

行政院國家科學委員會專題研究計畫 成果報告

多媒體共享之網路服務平台--子計畫三：多媒體共享網路 之群播通訊研究(3/3) 研究成果報告(完整版)

計畫類別：整合型
計畫編號：NSC 95-2221-E-002-066-
執行期間：95年08月01日至96年07月31日
執行單位：國立臺灣大學電信工程學研究所

計畫主持人：廖婉君

計畫參與人員：博士班研究生-兼任助理：郭嘉駿
碩士班研究生-兼任助理：施兆芳、林怡君、姚立偉

報告附件：國外研究心得報告
出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 96 年 07 月 13 日

行政院國家科學委員會專題研究計劃成果報告

多媒體共享之網路服務平台-

子計畫三：多媒體共享網路之群播通訊研究(3/3)

計劃編號：NSC 95-2221-E-002-066

執行期限：95年8月1日至96年7月31日

主持人：廖婉君博士 國立台灣大學電機工程學系

計畫參與人員：郭嘉駿、施兆芳、林怡君、姚立偉

國立台灣大學電機工程學系

摘要 — 本年度計畫主要研究多媒體內容傳輸網路 (Multimedia Content Delivery Network) 中群播樹之路由演算法的設計。我們所針對的應用是要求高頻寬需求服務。目前相關研究大都是結合史坦爾樹 (Steiner tree) 演算法以及重新路由 (rerouting) 機制，這類作法的主要缺點是其所產生的群播樹會使用較多的網路連結 (link)，因而較浪費網路資源。在本年計畫中我們提出演算法改進此缺點，並結合 IP multicast 及 overlay network 優點，藉由模擬驗證，我們所提出的演算法使用較少網路資源。此外，我們亦分析所提出的演算法在壞情況下之效能，並討論延申此演算法以支援動態群組的可能性。

關鍵詞：多媒體內容傳輸網路、群播

Abstract—In this project, we propose a new multicast delivery mechanism for bandwidth-demanding applications in IP networks. Our mechanism, Multiple-destination Overlay Multicast, combines the advantages of IP multicast and overlay multicast. We formulate the Multiple-destination Overlay Multicast routing problem as an optimization problem, which is NP-hard. We then design an algorithm based on Lagrangean relaxation on our formulation and propose a distributed protocol based on the algorithm. For network operators, our mechanism uses less network bandwidth than both IP multicast and overlay multicast. For end users, our mechanism uses less interface bandwidth than overlay multicast.

Keywords- *multicast; application-layer multicast; overlay multicast.*

I. INTRODUCTION

Multicast is an efficient way of one-to-many and many-to-many communications. Each multicast group consists of a set of members, and each member can be a sender or receiver of the group. The sender for a multicast

group delivers data in a multicast tree to all receivers of the group. For simplicity and robustness, current IP multicast routing protocols rely on unicast routing protocols, which find the shortest path from a sender to a receiver, to construct a multicast tree in the IP network. Therefore, an IP multicast tree is a shortest-path tree, which is the 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. 1 (a) is an IP network topology. Node 1 is the root, and nodes 6, 11, and 13 are the group members. With the shortest path tree (Fig. 1 (b)), it takes ten packet-hops for transmissions, where a packet hop corresponds to sending a packet over one hop. The path in the shortest-path tree 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, *overlay multicast*, i.e., application-layer multicast, has been proposed to construct a multicast tree in an overlay network. The overlay network can be considered as a complete graph in which only the members of a multicast group can be the nodes. Each edge in the overlay network corresponds to a shortest-path in the IP network, connecting the two

