On Bandwidth-Efficient Overlay Multicast

De-Nian Yang, *Member*, *IEEE*, and Wanjiun Liao, *Senior Member*, *IEEE*

Abstract—In this paper, we propose a new multicast delivery mechanism for bandwidth-demanding applications in IP networks. Our mechanism, referred to as Multiple-destination Overlay Multicast (MOM), combines the advantages of IP multicast and overlay multicast. We formulate the MOM routing problem as an optimization problem. We then design an algorithm based on Lagrangian relaxation on our formulation and propose a distributed protocol based on the algorithm. For network operators, MOM consumes less network bandwidth than both IP multicast and overlay multicast. For users, MOM uses less interface bandwidth than overlay multicast.

Index Terms—Application-layer multicast, overlay multicast.

1 Introduction

Multicast is an efficient way of one-to-many and many-to-many communications [1]. Each multicast group consists of a set of members, and each member can be a sender or receiver of the group. The sender of a multicast group delivers data over a multicast tree to reach all receivers of the group. For simplicity and robustness, current IP multicast routing protocols [2], [3], [4], [5] rely on shortest path unicast routing protocols to construct a multicast tree in the IP network. Therefore, an IP multicast tree is a shortest-path tree, which is a union of the shortest path from each member to the root of the tree. Although finding a shortest-path tree is not difficult, the bandwidth used in a shortest-path tree is not optimal for multicast communications because the routing of a shortest-path tree is inflexible. The path from the root to each member is fixed in a shortest-path tree, regardless of the identities and locations of other members. Consider Fig. 1, for example, where Fig. 1a is an IP network topology, node 1 is the root, and nodes 6, 11, and 13 are group members. With the shortest-path tree (Fig. 1b), it takes 10 packet hops for transmission, where a packet hop corresponds to sending a packet over one hop. The path from node 1 to node 11 is independent of the path from node 1 to node 13, even though nodes 11 and 13 are quite close. The total bandwidth consumption can be reduced by first sharing a common path from node 1 to node 12 and then connecting nodes 11 and 13 to node 12, but these paths are not the shortest paths in the IP network. Therefore, the above example shows that a shortest-path tree may induce higher bandwidth consumption.

Due to the slow deployment of IP multicast, an overlay multicast, that is, application-layer multicast, is proposed to construct a multicast tree in an overlay network. The routing of an overlay multicast tree is more flexible than the routing of an IP multicast tree because the path from the root to each member is not constrained to be the shortest path in the IP network. The path from the root to each member in an overlay multicast tree can include other members to relay data.

The bandwidth used in an overlay multicast tree corresponds to the total bandwidth consumption of all edges in the tree of the overlay network, where the bandwidth consumption of an edge connecting two members in the overlay network is the total bandwidth consumption of the shortest path connecting the two members in the IP network. Therefore, the overlay multicast tree using the least amount of bandwidth is the minimum spanning tree in the overlay network. Although the routing for an overlay multicast tree is more flexible than the routing for an IP multicast tree, each link in the IP network needs to deliver an identical packet multiple times if the link is located in the shortest paths from a member to more than one other member. For example, in Fig. 1c, there are eight packet hops in the overlay multicast tree. The links from node 6 to node 12 deliver an identical packet twice because node 6 sends the packet to nodes 11 and 13 individually. However, each link in an IP multicast tree delivers each packet exactly once. Thus, an overlay multicast tree is not guaranteed to use less bandwidth than an IP multicast tree. In addition, each member in overlay multicast uses more interface bandwidth if the member is required to send an identical packet multiple times to other members. Here, interface bandwidth refers to the bandwidth for a member to send data to the Internet.

Given a set of the members for a multicast group, the Steiner tree is the multicast tree using the least amount of bandwidth in the IP network. A Steiner tree outperforms an IP multicast tree because the path from the root to each member in the Steiner tree is not constrained to be the shortest path in the IP network. A Steiner tree also outperforms an overlay multicast tree because each link in a Steiner tree delivers each packet exactly once. Unfortunately, the Steiner tree is not currently adopted for multicast communications due to its extremely high computational overhead. In addition, deploying Steiner trees in the Internet requires each router to update and implement a standardized Steiner tree algorithm, which would lead to very high deployment cost and slow deployment for ISPs.

In this paper, we propose a bandwidth-efficient multicast delivery mechanism MOM that adopts the routing flexibility of overlay multicast but alleviates the stress

• *The authors are with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan.*
E-mail: dnyang@cc.ee.nut.edu.tw, wjliao@ntu.edu.tw.

Manuscript received 1 May 2006; revised 16 Dec. 2006; accepted 8 Feb. 2007; published online 23 Apr. 2007.
Recommended for acceptance by C. Raghavendra.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0107-0506.
Digital Object Identifier no. 10.1109/TPDS.2007.1104.

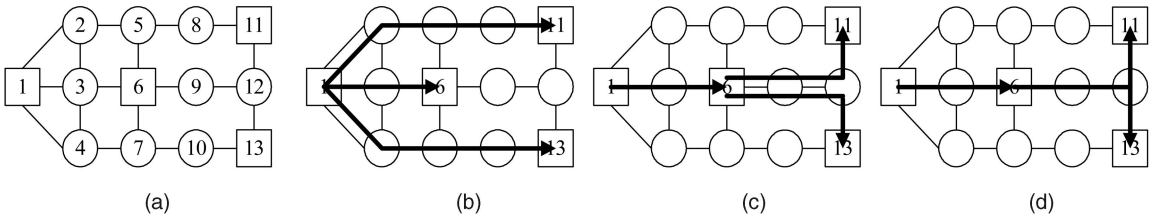

Fig. 1. An example comparing the bandwidth consumption in a shortest-path tree (10 packets), an overlay multicast tree (eight packets), and a Multiple-destination Overlay Multicast (MOM) tree (six packets). (a) Network topology. (b) Shortest-path tree. (c) Overlay multicast tree. (d) MOM tree.

problem by including multiple destination addresses in each IP header [6], [7], [8]. With MOM, a router uses the existing unicast routing protocol to find the neighbor routers for packet delivery. A packet is duplicated only when it must be delivered to more than one neighbor router. Including multiple addresses in a packet has many advantages, as have been demonstrated in the literature (for example, [9], [10], [11], [12]).

MOM combines the advantages of overlay multicast and IP multicast. The path from the root to each member is not constrained to be the shortest path in the IP network, and each link in MOM sends much fewer packets. Consider Fig. 1d, for example, where there are six packet hops sent in an MOM tree. Node 6 sends one packet with the addresses of nodes 11 and 13. The packet follows the shortest paths to nodes 11 and 13 and is duplicated at node 12. Each link sends exactly one packet. This example shows that MOM has the following advantages. For network operators, MOM consumes less network bandwidth than both IP multicast and overlay multicast. MOM uses less network bandwidth, thanks to the flexibility of routing in overlay multicast and much fewer packets sent on each link. For users, MOM consumes less interface bandwidth than overlay multicast, thanks to multiple addresses being included in each packet. As a result, each member can send data to other members with much fewer packets. In the next section, we prove that IP multicast and overlay multicast are two special cases of MOM.

In this paper, we model the MOM routing problem as an optimization problem. The problem is to minimize the total number of packet hops sent in an MOM tree. We use Integer Linear Programming (ILP) to formulate the MOM routing problem, which is NP-hard with the reduction from Vertex Cover [29]. We design an algorithm based on Lagrangian relaxation on our formulation. We adopt the Lagrangian relaxation instead of other optimization techniques due to the following reasons. First, our algorithm decomposes the original problem into multiple subproblems such that each subproblem can be solved by each member independently. In other words, the algorithm can be implemented in a distributed manner. Second, the algorithm adapts to the change to group membership and network topology. The algorithm iteratively reduces the cost of a multicast tree according to the current group membership and the IP network topology. Third, the algorithm can find the lower bound on the total number of packets sent in the optimal MOM tree (MOM-OPT). Since using the ILP formulation to find the MOM-OPT with a large number of members in the large IP network is computationally infeasible, the lower bound obtained by our algorithm provides the benchmark to compare with other algorithms for the problem.

Compared with the algorithms that find the Steiner trees in the IP network, our algorithm can support a large number of multicast groups because the members collaboratively find the multicast trees. We believe that members are more suited than routers to find a multicast tree because each member tends to participate in much fewer multicast groups than routers, allowing each of them to contribute computation power to find the tree for each participating group. In addition, MOM does not require any new routing algorithm, and ISPs only need to enable the multidestination delivery in routers. We also design a protocol based on the proposed algorithm, and the protocol can be implemented as a client middleware for overlay multicast services.

MOM is applied to applications different from IP multicast and overlay multicast. Its design goal is to minimize the bandwidth consumption, whereas the IP multicast is to support a large number of members. Previous research results for overlay multicast focus mainly on scalability [13], [14], [15], [16], [17] or on real-time applications to minimize end-to-end delay [18], [19], [20], [21]. The approaches that improve the scalability are typically based on hierarchical network structures or Delaunay triangulation overlays. For real-time applications, approximation and heuristic algorithms are used to find the overlay networks with bandwidth and delay constraints. However, this paper focuses on bandwidth-demanding applications such as the distribution of large files between corporate servers or long-term applications with medium data rate.

The rest of this paper is summarized as follows: Section 2 addresses the problem description and our proposed ILP formulation. Section 3 explains our algorithm based on the Lagrangian relaxation on our ILP formulation. Section 4 presents the simulation results. Finally, this paper is concluded in Section 5.

