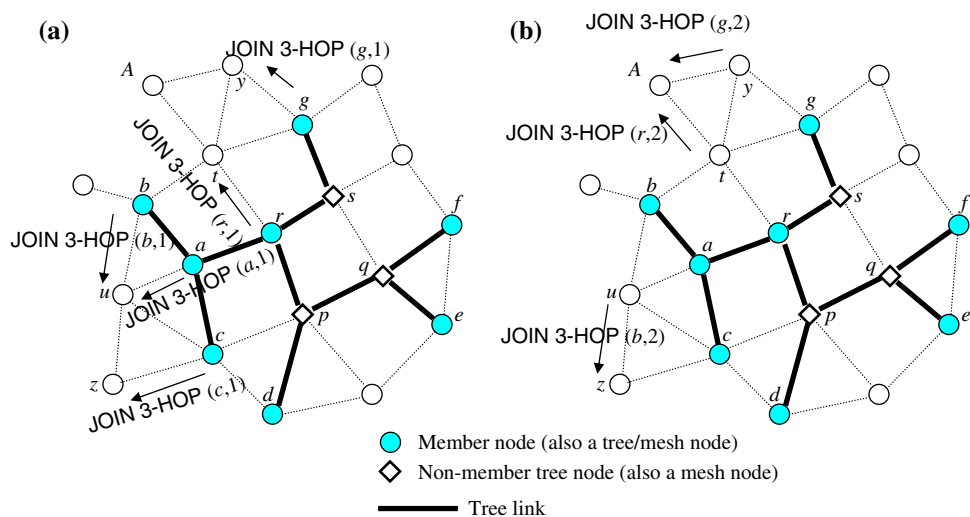
First, a multicast tree is constructed based on JOIN TREE messages as described in Sect. 2.1. A JOIN TREE message is sent periodically by multicast members to a predetermined root node. Unlike in multicast trees, the choice of the root node is not important in  $TBM_k$  because it constructs a mesh afterwards and the root does not play an important role in the mesh. As discussed in Sect. 2, some recent tree-based multicast schemes [9, 16] prefer to repair the tree on-demand whenever a tree link breaks and keep a valid tree structure all the time. However, since  $TBM_k$  appends the tree with redundant paths, the reactive repair does not bring a significant performance improvement while increasing the control overhead. Therefore, the tree construction in  $TBM_k$  is done in proactive manner via periodic JOIN TREE messages like multicast meshes. As

in most of multicast tree algorithms, JOIN TREE messages can be routed using an underlying unicast routing protocol.

Second, after constructing a tree, the root node chooses the optimal value of  $k$  depending on network status. It can be measured in terms of link layer parameters (such as collision ratio and node connectivity) or network layer parameters (such as route discovery frequency). We do not discuss this issue any further in this paper and leave it as an important future work. However, it is important to note that the root node does not necessarily transmit the value of  $k$  every time the tree is refreshed. Since the network status does not change abruptly over time, the same value of  $k$  could be used during the lifetime of a multicast group.

Third, each tree node constructs  $k$ -hop redundant paths by broadcasting a JOIN  $k$ -HOP message with TTL (Time To Live) of  $k$ . This message also includes three more parameters, multicast group id, tree node id that initiates the JOIN  $k$ -HOP message and the number of nodes it already traversed. A non-member, non-tree node that receives this message considers itself a mesh node for this particular multicast group if it can connect two tree nodes within  $k$ -hop. For example, Fig. 4 shows the process of constructing a  $TBM_3$  with JOIN 3-HOP messages. It is shown in Fig. 4(a) that tree nodes  $a$ ,  $b$ ,  $c$ ,  $g$ , and  $r$  broadcast JOIN 3-HOP messages. They are forwarded by, for example, nodes  $t$ ,  $u$ , and  $y$  toward nodes  $z$  and  $A$  as in Fig. 4(b). The message from node  $g$  in Fig. 4(a) is JOIN 3-HOP ( $g, 1$ ), which denotes that the tree node  $g$  initiated the message and only one node (node  $g$ ) has been traversed so far. Node  $u$  determines itself as a mesh node because it receives two JOIN 3-HOP messages from two different tree nodes (nodes  $a$  and  $b$ ) and the combined hop count is less than 3. Node  $z$  also determines itself as a mesh node because it receives two JOIN 3-HOP messages from two different tree nodes (nodes  $c$  and  $u$ ) and the combined hop

**Fig. 4** Construction of 3-hop redundant paths. (Multicast group id is not shown for simplicity.) (a) JOIN 3-HOP messages are initiated (b) JOIN 3-HOP messages are forwarded



count is equal to 3. However, node  $A$  in Fig. 4(b) does not elect itself as a mesh node because it receives two JOIN 3-HOP messages from node  $y$  and  $t$  but the combined hop count is larger than 3.