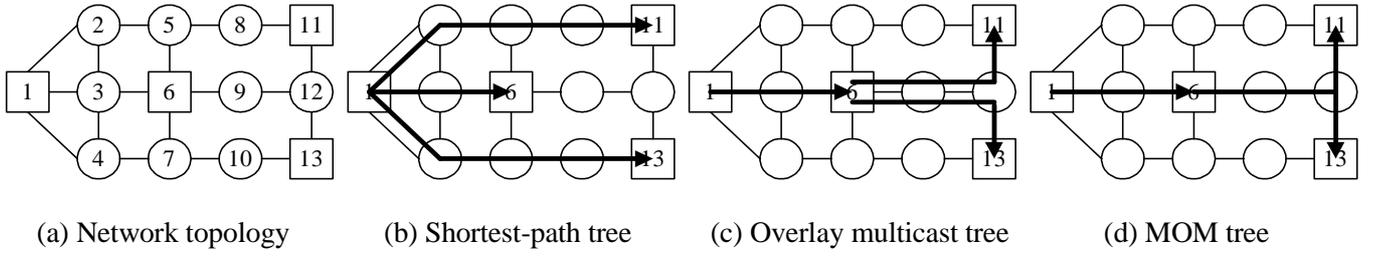


Fig. 1. An example comparing the bandwidth consumption in a shortest-path tree (10 packets), an overlay multicast tree (8 packets), and an Multiple-Destination Overlay Multicast tree (6 packets).

terminal nodes of the edge. Therefore, each member in the overlay network delivers data to any other member only in unicast, and the network is not required to support IP multicast. An overlay multicast tree is a spanning tree in the 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 is the total bandwidth consumption of all edges in the tree, where the bandwidth consumption of an edge in the overlay network is total bandwidth consumption of all links in the corresponding shortest path of 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 of an overlay multicast tree is more flexible than the routing of 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. 1 (c), there are eight packet-hops in the overlay multicast tree. The links from node 6 to 12 send 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. Therefore, 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*, i.e., the bandwidth for a member to send data to the Internet, if the member is required to send an identical packet multiple times to other members. For some members, the required interface bandwidth may exceed the subscribed interface bandwidth. Previous

research results regard the above problem as the *stress problem* in overlay multicast.

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 the following reasons. First, to find a Steiner tree in a graph is NP-complete **錯誤! 找不到參照來源**. To find a Steiner tree is much more computationally expensive than to find a shortest-path tree in the IP network or a minimum spanning tree in the overlay network. Due to the limited computational power in routers, the IP multicast and overlay trees adopt the shortest paths to support a large number of multicast groups. Second, 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 contrast, shortest-path computation and delivery is available in all routers, facilitating fast deployment of IP multicast and overlay multicast in the Internet. Third, under the current host group model **錯誤! 找不到參照來源**, members are allowed to join or leave a group dynamically at any time. In other words, routers need to find a new Steiner tree for a multicast group whenever a member joins or leaves the group. Therefore, the large overhead incurred by frequently computing a large number of Steiner trees is infeasible for today's routers.

In this paper, we propose a bandwidth-efficient multicast delivery mechanism, Multiple-destination Overlay Multicast (MOM), that adopts the routing

flexibility of overlay multicast but alleviates the stress problem by including multiple destination addresses in each IP header [20][21]. With MOM, a router uses the existing unicast routing protocol with the receivers' addresses 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 (e.g., [22-25]).

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. 1 (d) 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 in each link.
- For end users, MOM consumes less interface bandwidth than overlay multicast. MOM uses less interface bandwidth 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 show that the above two statements hold for all cases by proving that IP multicast and overlay multicast are two special cases of MOM.

In this paper, we model the Multiple-destination Overlay Multicast (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 [36]. We design an algorithm based on Lagrangean relaxation [錯誤! 找不到參照來源。](#) on our proposed ILP formulation. We adopt Lagrange relaxation, instead of other optimization techniques, due to the following reasons. First, our algorithm decomposes the original problem into multiple sub-problems such that each sub-problem can be solved by each sender

and receivers 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. Since using the ILP formulation to find the optimal MOM tree with a large number of members in large IP network is computationally infeasible, the lower bound obtained by our algorithm provides the benchmark to compare with other algorithms for the problem. Please note that we also design a 2-approximation algorithm for this problem in [錯誤! 找不到參照來源。](#), but the algorithm requires centralized computation and is not suitable to be implemented in the distributed IP network.