2 Problem Description

In this paper, we focus on one-to-many multicast communications. The root of a multicast tree is the sender of the group, and other members are the receivers of the group. We formulate the MOM routing problem as an optimization problem. The problem is to find the MOM tree such that the total number of packets sent in the tree in the IP network is minimized. The sender in the MOM tree delivers data to some other members. Some members may need to relay data to other members such that each member can receive the data from the sender. Each packet can include the addresses of multiple receivers. Since the forwarding delay of a packet in the router is proportional to the number of

TABLE 1
Input Parameters of the MOM Routing Problem

Symbol	Description
s	the sender of the multicast group
R	the set of receivers in the multicast group
M	the set of members in the multicast group; $M = \{s\} \cup R$
G	the directed graph modeling the IP network
V, A	the set of nodes and arcs in the IP network G
G_c	the complete graph corresponding to the overlay network of M
V_c, A_c	the set of nodes and arcs in the overlay network G_c
T_p	the shortest-path tree in G rooted at member p; each leaf node q in T_p is another member in M, and only the root and the leaf nodes in T_p can be the members in M
V_p, A_p	the set of nodes and arcs in T_p, $V_p \subseteq V$, $A_p \subseteq A$, $\forall p \in M$
L_p	the set of leaf nodes in T_p, $L_p \subseteq M$, $\forall p \in M$
C_u^p	the set of child nodes of u in T_p, $C_u^p \subseteq V_p$, $\forall p \in M$, $\forall u \in V_p$
$P_{u,v}$	the set of arcs in the shortest path from u to v in G, $P_{u,v} \subseteq A_p$, $\forall u, v \in V$
δ	the maximum number of addresses that can be included in each packet

TABLE 2
Decision Variables of the MOM Routing Problem

Symbol	Description
$\chi_{p,q}^m$	a binary variable that represents if arc (p,q) in G_C is in the MOM tree and in the path from s to member m, $\forall m \in R$, $\forall (p,q) \in A_C$
$\tau_{u,v}^p$	an integer variable that represents the number of leaf nodes in T_p with the addresses in the packets sent in arc (u,v) of T_p, $\forall p \in M$, $\forall (u,v) \in A_p$
$\pi_{u,v}^p$	an integer variable that represents the number of packets delivered in arc (u,v) of T_p, $\forall p \in M$, $\forall (u,v) \in A_p$
$D_{u,v}^p$	the set of leaf nodes in T_p with the addresses in the packets sent in arc (u,v) of T_p, $\forall p \in M$, $\forall (u,v) \in A_p$, namely, $\left\|D_{u,v}^p\right\| = \tau_{u,v}^p$
$K_{u,v}^p$	a subset of $D_{u,v}^p$ such that each leaf node in $K_{u,v}^p$ belongs to the packet sent in arc (u,v) with δ addresses, $\forall p \in M$, $\forall (u,v) \in A_p$
$U_{u,v}^p$	a subset of $D_{u,v}^p$ such that each leaf node in $U_{u,v}^p$ belongs to the packet sent in arc (u,v) with fewer than δ addresses; namely, $U_{u,v}^p = D_{u,v}^p - K_{u,v}^p$

addresses in the header [7], we have a constraint that each packet contains at most δ addresses to limit the packet forwarding delay, where δ is a positive integer. Therefore, a member is required to send more than one packet to a neighbor router if the router needs to send data to more than δ members. Network operators can adjust δ to find the best trade-off between the packet forwarding delay and the bandwidth consumption according to the forwarding speed of a router.

We list the input parameters and decision variables of the MOM routing problem in Tables 1 and 2. Let ϕ_{SPT}, ϕ_{ST}, ϕ_{OM}^*, and $\phi_{\mathrm{XOM}}^*(\delta)$ denote the number of packets sent in the shortest-path tree, Steiner tree, optimal overlay multicast tree (minimum spanning tree), and MOM-OPT, with at most δ addresses in each packet, respectively. We first prove that IP multicast and overlay multicast are two special cases of MOM.

Lemma 1. *If δ_1 is no less than δ_2, then the following inequality holds:*

$$\phi_{\mathrm{XOM}}^*(\delta_1) \leq \phi_{\mathrm{XOM}}^*(\delta_2).$$

Proof. The above inequality holds because any MOM tree with at most δ_2 addresses in each packet is a feasible solution to the MOM routing problem with at most δ_1 addresses in each packet. □

Theorem 1. *The following inequalities hold:*

$$\phi_{\mathrm{SPT}} \geq \phi_{\mathrm{XOM}}^*(|R|),\ \phi_{\mathrm{OM}}^* \geq \phi_{\mathrm{XOM}}^*(\delta),$$
$$\textit{and}\quad \phi_{\mathrm{XOM}}^*(\delta) \geq \phi_{\mathrm{ST}},\ \textit{where}\quad \delta \geq 1.$$

Proof. The routing of an MOM tree is identical to the shortest-path tree if the sender directly sends the data to all receivers. Besides that, each arc in the MOM tree delivers exactly one packet if δ is $|R|$. Therefore, the shortest-path tree is a feasible solution to the MOM routing problem, with δ being $|R|$, and ϕ_{SPT} is no less than $\phi_{\mathrm{XOM}}^*(|R|)$. The optimal overlay multicast tree is the minimum spanning tree, which is also the optimal solution to the MOM routing problem, with δ being 1. In other words, ϕ_{OM}^* is identical to $\phi_{\mathrm{XOM}}^*(1)$. Therefore, ϕ_{OM}^* is no less than $\phi_{\mathrm{XOM}}^*(\delta)$ according to Lemma 1, $\delta \geq 1$. Since any MOM tree, with δ being $|R|$, is a feasible solution to the Steiner tree problem, $\phi_{\mathrm{XOM}}^*(|R|)$ is no less than ϕ_{ST}. Therefore, $\phi_{\mathrm{XOM}}^*(\delta)$ is no less than ϕ_{ST} according to Lemma 1, $\delta \geq 1$. □

We formulate the MOM routing problem as an ILP problem. The formulation has the following objective function:

$$\min \sum_{p \in M} \sum_{(u,v) \in A_p} \pi_{u,v}^p.$$

The above objective function minimizes the number of packets sent in an MOM tree in the IP network. The formulation has the following constraints:

$$\sum_{p:(p,q) \in A_C} \chi_{p,q}^m - \sum_{p:(q,p) \in A_C} \chi_{q,p}^m = 0,\ \forall m \in R, \forall q \in R - \{m\}, \tag{1}$$

$$\sum_{p:(p,m) \in A_C} \chi_{p,m}^m - \sum_{p:(m,p) \in A_C} \chi_{m,p}^m = 1,\ \forall m \in R, \tag{2}$$

$$\sum_{q:(s,q) \in A_C} \chi_{s,q}^m - \sum_{q:(q,s) \in A_C} \chi_{q,s}^m = 1,\ \forall m \in R, \tag{3}$$

$$\chi_{p,q}^m \leq \tau_{u,q}^p,\ \forall (p,q) \in A_C, \forall m \in R, \forall (u,q) \in A_p, \tag{4}$$

$$\sum_{x \in C_v^p} \tau_{v,x}^p = \tau_{u,v}^p, \ \forall p \in M, \forall (u,v) \in A_p, v \notin L_p, \quad (5)$$

$$\tau_{u,v}^p \leq \delta \times \pi_{u,v}^p, \ \forall p \in M, \forall (u,v) \in A_p. \quad (6)$$

Constraints (1), (2), and (3) obtain the identities of relaying members in the path from the sender s to each member m in the MOM tree. For each relaying member, constraint (1) finds the two adjacent members in the path. Constraints (2) and (3) decide the adjacent relaying member for member m and sender s in the path. Therefore, for each member p, we can obtain the set of other members to which p must relay data with the three constraints. Note that member p must send data to other members via T_p. For each arc (u,v) in T_p, constraints (4) and (5) find $\tau_{u,v}^p$, which is the number of downstream members served by arc (u,v). If member q is relayed by p, then constraint (4) enforces that $\tau_{u,q}^p$ is 1 for the incident arc (u,q). For each arc (u,v) in T_p, constraint (5) obtains $\tau_{u,v}^p$ according to $\tau_{v,x}^p$ of each incident arc of v. Therefore, we can obtain the addresses in the packets sent in each arc (u,v) with the above two constraints, and constraint (6) can therefore find the number of packets required to be sent in (u,v). In addition to the six constraints above, there are the constraints that enforce that $\chi_{p,q}^m$, $\tau_{u,v}^p$, and $\pi_{u,v}^p$ are all binary variables. We regard an MOM tree that obeys the above constraints as a *feasible solution* of the MOM routing problem.

3 Algorithm Based on Lagrangian Relaxation

In this section, we design an algorithm based on Lagrangian relaxation on our formulation. The algorithm finds both a feasible MOM tree and the lower bound on the total number of packets sent in the MOM-OPT. For a multicast group with a large number of members in a large IP network, the lower bound provides the benchmark to compare with other algorithms, since finding a large MOM-OPT with the ILP formulation is computationally infeasible.

The algorithm relaxes a constraint of our formulation to transform the MOM routing problem into the *Lagrangian Relaxation Problem (LRP)*. LRP has a new objective function with the *Lagrange multipliers* and fewer constraints such that we can decompose LRP into multiple subproblems, where each subproblem corresponds to a member and can be solved independently by the member. The members in our algorithm periodically exchange and update the Lagrange multipliers to iteratively reduce the total bandwidth consumption of an MOM tree according to the current group membership and the network topology. We describe how we can solve the MOM routing problem as follows:

- Transform the MOM routing problem into the LRP.
- Decompose LRP into multiple subproblems and solve each subproblem.
- Construct an MOM tree according to the solutions to the subproblems.
- Reduce the total bandwidth consumption of the MOM tree by iteratively updating the Lagrange multipliers.