Figure 5 describes the detailed steps of the algorithm. Here, JOIN  $k$ -HOP message includes three parameters in addition to  $k$  as described earlier. JOIN TREE message include two parameters, multicast group id and root node. A JOIN TREE message (line 2 in Fig. 5) is routed using an underlying unicast routing protocol, and a node becomes a tree node when it receives this message (line 3). It then forwards the message toward the root if it is not the root (line 4). Tree nodes are supposed to initiate JOIN  $k$ -HOP messages once the optimal value of  $k$  is known (line 6). Note that when  $k = 0$ , no JOIN  $k$ -HOP message is sent because  $TBM_0$  is the same as the tree and is already constructed. When  $k = 1$ , no JOIN  $k$ -HOP message is sent either because  $TBM_1$  adds no extra mesh node as described earlier. However, each tree node considers itself as a mesh node and the delivery of a multicast message is not limited along the tree.

Once a non-tree node receives two JOIN  $k$ -HOP messages from two different tree nodes (say, nodes  $a$  and  $b$ ) of the same multicast group, it evaluates its eligibility as a mesh node for the particular multicast group. The

evaluation is based on the sum of hop counts in the two JOIN  $k$ -HOP messages. If it is less than or equal to  $k$ , the node becomes a mesh node (line 12) because there exists a path of length  $k$  or less between two tree nodes ( $a$  and  $b$ ). For this purpose, a non-tree node keeps track of the shortest hop count ( $h_{\min}$ ) to any tree node of the multicast group (line 13). A non-tree node is supposed to forward the JOIN  $k$ -HOP message after incrementing the hop count unless the TTL value of the message expires (line 15).

Lines 16–32 in Fig. 5 shows the steps upon receiving a multicast data message,  $m(t)$ , where  $t$  is the source. If  $k = 0$  for the corresponding multicast group, it is equivalent to a multicast tree ( $TBM_0$ ) and thus the multicast data is forwarded along the tree (line 20–26). However, if  $k > 0$ , it allows redundant paths and thus each mesh node is supposed to forward the multicast message whenever it receives one (line 31).

### 3.3 Complexity analysis

This subsection discusses the complexity and sensitivity of the  $TBM_k$  algorithm in comparison to tree- and mesh-based multicast protocols in terms of control overhead (message complexity) and forwarding load. It is noted that there have

**Fig. 5** The  $TBM_k$  algorithm. (Multicast group id is not shown for simplicity)

```

//  $TBM_k$  (Tree-Based Mesh with  $k$ -hop redundant paths) at node  $l$ 
1:  $h_{\min} = \infty$ ; // keep track of the least hop count to a tree node
2: Upon receiving a JOIN TREE ( $r$ ) message with root  $r$  for a particular multicast group (unicast)
3: mark itself as a tree node of the multicast group;
4: if ( $l \neq r$ ) forward the message to the next hop node toward  $r$ ;
5: // wait until the root informs the optimal value of  $k$ 
6: if ( $k > 1$ ) send a JOIN  $k$ -HOP ( $l, 1$ ) message; // no JOIN  $k$ -HOP message is sent when  $k = 0$  or 1
7: Upon receiving a JOIN  $k$ -HOP ( $t, h$ ) message originated from node  $t$  with the hop count  $h$ 
8: if ( $l$  is a tree node) return;
9: if (recently received the similar message from node  $t$ ) return; // it came through a different path
10: // since it can reach a tree node within  $h_{\min}$  hop count and can reach another tree node within  $h$ ,
11: // the two tree nodes can communicate through node  $l$  within  $k$  hops if  $h_{\min} + h \leq k$ 
12: if ( $h_{\min} + h \leq k$ ) mark itself as a mesh node of the multicast group;
13: if ( $h_{\min} > h$ )  $h_{\min} = h$ ; // update  $h_{\min}$ 
14: // decrement TTL and forward the  $k$ -hop redundant path message if TTL > 0
15: if ( $h + 1 < k$ ) forward JOIN  $k$ -HOP ( $t, h + 1$ ) message;
16: Upon receiving a multicast data  $m(t)$  message from node  $t$  for the multicast group
17: if ( $l$  is not a mesh node) return;
18: // when  $k = 0$  for the multicast group, multicast tree with no redundant nodes/links
19: // a multicast message would be delivered only from a parent
20: if ( $k = 0$ ) {
21:     if ( $t$  is  $l$ 's parent) {
22:         process the multicast message;
23:         forward  $m(t)$  to its children;
24:     }
25:     else return;
26: }
27: // when  $k > 0$  for the multicast group, tree-based mesh
28: // a multicast message would be broadcasted to the rest of the mesh
29: else {
30:     process the multicast message;
31:     forward  $m(t)$  to the rest of the mesh;
32: }

```



been very few works reported in the literature that analyzes the multicast algorithm for MANETs [26, 27]. Instead, most of previous studies resort to simulations to assess the performance of multicast protocols. In the following, we do not intend to provide a thorough complexity analysis but offer a qualitative comparison among tree-based, mesh-based, and the proposed  $TBM_k$  multicast schemes.