Compared with the algorithms to find 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 the members are more suited than the 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 computing power to collaboratively find the tree for each participating group. In addition, MOM does not require any new routing algorithm, and ISPs only need to enable the multi-destination delivery in routers, which is a standard in IPv4 as defined in [10]. We also design a protocol, which is described briefly at the end of Section III, based the proposed algorithm, and the protocol can be implemented as a client middleware that can be accessed by any application software, such as Bit Torrent, for overlay multicast services. Therefore, compared with Steiner tree algorithms, we believe that MOM is more suitable for bandwidth demanding applications in the Internet.

MOM is applied to applications different from IP multicast and overlay multicast. Its design goal is to minimize the bandwidth consumption, while IP multicast is to support a large number of members. Previous research results for overlay multicast focus mainly on scalability (i.e., to support more groups) [錯誤! 找不到參照來源。](#)–[錯誤! 找不到參照來源。](#) or on real-time applications to minimize end-to-end delay [錯誤! 找不到參照來源。](#)–[錯誤! 找不到參照來源。](#)

TABLE I. 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

參照來源。 The approach to improving scalability is 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, in this paper, we focus 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 II addresses the problem description and our proposed ILP formulation. Section III explains our algorithm based on the Lagrangean relaxation on our ILP formulation. Section IV presents the simulation results. Finally, this paper is concluded in Section V.

II. PROBLEM DESCRIPTION

In this paper, we focus on one-to-many multicast communications, such as the Source-Specific Multicast (SSM) 錯誤! 找不到參照來源。 The members in a multicast group contain one sender and at least one receiver. The root of a multicast tree is the sender of the group, and other members are the receivers of the group. We formulate the Multiple-destination Overlay Multicast 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 multiple receiver addresses. Since the forwarding delay of a packet in the router is proportional to the number of addresses in the header 錯誤! 找不到參照來源。 , 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 Table I and Table II. Let ϕ_{SPT} , ϕ_{ST} , ϕ_{OM}^* , $\phi_{XOM}^*(\delta)$ denote the number of packets sent in the shortest-path tree, Steiner tree, optimal overlay multicast tree, and optimal MOM tree with at most δ addresses in each packet, respectively. In the following, we 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,*

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

TABLE II. 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, $ D_{u,v}^p = \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, $U_{u,v}^p = D_{u,v}^p - K_{u,v}^p$

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. \square

Theorem 1. *The following inequalities hold,*

$$\phi_{\text{SPT}} \geq \phi_{\text{XOM}}^*(|R|), \phi_{\text{OM}}^* \geq \phi_{\text{XOM}}^*(\delta), \text{ and } \phi_{\text{XOM}}^*(\delta) \geq \phi_{\text{ST}},$$

where $\delta \geq 1$.

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 as $|R|$, and ϕ_{SPT} is no less than $\phi_{\text{XOM}}^*(|R|)$. The optimal overlay multicast tree is the minimum spanning tree, which is also the optimal solution to the MOM routing problem with δ as one. In other words, ϕ_{OM}^* is identical to $\phi_{\text{XOM}}^*(1)$. Therefore, ϕ_{OM}^* is no less than $\phi_{\text{XOM}}^*(\delta)$ according to Lemma 1, $\delta \geq 1$. Since any MOM tree with δ as $|R|$ is a feasible solution to the Steiner tree problem, $\phi_{\text{XOM}}^*(|R|)$ is no less than ϕ_{ST} . Therefore, $\phi_{\text{XOM}}^*(\delta)$ is no less than ϕ_{ST} according to Lemma 1, $\delta \geq 1$. \square

We formulate the MOM routing problem as an integer linear programming 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,

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

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

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

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

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

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

Constraints (1)–(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 $\tau_{u,v}^p$, the number of downstream members served by arc (u,v) . If member q is relayed by p , constraint (4) enforces that $\tau_{u,q}^p$ is one 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) therefore can 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 $\chi_{p,q}^m$, $\tau_{u,v}^p$, 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.