3.1 Problem Transformation and Decomposition

Our algorithm relaxes constraint (4) to transform the MOM routing problem into LRP, and the objective function of LRP is expressed as follows:

$$\begin{aligned}
&\min \sum_{p \in M} \sum_{(u,v) \in A_p} \pi_{u,v}^p + \sum_{(p,q) \in A_C} \sum_{m \in R} \sum_{u:(u,q) \in A_p} \alpha_{p,q}^m \times \left(\chi_{p,q}^m - \tau_{u,q}^p \right) \\
&= \sum_{m \in R} \sum_{(p,q) \in A_C} \alpha_{p,q}^m \times \chi_{p,q}^m + \sum_{p \in M} \sum_{(u,v) \in A_p} \pi_{u,v}^p - \sum_{(p,q) \in A_C} \sum_{u:(u,q) \in A_p} \\
&\quad \left(\sum_{m \in R} \alpha_{p,q}^m \right) \times \tau_{u,q}^p \\
&= \sum_{m \in R} \sum_{(p,q) \in A_C} \alpha_{p,q}^m \times \chi_{p,q}^m + \sum_{p \in M} \Bigg[\sum_{(u,v) \in A_p} \pi_{u,v}^p - \sum_{q \in L_p} \sum_{u:(u,q) \in A_p} \\
&\quad \left(\sum_{m \in R} \alpha_{p,q}^m \right) \times \tau_{u,q}^p \Bigg],
\end{aligned}$$

where $\alpha_{p,q}^m$ is the Lagrange multiplier, $\alpha_{p,q}^m \geq 0$, $\forall m \in R$, and $\forall (p,q) \in A_C$. LRP includes constraints (1), (2), (3), (5), and (6). Compared with the objective function of the MOM routing problem, the objective function of LRP owns a new term corresponding to the relaxed constraint (4). Intuitively, for any feasible solution to LRP that contradicts constraint (4), namely, $\chi_{p,q}^m > \tau_{u,q}^p$, the objective function penalizes the solution with a larger objective value. Moreover, any feasible solution to the MOM routing problem must also act as a feasible solution to LRP, since the set of constraints of LRP is a subset of the constraints of the MOM routing problem. If we adopt the optimal solution to the MOM routing problem as a feasible solution to both LRP and the MOM routing problems, then the objective value of LRP must be no more than the objective value of the MOM routing problem because the new term in the objective function of LRP must be nonpositive. Therefore, the objective value of the optimal solution to LRP must be no more than the objective value of the optimal solution to the MOM routing problem. In other words, the optimal solution to LRP provides a lower bound on the objective value of the optimal solution to the MOM routing problem, where the objective value of the MOM routing problem is the total number of packets sent in the MOM-OPT.

We solve LRP by decomposing LRP into two subproblems. We divide the objective function and the constraints of LRP into two parts, where each subproblem owns one part of the objective function and constraints of LRP. The variables in the two subproblems are mutually independent such that we can solve each subproblem individually, and the solution to LRP is just the combination of the solutions to the two subproblems.

Since the objective function of LRP can be divided into two terms, our algorithm decomposes LRP into two subproblems, where the objective function of the first subproblem is expressed as

$$\min \sum_{m \in R} \sum_{(p,q) \in A_C} \alpha_{p,q}^m \times \chi_{p,q}^m. \quad (7)$$

The constraints of the first subproblem include constraints (1), (2), and (3). The first subproblem is identical to the shortest path problem for sender s and each receiver m in the overlay network. We can solve the problem with any distributed shortest path algorithm. Note that the cost $\alpha^m_{p,q}$ of each arc (p,q) in the overlay network can be different for each receiver m in the shortest path problem.

The objective function of the second subproblem is expressed as follows:

$$\min \sum_{p\in M}\left[\sum_{(u,v)\in A_p}\pi^p_{u,v} - \sum_{q\in L_p}\sum_{u:(u,q)\in A_p}\left(\sum_{m\in R}\alpha^m_{p,q}\right)\times\tau^p_{u,q}\right]. \quad (8)$$

The constraints of the second subproblem include constraints (5) and (6). We decompose the subproblem into the *Leaf Selection Problem (LSP)* for each member p in M. For each member p in M, LSP selects some leaf nodes in T_p to which p must send data. The objective function of LSP for member p contains two terms. The first one is the total number of packets sent in T_p, and the second one is the sum of the profit of each selected leaf node. Each leaf node q creates profit $\sum_{m\in R}\alpha^m_{p,q}$ if it is selected; that is, variable $\tau^p_{u,q}$ is 1, where u is the parent node of q in T_p. Therefore, the objective function is to minimize the *net cost* for each member p to send data to the selected leaf nodes in T_p. The constraints in LSP for each member p enforces that each router in T_p must deliver packets according to the destination addresses. Intuitively, we obtain more profit in this problem if we choose more leaf nodes. However, more selected leaf nodes also consume more bandwidth in T_p. Therefore, we have to find the best tradeoff to select the leaf nodes.

3.2 Solving the LSP

We design a dynamic programming algorithm to find the optimal solution to LSP. We select the dynamic programming method to exploit the tree structure of this problem. For each member p, the method enables each node u in tree T_p to avoid storing and computing the net costs of all possible selections of the downstream leaf nodes. In contrast, each node u in the dynamic programming method stores and computes only the best selections of downstream leaf nodes when u receives a packet with k addresses, where $1\le k\le\delta$. In other words, node u is required to store and compute only δ possible selections of downstream leaf nodes. Therefore, the dynamic programming method can effectively reduce the memory and computational time to solve the problem.

Each arc (u,v) in T_p needs to deliver packets to the downstream selected leaf nodes $D^p_{u,v}$. To minimize the number of packets sent in the arc, we have to include the addresses of more downstream selected leaf nodes in each packet. Therefore, our algorithm allows only one packet with fewer than δ addresses, whereas every other packet must include δ addresses. For arc (u,v) in T_p, let $K^p_{u,v}$ denote the set of *packed selected leaves* of the arc, where each selected leaf node in the set is located in a packet with δ addresses. In contrast, let $U^p_{u,v}$ denote the set of *unpacked selected leaves* of the arc, and each selected leaf nodes in the set is located in a packet with fewer than δ addresses. In our dynamic programming algorithm, we obtain $K^p_{u,v}$ and $U^p_{u,v}$ from $K^p_{v,w}$ and $U^p_{v,w}$ of each downstream arc (v,w) of (u,v) such that the size of $U^p_{u,v}$ is smaller than δ. More specifically, we pack the unpacked selected leaves of each downstream arc (v,w) to $K^p_{u,v}$ to reduce the size of $U^p_{u,v}$. Therefore, our algorithm is operated in a bottom-up manner.

For each member p and each arc (u,v) in T_p, let $\varepsilon^p_{u,v}(j_{u,v})$ denote the optimal net cost obtained in the subtree that includes all arcs in the path from p to v and all arcs downstream to v, where $j_{u,v}$ is the number of unpacked selected leaves in the subtree. We find $\varepsilon^p_{u,v}(j_{u,v})$ in the bottom-up manner as follows: We first consider the leaf node. For each leaf node q, with u being the parent node in T_p, net cost $\varepsilon^p_{u,q}(1)$ is therefore $1-\sum_{m\in R}\alpha^m_{p,q}$ because q is a selected leaf node in this case. In contrast, net cost $\varepsilon^p_{u,q}(0)$ is zero and corresponds to the case that q is not selected. We then consider each node v with fewer than δ child nodes, where a child node is a leaf node of T_p. Net cost $\varepsilon^p_{u,v}(j_{u,v})$ represents the case that $j_{u,v}$ child nodes $\{q_1,q_2,\ldots,q_{j_{u,v}}\}$ are selected, and we obtain $\varepsilon^p_{u,v}(j_{u,v})$ as follows:

$$\varepsilon^p_{u,v}(j_{u,v}) = \begin{cases} 0, j_{u,v}=0 \\ 1+j_{u,v}-\sum_{k=1}^{j_{u,v}}\sum_{m\in R}\alpha^m_{p,q_k}, 1\le j_{u,v}<\delta. \end{cases} \quad (9)$$

Here, arc (u,v) in T_p needs to send one packet with the addresses of the $j_{u,v}$ leaf nodes if $j_{u,v}$ is positive. If v has exactly δ child nodes, where each child node is a leaf node, then $\varepsilon^p_{u,v}(j_{u,v})$ becomes

$$\varepsilon^p_{u,v}(j_{u,v}) = \begin{cases} \min\left\{0, |P_{p,v}|+\delta-\sum_{k=1}^{\delta}\sum_{m\in R}\alpha^m_{p,q_k}\right\}, j_{u,v}=0 \\ 1+j_{u,v}-\sum_{k=1}^{j_{u,v}}\sum_{m\in R}\alpha^m_{p,q_k}, 1\le j_{u,v}<\delta. \end{cases} \quad (10)$$

Here, $|P_{p,v}|$ is the number of arcs in the path from p to v. Note that we have two cases when $j_{u,v}$ is zero. No child node is selected, or all the δ child nodes are selected. In the latter case, all the δ child nodes become the packed selected leaves and belong to the same packet sent in each arc of the path from p to v. Therefore, each arc in the path needs to send one packet for the δ selected leaf nodes. Finally, if v has more than δ child nodes, where each child node is a leaf node, then net cost $\varepsilon^p_{u,v}(j_{u,v})$ selects the i child nodes $\{q_1,q_2,\ldots,q_i\}$ with the largest profit such that $\varepsilon^p_{u,v}(j_{u,v})$ satisfies the following:

$$\varepsilon^p_{u,v}(j_{u,v}) = \min_i\left\{\left\lceil\frac{j_{u,v}}{\delta}\right\rceil + |P_{p,v}|\times\left\lfloor\frac{i}{\delta}\right\rfloor + i - \sum_{k=1}^{i}\sum_{m\in R}\alpha^m_{p,q_k}\,\middle|\, i \bmod \delta = j_{u,v}\right\}, 0\le j_{u,v}<\delta. \quad (11)$$

Function LSP_DP

Input: $T_p=(V_p,A_p)$, $\alpha_{p,q}^m$, $\forall m\in M$, $\forall q\in L_p$.

Find: $\tau_{u,v}^p$, and $\pi_{u,v}^p$, $\forall p\in M$, $\forall (u,v)\in A_p$.