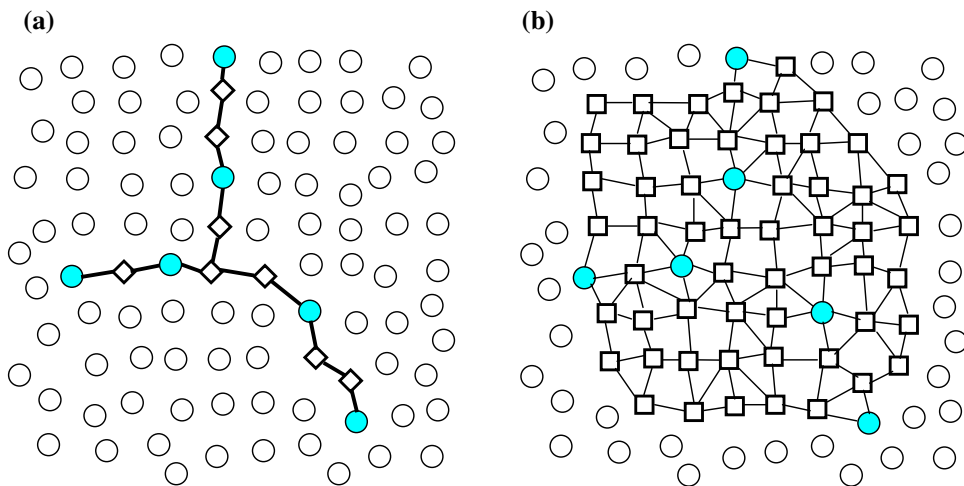
First of all, control overhead denotes the number of control messages spent to construct and maintain the multicast structure as described earlier in Sect. 1. For a multicast group of  $n$  members, a tree-based multicast scheme generates  $(n - 1)$  JOIN TREE messages (unicast) resulting in very low control overhead regardless the node mobility. However, when node mobility is moderate to high, tree-based schemes suffer in terms of packet delivery functionality. Demand-driven, multicast tree-based schemes such as MAODV [9] or ADMR [16] address this problem by repairing the links whenever they are detected broken (explained in Sect. 2.1), thus increasing the control overhead. In fact, it is reported via simulation that the control overhead of such schemes is much higher than that of mesh-based schemes such as ODMRP [11] when the average node speed exceeds 10 m/s [10]. A mesh-based multicast scheme generates  $n$  JOIN REQUEST messages (broadcast) flooded into the network and  $n(n - 1)$  JOIN MESH messages (unicast) toward group members per period. Thanks to redundant paths, mesh-based protocols overcome the link breaks during any two consecutive periods and maintain a reasonable packet delivery ratio. However, they are usually far more expensive than tree-based schemes in terms of control overhead. In  $TBM_k$ ,  $(n - 1)$  JOIN TREE messages (unicast) and  $m$  JOIN  $k$ -HOP messages (broadcast, but with TTL of  $k$ ) are initiated per period by members to construct a tree and by tree nodes to construct redundant path, respectively, where  $m$  denotes the number of tree nodes ( $m \geq n$ ). It is larger than

tree-based schemes but just a fraction of mesh-based schemes.