In the next section we design an algorithm based on Lagrangean relaxation on the above ILP formulation.

III. ALGORITHM BASED ON LAGRANGEAN RELAXATION

In this section, we design an algorithm based on Lagrangean 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 optimal MOM tree. 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 optimal MOM tree with the ILP formulation is computationally infeasible.

The algorithm relaxes a constraint of our formulation to transform the MOM routing problem into the *Lagrangean Relaxation Problem (LRP)*. The LRP has a new objective function with the *Lagrange multipliers* and fewer constraints such that we can decompose the LRP into multiple sub-problems, where each sub-problem corresponds to a member and can be solved independently by the member. The members in our algorithm collaboratively construct the MOM tree according to the solutions to the sub-problems. In addition, 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 to solve the MOM routing problem as follows.

- Transform the MOM routing problem into the Lagrangean Relaxation Problem (LRP).
- Decompose the LRP into multiple sub-problems and solve each sub-problem.
- Construct an MOM tree according to the solutions to the sub-problems.

- Reduce the total bandwidth consumption of the MOM tree by iteratively updating the Lagrange multipliers.

A. Problem Transformation and Decomposition

Our algorithm relaxes constraint (4) to transform the MOM routing problem into the LRP, and the objective function of the LRP is the Lagrange multiplier, $\alpha_{p,q}^m \geq 0$, $\forall m \in R$, $\forall (p, q) \in A_c$. The LRP includes constraints (1), (2), (3), (5), and (6). Compared with the objective function of the MOM routing problem, the objective function of the LRP owns a new term corresponding to the relaxed constraint (4). Intuitively, for any feasible solution to the 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 the LRP since the set of constraints of the 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 the LRP and the MOM routing problems, the objective value of the LRP must be no more than the objective value of the MOM routing problem because the new term in the objective function of the LRP must be non-positive. Therefore, the objective value of the optimal solution to the 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 the 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 optimal MOM tree.

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

Since the objective function of LRP can be divided into two terms, our algorithm decomposes the LRP into two sub-problems, where the objective function of the first sub-problem is expressed as follows,

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

The constraints of the first sub-problem include constraints (1)–(3). The first sub-problem 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 $\alpha_{p,q}^m$ 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 sub-problem is expressed as follows,

$$\min \sum_{p \in M} \left[\sum_{(u,v) \in A_p} \pi_{u,v}^p - \sum_{q \in L_p} \sum_{u: (u,q) \in A_p} \left(\sum_{m \in R} \alpha_{p,q}^m \right) \times \tau_{u,q}^p \right].$$

The constraints of the second sub-problem include constraints (5) and (6). We decompose the sub-problem into the *Leaf Selection Problem (LSP)* for each member p in M . For each member p in M , LSP is to select 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 $\sum_{m \in R} \alpha_{p,q}^m$ if it is selected, i.e., variable $\tau_{u,q}^p$ is one, 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 the 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 incur more bandwidth consumption in T_p . Therefore, we have to find the best trade-off to select the leaf nodes.

B. Solve the Leaf Selection Problem

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, $1 \leq k \leq \delta$. 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 $D_{u,v}^p$. 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, while every other packet must include δ addresses. For arc (u,v) in T_p , let $K_{u,v}^p$ 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 $U_{u,v}^p$ 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 $K_{u,v}^p$ and $U_{u,v}^p$ from $K_{v,w}^p$ and $U_{v,w}^p$ of each downstream arc (v,w) of (u,v) such that the size of $U_{u,v}^p$ is less than δ . More specifically, we pack the unpacked selected leaves of each downstream arc (v,w) to $K_{u,v}^p$ to reduce the size of $U_{u,v}^p$. Therefore, our algorithm is operated in a bottom-up manner.