1. Number all nodes in T_p by the breadth-first-search algorithm;

   let $\varepsilon_{u,q}^p(0)\leftarrow 0$, $\varepsilon_{u,q}^p(1)\leftarrow 1-\sum_{m\in R}\alpha_{p,q}^m$, $\varepsilon_{u,q}^p(j_{u,q})\leftarrow\infty$, $\forall q\in L_p$, $(u,q)\in A_p$, $2\le j_{u,q}<\delta$.

2. For v from $|V_p|$ to 1, $v\notin L_p$,

   for $j_{u,v}$ from 0 to $\delta-1$, $(u,v)\in A_p$,

   $$\text{let } \varepsilon_{u,v}^p(j_{u,v})\leftarrow \min_{\{j_{v,w}:w\in C_v^p\}}\left\{\left\lceil\frac{j_{u,v}}{\delta}\right\rceil+|P_{p,v}|\times\left\lfloor\frac{1}{\delta}\times\sum_{w\in C_v^p}j_{v,w}\right\rfloor+\sum_{w\in C_v^p}\varepsilon_{v,w}^p(j_{v,w})\;\middle|\;0\le j_{v,w}<\delta,\sum_{w\in C_v^p}j_{v,w}\bmod\delta=j_{u,v}\right\};$$

   end for;

   end for.

3. Let $\bar{j}_{p,x}\leftarrow\arg\min_{0\le j_{p,x}<\delta}\{\varepsilon_{p,x}^p(j_{p,x})\}$, $\forall x\in C_p^p$;

   for v from 1 to $|V_p|$, $v\notin L_p$, $(u,v)\in A_p$,

   $$\text{let } \{\bar{j}_{v,w}\}\leftarrow\arg\min_{\{j_{v,w}:w\in C_v^p\}}\left\{\left\lceil\frac{j_{u,v}}{\delta}\right\rceil+|P_{p,v}|\times\left\lfloor\frac{1}{\delta}\times\sum_{w\in C_v^p}j_{v,w}\right\rfloor+\sum_{w\in C_v^p}\varepsilon_{v,w}^p(j_{v,w})\;\middle|\;0\le j_{v,w}<\delta,\sum_{w\in C_v^p}j_{v,w}\bmod\delta=\bar{j}_{u,v}\right\};$$

   end for.

4. For each leaf node q in L_p, $(u,q)\in A_p$,

   if $\bar{j}_{u,q}=1$, then let $\tau_{u,q}^p\leftarrow 1$, $\pi_{u,q}^p\leftarrow 1$; else let $\tau_{u,q}^p\leftarrow 0$, $\pi_{u,q}^p\leftarrow 0$;

   end if;

   end for;

   for v from $|V_p|$ to 1, $v\notin L_p$, let $\tau_{u,v}^p\leftarrow\sum_{w\in C_v^p}\tau_{v,w}^p$, $\pi_{u,v}^p\leftarrow\lceil\tau_{u,v}^p/\delta\rceil$, $(u,v)\in A_p$;

   end for.

Fig. 2. Dynamic programming algorithm for the LSP.

Here, $\lceil j_{u,v}/\delta\rceil$ packet with the unpacked selected leaves must be sent in arc (u,v), and $\lfloor i/\delta\rfloor$ packets with the packed selected leaves must be sent in each arc of the path from p to v. Note that we do not consider the packet with the unpacked selected leaves sent in the path from p to v, since the packet can be merged with other packets with other unpacked selected leaves located in the different branches of T_p. Therefore, we count the packet induced by these unpacked selected leaves later when we consider the upstream arcs. Note that (9) and (10) are two special cases of (11), and the above scenario considers only each node v, with all child nodes being the leaf nodes of T_p. For each other node v in T_p, we generalize $\varepsilon_{u,v}^p(j_{u,v})$ as follows:

$$\varepsilon_{u,v}^p(j_{u,v})=\min_{\{j_{v,w}:w\in C_v^p\}}\left\{\left\lceil\frac{j_{u,v}}{\delta}\right\rceil+|P_{p,v}|\times\left\lfloor\frac{1}{\delta}\times\sum_{w\in C_v^p}j_{v,w}\right\rfloor+\sum_{w\in C_v^p}\varepsilon_{v,w}^p(j_{v,w})\;\middle|\;0\le j_{v,w}<\delta,\sum_{w\in C_v^p}j_{v,w}\bmod\delta=j_{u,v}\right\},\tag{12}$$

where $\{j_{v,w}:w\in C_v^p\}$ is a set containing a $j_{v,w}$ for each downstream arc (v,w). The net cost consists of the packet with the unpacked selected nodes sent in (u,v), the packets merged from the downstream arcs and sent in path $P_{p,v}$, and the sum of the net costs obtained from the downstream arcs. One way to find $\varepsilon_{u,v}^p(j_{u,v})$ is to consider all the $O(\delta^{|C_v^p|})$ combinations of $j_{v,w}$ for each downstream arc (v,w). However, we reduce the runtime to $O(\delta^2\times|C_v^p|^2)$ with an auxiliary graph in [1].

Fig. 2 gives our dynamic programming algorithm for LSP. Step 1 sets the initial value for each leaf node. Step 2 finds $\varepsilon_{u,v}^p(j_{u,v})$ for each arc (u,v) and each $j_{u,v}$ in the bottom-up manner. Step 3 determines the optimal $\bar{j}_{u,v}$ for each arc (u,v) in the top-down manner, where $\bar{j}_{u,v}$ is the optimal number of unpacked selected leaves obtained in our algorithm. Finally, step 4 finds the selected leaf nodes and the net cost of the tree T_p. For example, consider the tree in Fig. 3, where δ is 3, and the number below each receiver is the profit associated with the receiver q, that is,

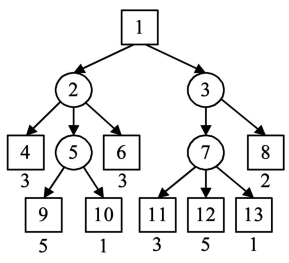

Fig. 3. An example multicast tree for the LSP, $\delta=3$.

TABLE 3
An Example of the Dynamic Programming Algorithm for the LSP

$j_{2,5}$	$(j_{5,9}, j_{5,10})$	$\varepsilon^1_{2,5}(j_{2,5})$	$j_{1,2}$	$(j_{2,4}, j_{2,5}, j_{2,6})$	$\varepsilon^1_{1,2}(j_{1,2})$
0	(0,0)	$\min\{0\}$	**0**	(0,0,0) ,**(1,1,1)**, (1,2,0) , (0,2,1)	$\min\{0,-6,-4,-4\}$
1	**(1,0)**, (0,1)	$\min\{-3,1\}$	1	(1,0,0) , (0,1,0) , (0,0,1) , (1,2,1)	$\min\{-1,-2,-1,-5\}$
2	(1,1)	$\min\{-3\}$	2	(1,1,0) , (1,0,1) , (0,1,1)	$\min\{-4,-3,-4\}$
$j_{3,7}$	$(j_{7,11}, j_{7,12}, j_{7,13})$	$\varepsilon^1_{3,7}(j_{3,7})$	$j_{1,3}$	$(j_{3,7}, j_{3,8})$	$\varepsilon^1_{1,3}(j_{1,3})$
0	(0,0,0) , (1,1,1)	$\min\{0,-4\}$	**0**	(0,0) ,**(2,1)**	$\min\{-4,-5\}$
1	(1,0,0) , (0,1,0) , (0,0,1)	$\min\{-1,-3,1\}$	1	(1,0) , (0,1)	$\min\{-2,-4\}$
2	**(1,1,0)**, (1,0,1) , (0,1,1)	$\min\{-5,-1,-3\}$	2	(2,0) , (1,1)	$\min\{-4,-3\}$

$\sum_{m\in R} \alpha^m_{1,q}$. For each arc (u,v) in T_1, Table 3 lists $j_{u,v}$ for each arc (u,v), the possible combinations of $j_{v,w}$ of the each downstream arc (v,w), and the net cost $\varepsilon^p_{u,v}(j_{u,v})$. The $j_{u,v}$ in boldface is the optimal $\bar{j}_{u,v}$, which corresponds to an optimal collection of $\bar{j}_{v,w}$ found in step 3 of our algorithm. For example, considering arc (1, 2), $\bar{j}_{1,2}$ is zero, since $\varepsilon^1_{1,2}(0)$ has the minimum net cost. Moreover, $(\bar{j}_{2,4}, \bar{j}_{2,5}, \bar{j}_{2,6})$ of the downstream arcs corresponding to $\bar{j}_{1,2}$ is (1, 1, 1). Therefore, $\bar{j}_{2,5}$ is 1, $(\bar{j}_{5,9}, \bar{j}_{5,10})$ is (1, 0), and member 9 is a selected leaf node. The selected leaf nodes are nodes 4, 9, 6, 11, 12, and 8 in this example.

Next, we prove that our algorithm obtains the optimal solution to LSP in $O\left(\delta^2 \times |V_p|^2\right)$.