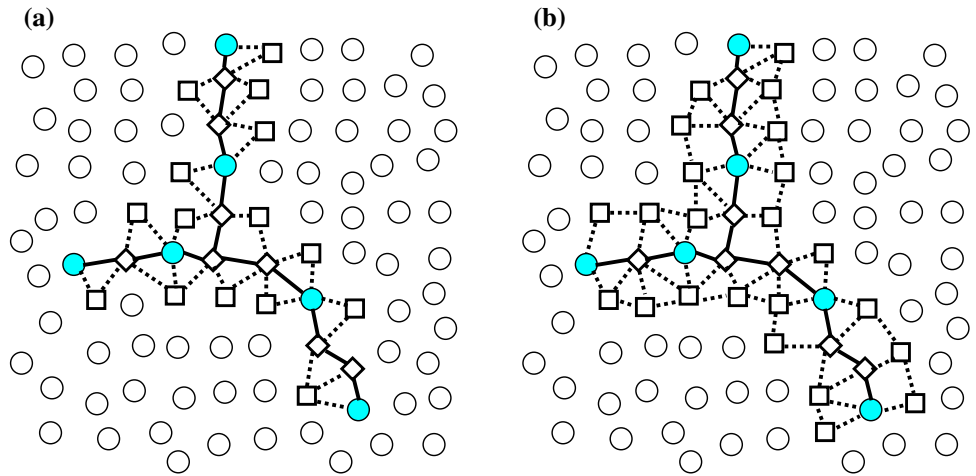
Second, forwarding load measures the quantity of data traffic caused by a single multicast data packet as explained in Sect. 1. It is equivalent to the number of participating nodes in the multicast structure. In tree-based schemes, it is  $(m - 1)$  or  $O(m)$  because there are  $m$  nodes participating in the tree for a multicast group of  $n$  nodes and  $m$  tree nodes ( $m \geq n$ ). On the other hand, forwarding load of mesh-based schemes such as ODMRP [11] is not larger than  $(nm - 1)$  or  $O(m^2)$  because a mesh is a superposition of  $n$  trees, each of which has  $m$  tree nodes, and those trees are disjoint with each other in the worst case. Since there is a total of  $nm$  participating nodes in the mesh and  $m \geq n$ , its complexity becomes  $O(m^2)$ .

Forwarding load analysis of  $TBM_k$  is not straightforward because its complexity depends on node density and the value of  $k$ . In  $TBM_k$ , a redundant path is a  $k$ -hop path between two tree nodes, and thus a participating node must be located within  $k/2$  times of the transmit range of a tree node. If node density or  $k$  is high,  $TBM_k$  could generate a higher forwarding load. To compare the three multicast schemes as well as to explore the effect of node density on forwarding load (number of participating nodes), Fig. 6 shows the tree and mesh structure for 6-member multicast group. There are 8 non-member tree nodes in Fig. 6(a), totaling 14 tree nodes in the multicast tree. In multicast mesh in Fig. 6(b), there are 54 non-member mesh nodes, totaling 60 mesh nodes. Forwarding load is more than four times higher in multicast mesh than in multicast tree. On the other hand, there are as four times higher path redundancy in multicast mesh, which is beneficial in dynamic environment. Note that a mesh is a superposition of source-rooted trees as mentioned in Sect. 2.2. Figure 6(b) draws the maximal possible mesh while Fig. 6(a) is in fact a minimal mesh. As discussed earlier, a main concern in

**Fig. 6** A multicast tree and a mesh for a 6-member multicast group. (a) Multicast tree (14 tree nodes). (b) Multicast mesh (60 mesh nodes). (Forwarding load, or equivalently the number of forwarding nodes, is more than four times higher in (b). It is possible to view figures in (a) and (b) as the minimal and maximal multicast mesh, respectively)



**Fig. 7**  $TBM_k$  for 6-member multicast group. (a)  $TBM_2$  (29 participating nodes). (b)  $TBM_3$  (37 participating nodes). ( $TBM_2$  and  $TBM_3$  reduce forwarding load by 52% and 38%, respectively, compared to the multicast mesh in Fig. 6b)



mesh-based schemes is that the redundancy is not controllable but is given by chance.

In contrast,  $TBM_k$  adaptively controls the density and redundancy of the mesh to be constructed. Figure 7 shows  $TBM_2$  and  $TBM_3$  of the same 6-member multicast group on the same example MANET. There are 8 non-member tree nodes in both  $TBM_2$  and  $TBM_3$  in Fig. 7(a), (b). And, there are 15 and 23 non-member mesh nodes in  $TBM_2$  (Fig. 7(b)) and  $TBM_3$  (Fig. 7(b)), respectively. Altogether, the number of participating nodes in the  $TBM_2$  and  $TBM_3$  are 29 and 37 nodes, which is 48% and 62% of the (maximal) multicast mesh in Fig. 6(b). Therefore, forwarding load of  $TBM_k$  is lower than multicast mesh, contributing to a less congested network. Therefore, it is not unreasonable to conclude that  $TBM_k$  causes less forwarding load than mesh-based schemes.

## 4 Performance evaluation

This section compares the performance of  $TBM_k$  ( $k = 1, 2$ , and 3) with that of simple flooding-based ( $TBM_\infty$ ), mesh-based and tree-based multicast ( $TBM_0$ ) via simulation. It is expected that  $TBM_\infty$  shows the best performance in terms of packet delivery ratio and thus provides the upper bound performance. On the other hand,  $TBM_0$  would show the least control overhead. The simulation environment is described in Sect. 4.1 including network model, node mobility and multicast traffic. Section 4.2 presents and discusses the simulation results.