Lemma 2. *For each member p, the following holds:*

$$\min \sum_{(u,v)\in A_p} \pi^p_{u,v} - \sum_{q\in L_p} \sum_{u:(u,q)\in A_p} \left(\sum_{m\in R} \alpha^m_{p,q}\right) \times \tau^p_{u,q} = \sum_{x\in C^p_p} \min_{0\le j_{p,x}<\delta} \left\{\varepsilon^p_{p,x}(j_{p,x})\right\}.$$

Proof. For each child node x of p, let $T_{p,x}$ denote the tree with node p, arc (p,x), and the subtree rooted at x. Let $A_{p,x}$ and $L_{p,x}$ denote the set of arcs and leaf nodes in $T_{p,x}$. Therefore, according to the definition of $\varepsilon^p_{p,x}(j_{p,x})$, the following holds:

$$\min \sum_{(u,v)\in A_{p,x}} \pi^p_{u,v} - \sum_{q\in L_{p,x}} \sum_{u:(u,q)\in A_{p,x}} \left(\sum_{m\in R} \alpha^m_{p,q}\right) \times \tau^p_{u,q} = \min_{0\le j_{p,x}<\delta} \left\{\varepsilon^p_{p,x}(j_{p,x})\right\}.$$

For any two nodes x and $\hat{x}$ that are different child nodes of p, the selection of leaf nodes in $T_{p,x}$ is independent of the selection of leaf nodes in $T_{p,\hat{x}}$. Therefore, the following holds:

$$\begin{aligned}
&\min \sum_{(u,v)\in A_p} \pi^p_{u,v} - \sum_{q\in L_p} \sum_{u:(u,q)\in A_p} \left(\sum_{m\in R} \alpha^m_{p,q}\right) \times \tau^p_{u,q} \\
&= \sum_{x\in C^p_p} \min \left\{ \sum_{(u,v)\in A_{p,x}} \pi^p_{u,v} - \sum_{q\in L_{p,x}} \sum_{u:(u,q)\in A_{p,x}} \left(\sum_{m\in R} \alpha^m_{p,q}\right) \times \tau^p_{u,q} \right\} \\
&= \sum_{x\in C^p_p} \min_{0\le j_{p,x}<\delta} \left\{\varepsilon^p_{p,x}(j_{p,x})\right\}.
\end{aligned}$$

The lemma thereby follows. □

Theorem 3. *The dynamic programming algorithm obtains the optimal solution to LSP in $O\left(\delta^2 \times |V_p|^2\right)$.*

Proof. For each arc (u,v) in T_p, we can obtain $\varepsilon^p_{u,v}(j_{u,v})$ for all $j_{u,v}$ in $O\left(\delta^2 \times |C^p_v|^2\right)$. The computational time of our algorithm is $O\left(\sum_{v\in V_p} \delta^2 \times |C^p_v|^2\right)$, which is at most $O\left(\delta^2 \times |V_p|^2\right)$. Therefore, with Lemma 2, our algorithm finds the optimal solution to LSP in $O\left(\delta^2 \times |V_p|^2\right)$. □

3.3 Finding and Improving the Feasible MOM Tree

The solution to LSP for each member may not build a feasible multicast tree for the MOM routing problem because each member independently selects the leaf nodes to send data. Some members may not be able to receive data from the sender. Therefore, we use the solution to the shortest path problem for each member in the overlay network, which is the first subproblem to LRP, to find the feasible MOM tree. For each receiver m, the shortest path problem for m decides the identity of each member p located in the path from the sender to m, and each member p can then obtain the identities of other members to which p must relay data. The solution guarantees that each member is able to receive the data from the sender. Note that the MOM tree is different from the IP multicast tree because the cost $\alpha^m_{p,q}$ of each arc (p,q) can be different for each receiver m. In other words, the routing for the MOM tree in the overlay network is not restricted, since each receiver can individually choose a path to connect to the sender by properly assigning the cost to each arc. In our algorithm, we

Input: $G=(V,E)$, M, δ.

Find: $\chi_{p,q}^m$, $\forall m \in R$, $\forall (p,q) \in A$.

1. Let $\omega \leftarrow 1$, $\alpha_{p,q}^m \leftarrow 1$, $\forall m \in R$, $\forall (p,q) \in A_C$.
2. Find $\chi_{p,q}^m$ using the shortest-path algorithm, $\forall m \in R$, $\forall (p,q) \in A_C$;
   LSP_DP(T_p, $\alpha_{p,q}^m$, $\forall m \in M$, $\forall q \in L_p$), $\forall p \in M$;
   find ϕ_{XOM} from $\chi_{p,q}^m$, $\forall m \in R$, $\forall (p,q) \in A_C$.
3. If $\phi_{\mathrm{XOM}} - \left[\sum_{p \in M}\sum_{(u,v)\in A_p} \pi_{u,v}^p + \sum_{(p,q)\in A_C}\sum_{m\in R}\sum_{(u,q)\in A_p} \alpha_{p,q}^m \times \left(\chi_{p,q}^m - \tau_{u,q}^p\right)\right] \geq \theta$ and $\omega \leq N$, then
   let $\mu \leftarrow \sum_{(p,q)\in A_C}\sum_{m\in R}\sum_{(u,q)\in A_p} \left(\chi_{p,q}^m - \tau_{u,q}^p\right)^2$, $\omega \leftarrow \omega + 1$;
   let $\alpha_{p,q}^m \leftarrow \max\left\{0, \alpha_{p,q}^m + \sigma \times \left(\chi_{p,q}^m - \tau_{u,q}^p\right)/\mu\right\}$, $\forall m \in R$, $\forall (p,q) \in A_C$, $(u,q) \in A_p$;
   if any Lagrange multiplier is adjusted, then return to step 2;
   end if;
   end if.

Fig. 4. The algorithm based on Lagrangian relaxation.

first assign unit cost to each arc and then iteratively adjust the cost according to the subgradient algorithm. Let $W(\bar{\alpha})$ denote the objective function of LRP in Section 3.1, where $\bar{\alpha} = \left(\alpha_{p,q}^m, \forall m \in R, \forall (p,q) \in A_C\right)$. The subgradient corresponding to the optimal solution to LRP is denoted by $\nabla W(\bar{\alpha}) = \left(\partial W(\bar{\alpha})/\partial \alpha_{p,q}^m, \forall m \in R, \forall (p,q) \in A_C\right)$, where

$$\frac{\partial W(\bar{\alpha})}{\partial \alpha_{p,q}^m} = \chi_{p,q}^m - \tau_{u,q}^p.$$

The feasible solution to the MOM routing problem obtained by an algorithm depends on cost $\alpha_{p,q}^m$, and the subgradient $\partial W(\bar{\alpha})/\partial \alpha_{p,q}^m$ indicates the direction of adjusting $\alpha_{p,q}^m$ to find an improved solution to the MOM routing problem, with $\forall m \in R$, and $\forall (p,q) \in A_C$. At each iteration, our algorithm increases or decreases the value of each cost according to the solutions to the two subproblems of LRP. Our algorithm increases $\alpha_{p,q}^m$ when $\chi_{p,q}^m - \tau_{u,q}^p$ is positive. In other words, arc (p,q) is used in the shortest path for m, but q is not selected in LSP for p in this case. On the other hand, our algorithm decreases $\alpha_{p,q}^m$ when $\chi_{p,q}^m - \tau_{u,q}^p$ is negative. In this case, arc (p,q) is not used in the shortest path for m, but q is selected in LSP for p.

We explain the adjustment of the cost in an intuitive way as follows. The solution to LSP provides insights to find a bandwidth-efficient tree, even though it may not be a feasible solution to the MOM routing problem. The dynamic programming algorithm for LSP tends to select the leaf nodes that share a longer common path to the root to reduce the net cost of the tree. In other words, for each member p and each leaf node q selected in LSP, the tree over which p relays data to q is a bandwidth efficient tree. Therefore, if p does not relay data to q in the MOM tree, then we decrease $\alpha_{p,q}^m$ for each member m such that the arc is more likely to be selected in the shortest path problem for m, and member p tends to relay data to q in the MOM tree. Therefore, the cost $\alpha_{p,q}^m$, which is the Lagrange multiplier in LRP, plays an important role in finding a bandwidth efficient MOM tree.

Fig. 4 shows the details of our algorithm for the MOM routing problem. Our algorithm assigns unit cost to each arc for each receiver in step 1. Then, our algorithm iteratively adjusts the cost $\alpha_{p,q}^m$ of each arc (p,q) for each receiver m to reroute the MOM tree. At each iteration, we first find the solutions to the shortest path problem and LSP and then find ϕ_{XOM}, which is the total bandwidth consumption in the MOM tree, in step 2 according to the solution to the shortest path problem for each receiver. We adjust the cost of each arc for each receiver in step 3 such that we can find the MOM tree with less bandwidth consumption at the next iteration. Our algorithm stops when the number of iterations ω is larger than a threshold N, when our algorithm can no longer adjust the cost, or when the difference of the total bandwidth consumption of our MOM tree and the lower bound on the total bandwidth consumption in the MOM-OPT is within a threshold θ. The parameter σ in step 3 is a parameter that dominates the modification of the Lagrange multipliers at each iteration of the subgradient algorithm. The MOM tree improves faster with a larger σ, but the obtained MOM tree tends to consume more bandwidth than the obtained tree with smaller σ. Therefore, we reduce the value of σ as the improvement of the MOM tree becomes smaller.

We illustrate this with an example in Fig. 5, where the corresponding IP network is shown in Fig. 1a. At each iteration, the three numbers in parentheses beside each arc in the overlay network are the costs of the arc for receivers 6, 11, and 13 in the shortest path problem. The solid line, long-dash line, and short-dash line in the overlay network are the shortest paths for receivers 6, 11, and 13, respectively. The number beside each member in the IP network is the profit of the member in LSP. For example, the profit of receiver 6 in T_1 is 2.4 at the second iteration, whereas the profits of receiver 11 in T_1 and T_6 are 3.3 and 2.1, respectively. The solid tree and the short-dash tree in the IP network indicate the selected leaf nodes in T_1 and T_6, respectively. For example, receivers 11 and 13 are selected in T_6 at the second iteration. Receivers 6, 11, and 13 are selected in T_1 at the fifth iteration, but no leaf node is selected in T_6. At the first iteration, the shortest paths for receivers 6, 11, and 13 contain arcs (1, 6), (1, 11), and (1, 13), respectively, in the overlay network, and the three arcs decide the routing of the MOM tree. Both receivers 11 and 13 are selected in LSP for T_6 because the total profit of receivers 11 and 13 is larger than the total bandwidth consumption in T_6 to serve the two receivers. The solution to LSP suggests that member 6 relays data to receivers 11 and 13 in the bandwidth-efficient MOM tree. In contrast, receivers 11 and 13 are not selected in LSP for T_1, and the solution does not suggest that sender 1 delivers data to receivers 11 and 13. Therefore, our algorithm reduces the costs of arcs (6, 11) and (6, 13) and increases the costs of arcs (1, 11) and (1, 13) at the end of the first and second iterations. At the third iteration, member 6 begins relaying data to receivers 11 and 13, and sender 1 stops sending data to the two receivers. The fourth and fifth iterations adjust the costs of the arcs in the overlay network such that the MOM tree uses the arcs corresponding to the selected leaf nodes in LSP at the sixth iteration. In other words, the shortest paths for receivers 6, 11, and 13 at the sixth iteration include only the selected leaf nodes of LSP for T_1 and T_6. For example, arc (6, 11) is in the shortest path for receiver 11, whereas receiver 11 is also selected in LSP for T_6. Therefore, our algorithm stops after the sixth iteration, since step 3 of our algorithm in Fig. 4 can no longer modify the cost of each arc.

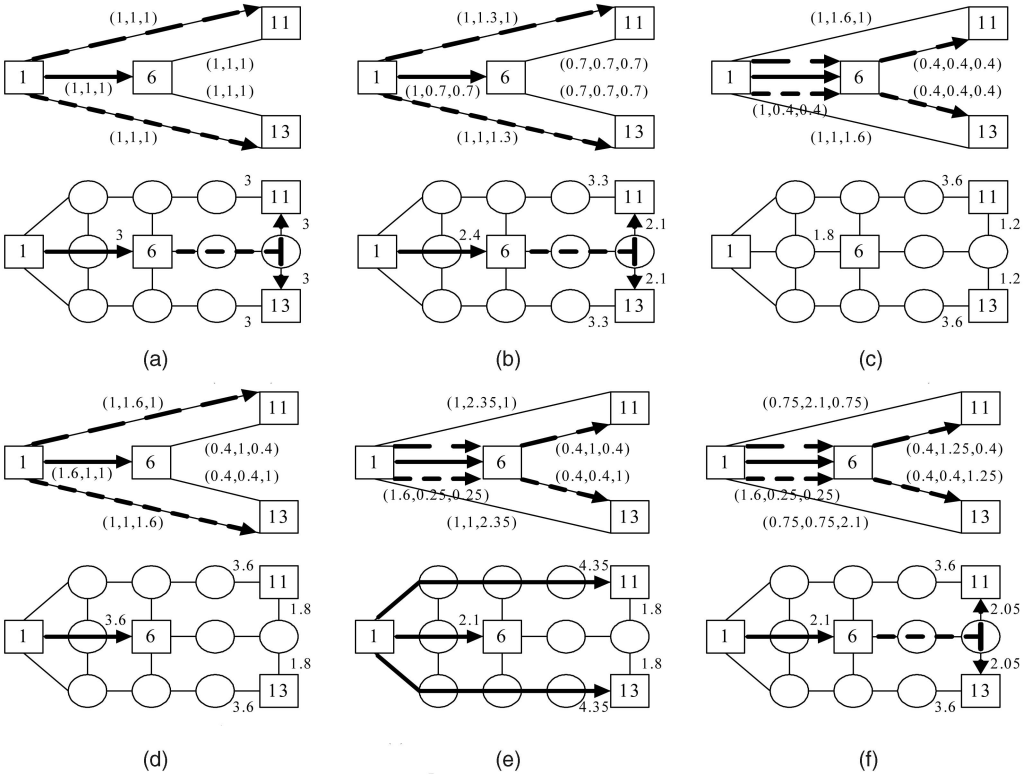

Fig. 5. An example of our algorithm, with δ being 2. (a) Iteration 1 $(\sigma = 3)$. (b) Iteration 2 $(\sigma = 3)$. (c) Iteration 3 $(\sigma = 3)$. (d) Iteration 4 $(\sigma = 3)$. (e) Iteration 5 $(\sigma = 2)$. (f) Iteration 6 $(\sigma = 1)$.

We design a protocol that supports dynamic group membership to implement the proposed algorithm in a distributed manner. Each new member in our protocol joins the multicast tree via the shortest path to the sender to connect to any on-tree parent member in the path, as in IP multicast. Our protocol has low initial setup delay because the data delivery starts once the new member is connected to the tree. To reduce the bandwidth consumption, our protocol has a rerouting procedure that distributes the Lagrange multipliers to allow each member to find the new parent member. The rerouting procedure does not lead to packet losses because each member leaves its original parent only after it has been successfully connected to the new parent member. In addition, when a member decides to leave the group, it is disconnected from its parent only after each child member has been connected to another member. The rerouting procedure is initiated by the sender, and thus, the sender can control the rerouting speed by adjusting the interval between two rerouting procedures. Our protocol is loop free, as long as the corresponding shortest path routing protocol is loop free.

4 Simulation

This section shows our simulation results for the MOM routing problem. In this simulation, we measure the total number of packets sent in the Steiner tree, the shortest-path tree, the minimum spanning tree, and the MOM tree generated by our algorithm (MOM-LAG). In addition, we compare the above with the MOM-OPT in small networks or with the lower bound on the number of packets in the MOM-OPT (MOM-LB) in large networks. We do not compare the MOM tree with the overlay multicast trees proposed in previous works, since the minimum spanning tree is the optimal overlay multicast tree. Note that the minimum spanning tree is the same as the MOM-OPT, with δ equal to 1.

In the simulation, we use the flat graph with the Waxman distribution [22] as the network topology to test our algorithm in networks with different graph characteristics. We also use MBone [23] and the Internet topology with the power-law distribution generated by Inet [24], [25] to verify our algorithm in more realistic networks. In our simulation, each node is a router, and each member is a host randomly attached to a router. The simulation results are averaged over 100 samples. The input parameters are listed as follows:

1. *Graph characteristic.* We generate random flat graphs with the Waxman distribution. Given the physical locations of two nodes, the distribution determines whether there exists a link connecting the two nodes. The distribution has two parameters: α and β. The graph with larger α and β has a larger node degree and a smaller graph diameter [26]. Therefore, the multicast trees in different graphs have different characteristics.
2. *Group size.* The group size is the number of receivers in a multicast tree.
3. δ. This is the maximum number of addresses that can be included in each packet.
4. *Distribution of receivers.*

We use the affinity-and-disaffinity model [27] to describe the distribution of receivers in a multicast tree. With a positive affinity index, receivers tend to cluster. With a negative affinity index, receivers tend to spread out. When the affinity index is zero, all receivers are chosen uniformly

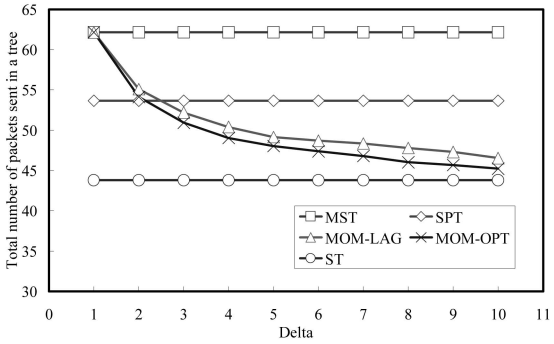

Fig. 6. Comparison of the bandwidth consumption for different multicast trees in the small network.

at random among all nodes. We measure the following performance metrics in our simulation:

1. cost of a multicast tree,
2. number of packets sent by each member,
3. stress of each link, which is the number of packets sent per link,
4. number of addresses in the header of each packet,
5. depth of a multicast tree, and
6. distance from each member to the parent member (this is the number of hops from the parent member to each member).

To find optimal solutions, we first adopt a small network with 30 nodes in Fig. 6 because solving a large ILP problem is computationally infeasible. We use CPLEX [28] with the ILP formulation and our proposed formulation to find the Steiner tree and the MOM-OPT. Fig. 6 compares the total number of packets sent in the Steiner tree, the shortest-path tree, the minimum spanning tree, the MOM-OPT, and the MOM-LAG when the group size is 20. Parameters α and β are both set to 0.28 in the Waxman distribution, and the affinity index is zero. Fig. 6 shows that the minimum spanning tree and the MOM tree with δ less than 2 consume more bandwidth than the shortest-path tree because some links need to send identical packets multiple times. However, the total bandwidth consumption in the MOM-OPT outperforms, that is, is smaller than, the shortest-path tree and is close to the Steiner tree as δ increases, and the total bandwidth consumption in the MOM-LAG approaches the MOM-OPT. Fig. 6 also shows that the MOM-OPT consumes about 20 percent and 30 percent less bandwidth than the shortest-path tree and minimum spanning tree, respectively.