### 4.1 Simulation environment

Our evaluation is based on the simulation of 70 mobile nodes moving over a square area of  $1,000 \text{ m} \times 1,000 \text{ m}$

for 900 s of simulation time. The transmission range is assumed to be 200 m and a free space propagation channel is assumed with a data transmission rate of 2 Mbps. Omni-directional antennas and symmetric radio links are assumed in conjunction with the same transmission power. Mobile nodes are assumed to move randomly according to the *random waypoint model* [4, 8]. In this mobility model, maximum node speed and pause time determine the mobility pattern of the mobile nodes. Each node starts its journey from a randomly selected location to a target location, which is also selected randomly in the simulated area. Node speed is randomly chosen between 0 and the specified maximum speed. When a node reaches the target location, it stays there for the pause time of 100 s and then repeats the same mobility behavior.

*Group size* (i.e., the number of member nodes in a multicast group) is varied in a meaningful range from 5 to 40 in the 70-node network. In order to evaluate the effect of node mobility, the maximum node speed is varied from 5 to 20 m/s. In our simulation, one constant bit rate (CBR) source and its multiple destinations (group size) are randomly selected among the mobile nodes. The CBR source sends 512-byte multicast packets during the simulation. In order to observe the performance trend based on different traffic, the packet rate is varied from 2 to 50 packets per second.

### 4.2 Simulation results and discussion

Performance metrics are *packet delivery ratio*, *forwarding load* and *control overhead* as described earlier in Sect. 1. Packet delivery ratio is the ratio of the number of multicast data packets successfully delivered to the destination over the number of multicast data packets sent by the source. Forwarding load measures how many times a multicast

packet is forwarded. In the worst case, every node in the network forwards a multicast packet (forwarding load is equal to the total number of nodes), which is the case with the flooding. Control overhead is the total number of control packets such as JOIN TREE and JOIN  $k$ -HOP messages transmitted per multicast packet. Each hop-wise transmission of a forwarding or control packet is counted as one transmission in measuring forwarding load and control overhead. Multicast tree is expected to exhibit the lowest control overhead and the lowest forwarding load but packet delivery ratio will be the lowest too. Flooding would be exactly the opposite and  $TBM_k$  makes a tradeoff between the two.

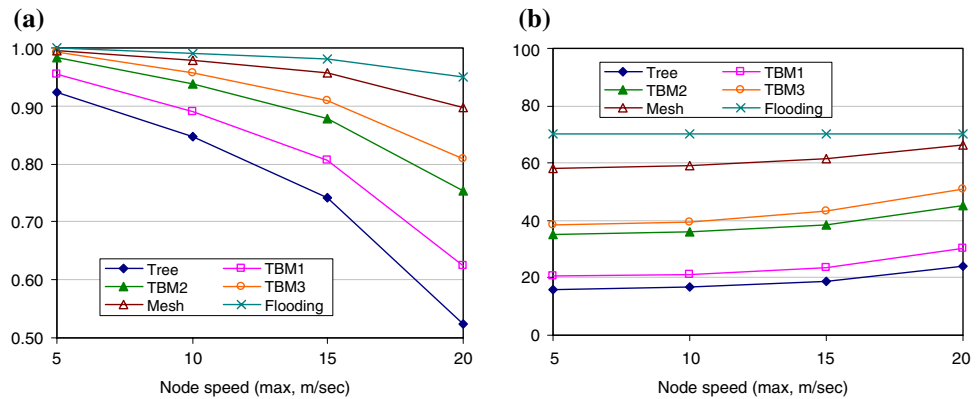
Figure 8(a), (b) compare packet delivery ratio and forwarding load of the  $TBM_k$  ( $k = 1, 2, \text{ and } 3$ ), flooding-based multicast ( $TBM_\infty$ ), mesh-based multicast and tree-based multicast ( $TBM_0$ ) with respect to node speed.  $TBM_k$  is better than the multicast tree in terms of packet delivery ratio and it performs better than the flooding as well as the multicast mesh in terms of forwarding load as shown in Fig. 8(a), (b), respectively. It is observed from Fig. 8(a) that the packet delivery ratio of  $TBM_k$  is improved with larger  $k$ . It is obvious that  $TBM_k$  enables prudent tradeoffs between multicast tree and flooding in terms of packet delivery ratio.