Fig. 7 shows the simulation results with different δ's and group sizes in a large network with 100 nodes. Parameters α and β are both set to 0.2 in the Waxman distribution, and the affinity index is zero. Since finding the MOM-OPT with our formulation in the large network is computationally infeasible, we compare the MOM-LAG with the MOM-LB on the optimal solution. We find the lower bound by solving LRP, as defined in Section 3. Fig. 7a shows that the MOM tree requires less bandwidth than the minimum spanning tree and the shortest-path tree, and a small δ can effectively achieve the bandwidth reduction. Each multicast tree consumes more bandwidth as the group size increases, as shown in Fig. 7b. Fig 7b also shows that MOM can save more bandwidth as the group size increases. Fig. 7c compares the maximum number of packets sent per link

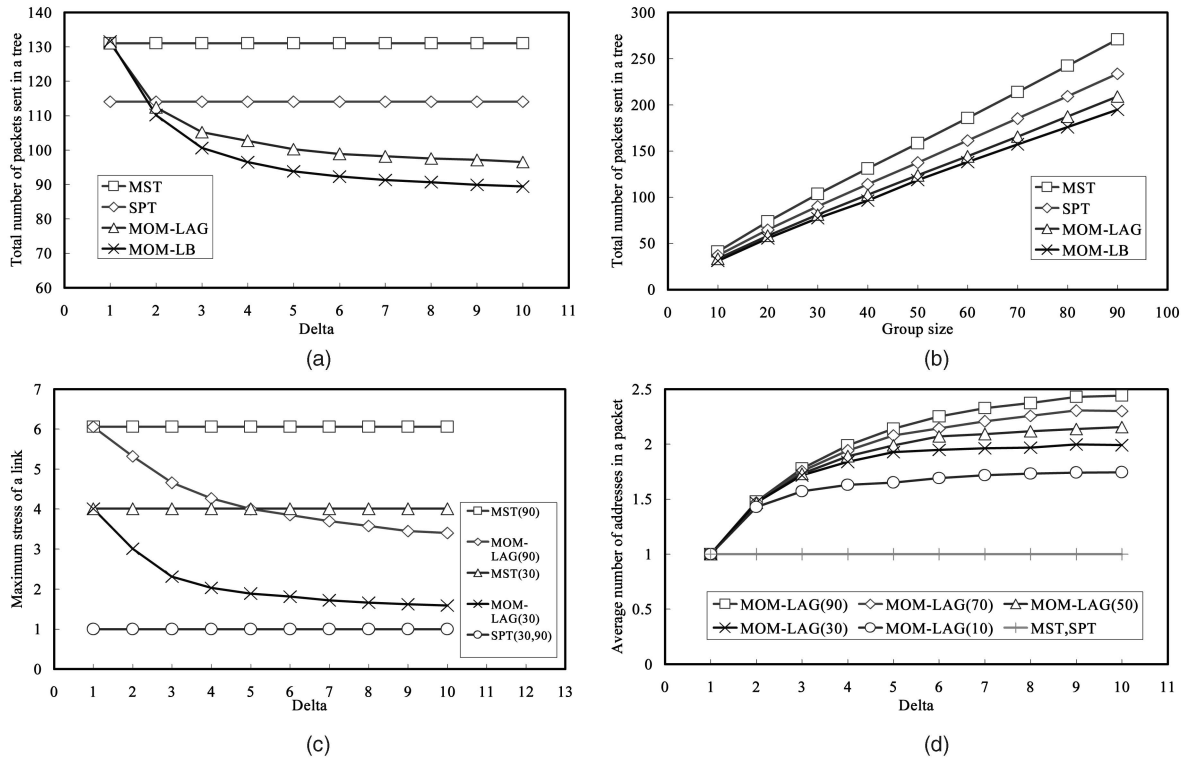

Fig. 7. Comparison of different multicast trees in the large network with different δ's and group sizes. (a) Bandwidth cost of a tree $(\text{group size} = 40)$. (b) Bandwidth cost of a tree ($\delta = 4$). (c) Stress of a link. (d) Number of addresses in a packet.

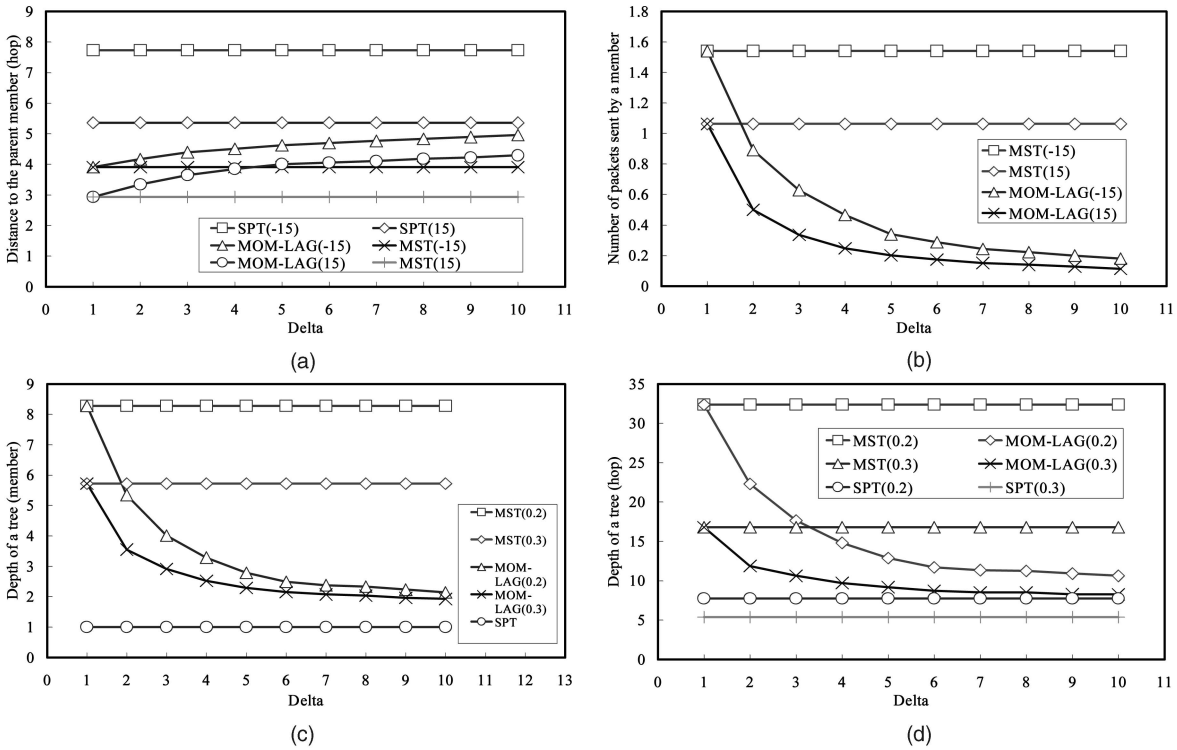

Fig. 8. Comparison of different multicast trees in a large network with different δ's, affinity indices, and (α, β) in the Waxman distribution, where the group size is 40. (a) Distance to the parent member. (b) Number of packets sent by a member. (c) Depth of a tree in terms of the number of members. (d) Depth of a tree in terms of the number of hops.

in different multicast trees, where the number in parentheses is the group size. Each link in the shortest-path tree, that is, the IP multicast tree, must deliver exactly one packet. Fig. 7c shows that each link in the MOM tree sends fewer packets than the link in the minimum spanning tree. Besides that, a small δ can effectively reduce the maximum number of packets sent per link. Fig. 7d compares the average number of addresses in each packet. Each packet in the overlay multicast and the IP multicast must contain one receiver address. Although MOM requires more addresses in each packet, Fig. 7d shows that the average number of addresses in each packet is limited, even when we have a large δ and a large group size.

Fig. 8 shows the simulation results with different δ's, affinity indices, and (α, β) in the Waxman distribution with 100 nodes in the network. We vary the affinity index in Figs. 8a and 8b. With a larger affinity index, the members tend to cluster together, leading to a small multicast tree. Fig. 8a compares the distance to the parent member in different multicast trees. The parent member in the shortest-path tree is just the root of the tree, and the shortest-path tree has the largest distance because there is no relaying member between the root and each member. Fig. 8a also shows that the distance increases as δ increases in MOM because each member can include more addresses in the packet and therefore can serve more distant members; Fig. 8b shows that each member in MOM delivers fewer packets as δ increases. The average number of packets sent by a member may be less than 1 because the members that are the leaf nodes in the multicast tree send no packet. Figs. 8c and 8d compare the depths of different multicast trees in terms of the number of members and the number of hops. We vary parameters α and β as 0.2 or 0.3 in the Waxman distribution to simulate the multicast trees in graphs with different characteristics. With larger α and β, each node has more incident links and tends to connect to more neighbor nodes, and the network has a larger node degree and a smaller graph diameter. Therefore, each node in a multicast tree has more child nodes, and the tree has a smaller depth.

Fig. 9 shows the simulation results in MBone and the Internet topology generated by Inet. We generate the Internet topology with the number of nodes identical to MBone to compare the multicast trees in two graphs with different characteristics, where each network has 4,177 nodes. In the Internet, there are a few nodes with large degrees [24], and these nodes tend to act as the backbone routers. Therefore, the graph that represents the Internet has a smaller diameter as compared to MBone, and we have smaller multicast trees in the graph, as shown in Figs. 9a and 9b. A multicast tree in the Internet tends to span the nodes with the backbone routers, and we have a multicast tree with a smaller depth, since each node in the tree tends to have more child nodes. Thus, Fig. 9c shows that a few links that connect to the backbone routers deliver a large number of identical packets in the Internet for overlay multicast, and MOM with a small δ can effectively reduce the redundancy of the packet delivery. Fig. 9d shows that the number of iterations in our algorithm is controllable. Parameter σ is 1 or 10 in this simulation. Our algorithm with a large σ converges with about 11 iterations. Note that it is possible that at some iterations, our algorithm obtains a solution worse than the solution of an iteration. The reason behind searching toward a locally worse direction is to avoid being trapped in a locally optimal solution. Our algorithm with a small σ in Fig. 9d converges more slowly and obtains a