In Fig. 8(b), the forwarding load is always 70 (total number of nodes in the network) in case of flooding. It is

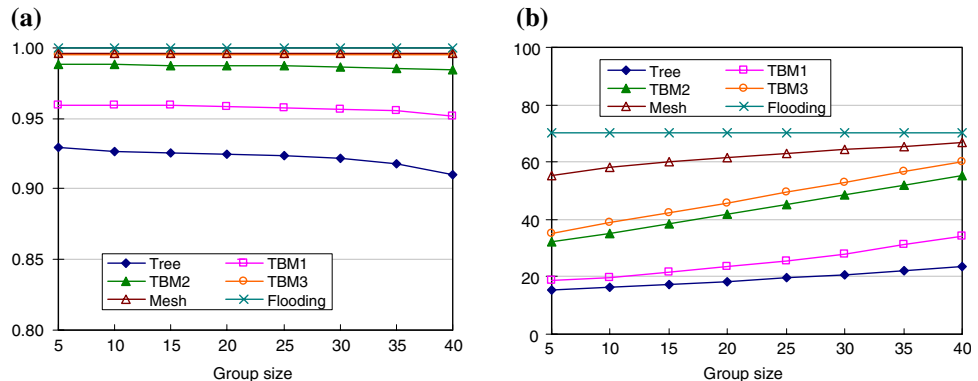
quite high for the multicast mesh but is significantly reduced with  $TBM_k$  ( $k = 1, 2, \text{ and } 3$ ) as shown in the figure. Note that the forwarding load increases slightly as node speed increases (except for flooding). This is because there are more link breakages in a more unstable network and thus multicast messages are forwarded more frequently (retransmission) with a high node speed. Notice the performance gap between the tree and other schemes in Fig. 8(b). The gap is due to redundant transmissions because the tree generates the smallest number of forwarding of a multicast message. Figure 8 shows that the acceptable value of  $k$  is 2, which is enough to achieve fairly high packet delivery ratio while incurring 31–39% less forwarding load than the mesh-based scheme.

Figure 9 shows performance variation with respect to multicast group size. The maximum node speed of 5 m/s is used. It is observed that  $TBM_k$  shows the significant improvement over the multicast tree as evident in Fig. 9(a). Another interesting observation in Fig. 9(a) is that the packet delivery ratio is slightly decreased with the increased group size. As the group size increases, a multicast packet is delivered to more number of receivers and a link breakage affects larger number of receivers. These factors contribute more in a negative way while the added redundancy contributes in a positive way. As  $k$  increases, the positive effect cancels out the negative effect and thus the packet delivery

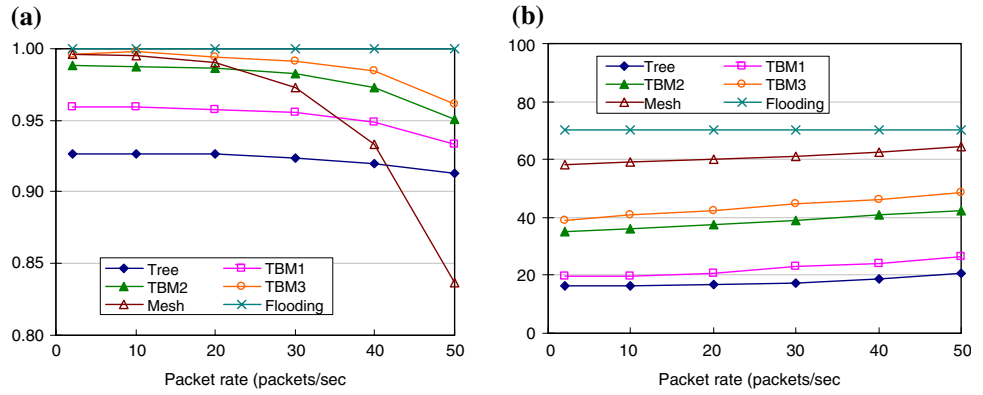
**Fig. 8** Performance comparison with different node mobility. (a) Packet delivery ratio (b) Forwarding load (Group size: 10, Packet rate: 2 pkts/s)



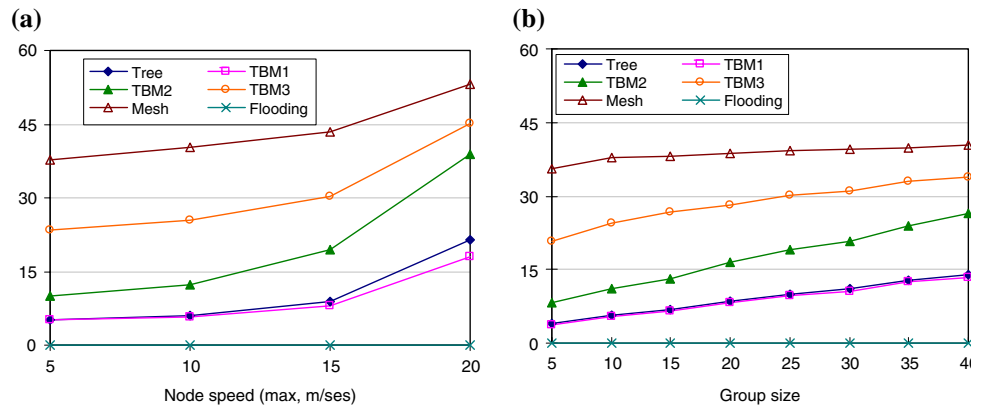
**Fig. 9** Performance comparison with different group size. (a) Packet delivery ratio (b) Forwarding load (Max. node speed: 5 m/s, Packet rate: 2 pkts/s)



**Fig. 10** Performance comparison with different traffic. (a) Packet delivery ratio (b) Forwarding load (Max. node speed: 5 m/s, Group size: 10)



**Fig. 11** Control overhead. (a) Varying node speed (b) Varying group size (Group size for (a): 10, Max. node speed for (b): 5 m/s, Packet rate: 2 pkts/s)



ratio becomes constant as shown in Fig. 9(a). In Fig. 9(b), the forwarding load in general increases linearly as group size increases (except for the flooding scheme). TBM<sub>2</sub> shows a higher forwarding load than the multicast tree but it still can be considered acceptable with the fairly high packet delivery ratio around 99%.

Figure 10 shows performance impact on different traffic load. As default values, the maximum speed of 5 m/s and the group size of 10 are used. TBM<sub>k</sub> shows the significant improvement over the multicast tree as shown in the figure. As the traffic increases, the packet delivery ratio is gradually dropped except for flooding. Note that there is an interesting point that the performance of the multicast mesh decreases significantly beyond 10 packets/s in Fig. 10(a). This is because many forwarding packets are unexpectedly dropped due to network congestion in the multicast mesh as explained earlier in Sect. 3.3. In Fig. 10(b), the forwarding load is increased as group size increases except for flooding.

Figure 11 compares control overhead with respect to the maximum node speed and the group size. Note that the flooding has no control overhead because no network structure is used and a multicast packet is just flooded in the network. The control overhead of the multicast tree (TBM<sub>0</sub>) and TBM<sub>1</sub> is almost the same because the 1-hop redundant paths are given for free as explained in Sect. 3.2. As in Fig. 11(a), the control overhead is increased as

maximum node speed increases but varies a lot with different values of  $k$  in TBM<sub>k</sub>. In Fig. 11(b), the control overhead in general increases linearly as group size increases (except for the flooding scheme). Again, multicast tree performs the same as TBM<sub>1</sub>.

## 5 Conclusions

In this paper, an *adaptive multicast* scheme, called *Tree-Based Mesh with  $k$ -hop redundant paths* (TBM<sub>k</sub>), has been proposed and evaluated. TBM<sub>k</sub> provides *variable density* of redundant paths depending on the status of the network such as traffic, mobility and node connectivity, resulting in *tradeoffs* between multicast tree and multicast mesh. In terms of performance metrics, it makes a tradeoff between control overhead and delivery efficiency. The  $k$ -hop redundant paths are locally obtained by running a distributed algorithm. The most important benefit of TBM<sub>k</sub> is that it can control the level of path redundancy based on network status. According to the performance study, the packet delivery ratio of the proposed TBM<sub>k</sub> is significantly improved compared to the multicast tree. The forwarding load of the proposed TBM<sub>k</sub> is greatly reduced compared to the multicast mesh.

In TBM<sub>k</sub>, the performance and overhead depends on the choice of  $k$ . We are currently investigating an adaptive



mechanism to determine  $k$ . Another interesting future work is to allow each tree node to choose  $k$ . This avoids the overhead of distributing the value of  $k$  once it is determined. Also, it may be effective when network status or node density varies from location to location. A single value of  $k$  cannot be optimal and thus each node comes up with its own optimal value and discovers a desired set of redundant paths in that specific area. Investigation of network parameters that affect the choice of  $k$  and their integration with the current  $TBM_k$  algorithm is another important future work.

**Acknowledgements** This work was supported in part by the Korea Research Foundation Grant funded by the Korean Government (MO-EHRD) (KRF-2005-214-D00355) and it was supported in part by research funds from Chosun University, 2007. A preliminary version of this work was presented at the International Conference on Wireless Algorithms, Systems and Applications, Xian, China, August 15–17, 2006.