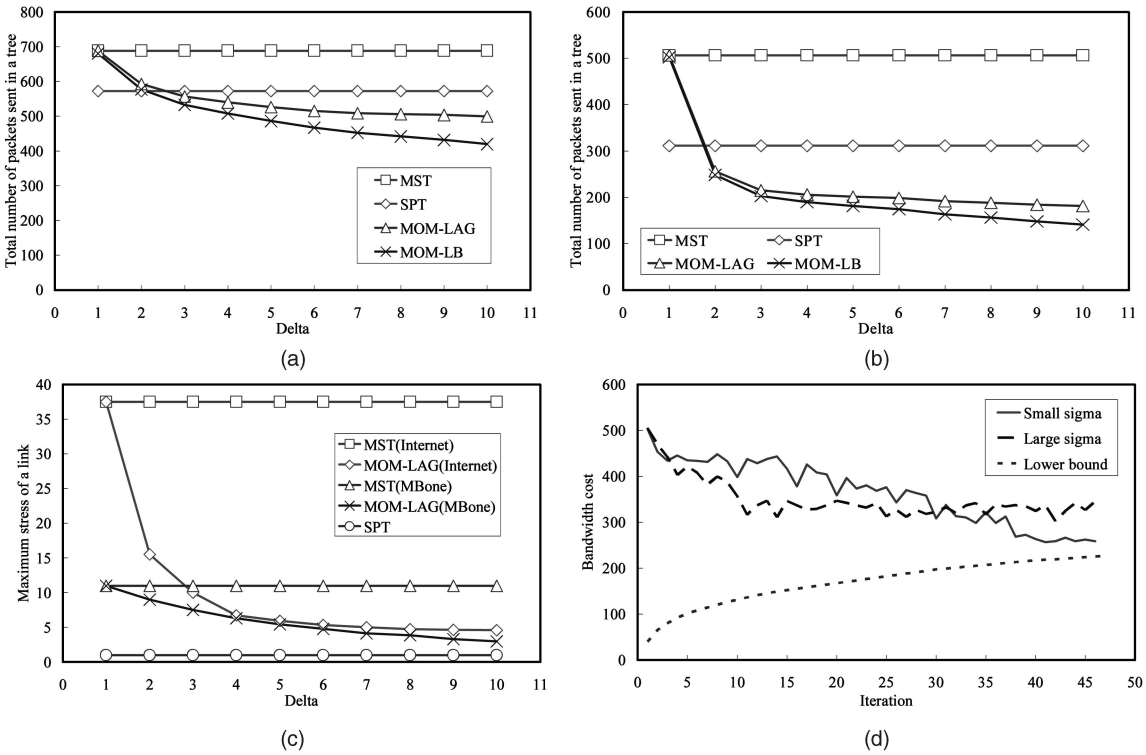

Fig. 9. Comparison of different multicast trees in MBone and the Internet topology generated by Inet, where the group size is 120. (a) Bandwidth cost of a tree in MBone. (b) Bandwidth cost of a tree in the Internet. (c) Stress of a link. (d) Bandwidth cost of a tree in the Internet.

solution at the 45th iteration, which is better than the one obtained at the 11th iteration with a large σ. Therefore, the number of iterations of our algorithm is controllable, and the sender can find the trade-off between the overhead and the quality of the solution with different σ's.

5 Conclusion

In this paper, we propose a new multicast delivery mechanism MOM for bandwidth-demanding applications. Our mechanism uses less bandwidth compared with both IP multicast and overlay multicast. The routing of our mechanism is more flexible than the IP multicast, and our mechanism avoids the stress problem in the overlay multicast. The bandwidth used in our mechanism is close to that used in Steiner trees. We model the MOM routing problem as an optimization problem with ILP. We design an algorithm based on Lagrangian relaxation. Our mechanism uses less network bandwidth than the IP multicast and overlay multicast. Moreover, our mechanism uses less interface bandwidth than the overlay multicast, thanks to including multiple addresses in each packet. Therefore, our mechanism is bandwidth efficient for both network operators and users.

Acknowledgments

This work was supported in part by the National Science Council (NSC), Taiwan, under a Center Excellence Grant NSC95-2752-E-002-006-PAE, and in part by NSC under Grant Numer NSC95-2221-E-002-066.

References

[1] P.V. Mieghem, G. Hooghiemstra, and R. Hofstad, "On the Efficiency of Multicast," *IEEE/ACM Trans. Networking,* vol. 9, no. 6, pp. 719-732, Dec. 2001.

[2] D. Waitzman, C. Partridge, and S. Deering, *Distance Vector Multicast Routing Protocol,* IETF RFC 1075, Nov. 1988.

[3] J. Moy, "Multicast Routing Extensions for OSPF," *Comm. ACM,* vol. 37, no. 8, pp. 61-66, Aug. 1994.

[4] T. Ballardie, P. Francis, and J. Crowcroft, "Core-Based Trees (CBT)," *ACM SIGCOMM Computer Comm. Rev.,* vol. 23, no. 4, pp. 85-95, 1993.

[5] S. Deering et al., "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Trans. Networking,* vol. 4, no. 2, pp. 153-162, Apr. 1996.

[6] L. Aguilar, "Datagram Routing for Internet Multicasting," *Proc. ACM SIGCOMM '84,* vol. 14, no. 2, pp. 58-63, 1984.

[7] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, "Explicit Multicast (Xcast) Concepts and Options," IETF internet draft, work in progress, Jan. 2007.

[8] C. Graff, *IPv4 Option for Sender Directed Multi-Destination Delivery,* IETF RFC 1770, 1995.

[9] L. Ji and M.S. Corson, "Explicit Multicasting for Mobile Ad Hoc Networks," *ACM Mobile Networks and Applications,* vol. 8, no. 5, pp. 535-549, Oct. 2003.

[10] C. Gui and P. Mohapatra, "Scalable Multicasting in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM '04,* vol. 3, pp. 2119-2129, 2004.

[11] D.-N. Yang and W. Liao, "Optimizing State Allocation for Multicast Communications," *Proc. IEEE INFOCOM '04,* vol. 4, pp. 2719-2730, 2004.

[12] M.-K. Shin, K. Kim, D.-K. Kim, and S.-H. Kim, "Multicast Delivery Using Explicit Multicast over IPv6 Networks," *IEEE Comm. Letters,* vol. 7, no. 2, pp. 91-93, Feb. 2003.

[13] Y.-H. Chu, S.G. Rao, S. Seshan, and H. Zhang, "A Case for End-System Multicast," *IEEE J. Selected Areas in Comm.,* vol. 20, no. 8, pp. 1456-1471, Oct. 2002.

[14] J. Liebeherr, M. Nahas, and W. Si, "Application-Layer Multicasting with Delaunay Triangulation Overlays," *IEEE J. Selected Areas in Comm.,* vol. 20, no. 8, pp. 1472-1488, Oct. 2002.

[15] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure," *IEEE J. Selected Areas in Comm.,* vol. 20, no. 8, pp. 1489-1499, Oct. 2002.
[16] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," *Proc. ACM SIGCOMM '02,* vol. 32, no. 4, pp. 205-217, 2002.
[17] M. Castro et al., "An Evaluation of Scalable Application-Level Multicast Built Using Peer-To-Peer Overlays," *Proc. IEEE INFOCOM '02,* vol. 2, pp. 1510-1520, 2003.
[18] S.Y. Shi and J.S. Turner, "Multicast Routing and Bandwidth Dimensioning in Overlay Networks," *IEEE J. Selected Areas in Comm.,* vol. 20, no. 8, pp. 1444-1455, Oct. 2002.
[19] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an Efficient Overlay Multicast Infrastructure for Real-Time Applications," *Proc. IEEE INFOCOM '03,* vol. 2, pp. 1521-1531, 2003.
[20] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal Resource Allocation in Overlay Multicast," *Proc. 11th IEEE Int'l Conf. Network Protocols (ICNP '03),* pp. 71-81, 2003.
[21] E. Brosh and Y. Shavitt, "Approximation and Heuristic Algorithms for Minimum-Delay Application-Layer Multicast Trees," *Proc. IEEE INFOCOM '04,* vol. 4, pp. 2697-2707, 2004.
[22] B.M. Waxman, "Routing of Multipoint Connections," *IEEE J. Selected Areas in Comm.,* vol. 6, no. 9, pp. 1617-1622, Dec. 1988.
[23] USC/ISI SCAN project, http://www.isi.edu/scan/mbone.html, 1999.
[24] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network Topology Generators: Degree-Based vs. Structural," *Proc. ACM SIGCOMM '02,* vol. 32, no. 4, pp. 147-159, 2002.
[25] Inet Topology Generator, http://topology.eecs.umich.edu/inet/, 2002.
[26] E.W. Zegura, K.L. Calvert, and M.J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," *IEEE/ACM Trans. Networking,* vol. 5, no. 6, pp. 770-783, Dec. 1997.
[27] G. Phillips, S. Shenker, and H. Tangmunarunkit, "Scaling of Multicast Trees: Comments on the Chuang-Sirbu Scaling Law," *Proc. ACM SIGCOMM '99,* vol. 29, no. 4, pp. 41-51, 1999.
[28] CPLEX Optimization Package, http://www.ilog.com/products/cplex/, 2007.
[29] D.-N. Yang, "Scalability in Xcast-Based Multicast," PhD dissertation, Nat'l Taiwan Univ., 2004.

De-Nian Yang received the BS and PhD degrees from the Department of Electrical Engineering, National Taiwan University, Taipei, in 1999 and 2004, respectively. He is currently a postdoctoral researcher for the military services in the Department of Electrical Engineering, National Taiwan University. His research interests include network planning, multicasting, and quality of service (QoS) in wireless networks. He is a member of the IEEE. He received the Best Student Paper Award from the First IEEE International Conferences on Multimedia and Expo (ICME) in 2000.

Wanjiun Liao received the BS and MS degrees from the National Chiao Tung University, Taiwan, in 1990 and 1992, respectively, and the PhD degree in electrical engineering from the University of Southern California, Los Angeles, in 1997. She joined the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, as an assistant professor in 1997. Since August 2005, she has been a full professor. She is currently an associate editor of the *IEEE Transactions on Wireless Communications* and the *IEEE Transactions on Multimedia*. She was a tutorial cochair of IEEE INFOCOM 2004, was the technical program committee (TPC) vice chair of 2005 IEEE Global Telecommunications Conference (GLOBECOM) Autonomous Network Symposium, and is the TPC cochair of 2007 IEEE GLOBECOM General Symposium. Her research interests include wireless networks, multimedia networks, and broadband access networks. She has received many research awards. Papers she coauthored with her students received the Best Student Paper Award from the First IEEE International Conferences on Multimedia and Expo (ICME) in 2000 and the Best Paper Award from the First International Conference on Communication, Circuits and Systems (ICCCAS) in 2002. She was awarded K.T. Li Young Researcher Award of the ACM for her research achievements in 2003. She was elected as one of the Ten Distinguished Young Women in Taiwan in 2000. She is a senior member of the IEEE.