## References

- Perkins, C. E. (2001). *Ad hoc networking*. Upper Saddle River: Addison-Wesley Pub. Co.
- Siva Ram Murthy, C., & Manoj, B. S. (2004). *Ad hoc wireless networks*. Upper Saddle River: Prentice Hall.
- Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter (2006). <http://www.ietf.org/html.charters/manet-charter.html>.
- Lee, S.-J., Su, W., Hsu, J., Gerla, M., & Bagrodia, R. (2000). A performance comparison study of ad hoc wireless multicast protocols. In *Proceedings of the IEEE Infocom*, Vol. 2, pp. 565–574.
- Pandey, M., & Zappala, D. (2005). A scenario-based performance evaluation of multicast routing protocols for ad hoc networks. In *Proceedings of 6th IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pp. 31–41.
- Malaguti, M., Taddia, C., Mazzini, G., & Zorzi, M. (2004). Analysis of performance of multicast routing protocols over 802.11b. In *Proceedings of IEEE 60th Semiannual Vehicular Technology Conference (VTC 2004-Fall)*, Vol. 5, pp. 3165–3169.
- Devarapalli, V., & Sidhu, D. (2001). MZR: A multicast protocol for mobile ad hoc networks. In *Proceedings of IEEE International Conference on Communications*, Vol. 3, pp. 886–891.
- Gerla, M., Chiang, C.-C., & Zhang, L. (1999). Tree multicast strategies in mobile, multihop wireless networks. *Baltzer/ACM Journal of Mobile Networks and Applications*, 3, 193–207.
- Royer, E. M., & Perkins, C. E. (1999). Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proceedings of the International Conference on Mobile Computing and Networking*, pp. 207–218.
- Vaishampayan, R., & Garcia-Luna-Aceves, J. J. (2004). Efficient and robust multicast routing in mobile ad hoc networks. In *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pp. 304–313.
- Lee, S., Gerla, M., & Chiang, C. (1999). On-demand multicast routing protocol. In *Proceedings of IEEE Wireless Communications and Networking Conference*, pp. 1298–1302.
- Lee, S., & Kim, C. (2000). Neighbor supporting ad hoc multicast routing protocol. In *Proceedings of the First Annual Workshop on Mobile Ad Hoc Networking and Computing*, pp. 37–44.
- Dhillon, H., & Ngo, H. Q. (2005). CQMP: A mesh-based multicast routing protocol with consolidated query packets. In *Proceedings of IEEE Wireless Communications and Networking Conference*, Vol. 4, pp. 2168–2174.
- Park, S., & Park, D. (2004). Adaptive core multicast routing protocol. *Wireless Networks*, 10, 53–60.
- Obraczka, K., Tsudik, G., & Viswanath, K. (2006). Exploring mesh- and tree-based multicast routing protocols for MANETs. *IEEE Transactions on Mobile Computing*, 5(1), 28–42.
- Jetcheva, J. G., & Johnson, D. B. (2001). Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 33–44.
- Royer, E. M., & Perkins, C. E. (2000). Multicast ad hoc on-demand distance vector (MAODV) routing. IETF Draft, draft-ietf-manet-maodv-00.txt.
- Hostylenos, G., & Polyzos, G. C. (1997). IP multicast for mobile hosts. *IEEE Communications Magazine*, 35, 54–58.
- Toh, C., Guichal, G., & Bunchua, S. (2000). ABAM: On-demand associativity-based multicast routing for ad hoc mobile networks. In *Proceedings of 52nd IEEE Vehicular Technology Conference (VTC Fall 2000)*, Vol. 3, pp. 987–993.
- Bommaiah, E., Liu, M., McAuley, A., & Talpade, R. (1998). AMRoute: Ad-hoc multicast routing protocol. Internet-Draft, draft-talpade-manet-amroute-00.txt.
- Wu, C., Tay, Y., & Toh, C. (1998). Ad hoc multicast routing protocol utilizing increasing id-numberS (AMRIS) functional specification. Internet-Draft, draft-ietf-manet-amris-spec-00.txt.
- Ji, L., & Corson, M. (1998). A lightweight adaptive multicast algorithm. In *Proceedings of IEEE Global Telecommunication Conference (GlobeCom '98)*, Vol. 2, pp. 1036–1042.
- Garcia-Luna-Aceves, J., & Madruga, E. (1999). The core assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17, 1380–1394.
- Sinha, P., Sivakumar, R., & Bharghavan, V. (1999). MCEDAR: Multicast core extraction distributed ad-hoc routing. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '99)*, Vol. 3, pp. 1313–1317.
- Lin, C., & Chao, S. (1999). A multicast routing protocol for multihop wireless networks. In *Proceedings of IEEE Global Telecommunication Conference (GlobeCom '99)*, pp. 235–239.
- Jacquet, P., & Rodolakis, G. (2005). Multicast scaling properties in massively dense ad hoc networks. In *Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, pp. 93–97.
- Rodolakis, G., Meraihi Naimi, A., & Laouiti, A. (2007). Multicast overlay spanning tree protocol for ad hoc networks. In *Proceedings of International Conferences on Wireless/Wired Internet Communications (WWIC)*, pp. 290–301.