For each member p and each arc (u,v) in T_p , let $\varepsilon_{u,v}^p(j_{u,v})$ denote the optimal net cost obtained in the sub-tree 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 sub-tree. We find $\varepsilon_{u,v}^p(j_{u,v})$ in the bottom-up manner as follows. We first consider the leaf node. For each leaf node q with u as the parent node in T_p , net cost $\varepsilon_{u,q}^p(1)$ is therefore $1 - \sum_{m \in R} \alpha_{p,q}^m$ because u is a selected leaf node in this case. In contrast, net cost $\varepsilon_{u,q}^p(0)$ is zero and corresponds to the case that u 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 $\varepsilon_{u,v}^p(j_{u,v})$ represents the case that $j_{u,v}$ child nodes $\{q_1, q_2, \dots, q_{j_{u,v}}\}$ are selected, and we obtain $\varepsilon_{u,v}^p(j_{u,v})$ as follows,

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

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

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

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

end for;

end for.

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

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

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

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 Leaf Selection Problem (LSP).

$$\varepsilon_{u,v}^p(j_{u,v}) = \begin{cases} 0, & j_{u,v} = 0, \\ 1 + j_{u,v} - \sum_{k=1}^{j_{u,v}} \sum_{m \in R} \alpha_{p,q_k}^m, & 1 \leq j_{u,v} < \delta, \end{cases} \quad (9)$$

where 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 $\varepsilon_{u,v}^p(j_{u,v})$ becomes

$$\varepsilon_{u,v}^p(j_{u,v}) = \begin{cases} \min \left\{ 0, |P_{p,v}| + \delta - \sum_{k=1}^{\delta} \sum_{m \in R} \alpha_{p,q_k}^m \right\}, & j_{u,v} = 0, \\ 1 + j_{u,v} - \sum_{k=1}^{j_{u,v}} \sum_{m \in R} \alpha_{p,q_k}^m, & 1 \leq j_{u,v} < \delta, \end{cases} \quad (10)$$

where $|P_{p,v}|$ 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 $\varepsilon_{u,v}^p(j_{u,v})$ selects the i child nodes $\{q_1, q_2, \dots, q_i\}$ with the largest profit such that $\varepsilon_{u,v}^p(j_{u,v})$ satisfies the following equation, where $\lceil j_{u,v}/\delta \rceil$ 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 equations (9) and (10) are two special cases of (11), and the above scenario considers only each node v with all child nodes as 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,

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 cost 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 running time to $O(\delta^2 \times |C_v^p|^2)$ with an auxiliary graph, as explained in the Appendix.

Fig. 2 gives our dynamic programming algorithm for LSP. Step 1 sets the initial values 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 four finds the selected leaf nodes and the net cost of the tree T_p . Consider the tree in Fig. 3 for example with δ as three, where the number

below each receiver is the profit associated with the receiver q , namely, $\sum_{m \in R} \alpha_{1,q}^m$. For each arc (u,v) in T_1 , Table III 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 $\varepsilon_{u,v}^p(j_{u,v})$. The $j_{u,v}$ in bold 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 $\varepsilon_{1,2}^1(0)$ 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 one, $(\bar{j}_{5,9}, \bar{j}_{5,10})$ 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.

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

Proof: For each arc (u,v) in T_p , we can obtain $\varepsilon_{u,v}^p(j_{u,v})$ for all $j_{u,v}$ in $O(\delta^2 \times |C_v^p|^2)$ according to Lemma 2. The computational time of our algorithm is $O\left(\sum_{v \in V_p} \delta^2 \times |C_v^p|^2\right)$. Therefore, our dynamic programming algorithm finds the optimal solution to LSP in $O(\delta^2 \times |V_p|^2)$. \square

C. Find and Improve 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 sub-problem 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. The advantage of adopting the shortest-path tree in the overlay network as the feasible solution to the MOM routing problem is that the members can find the tree in a distributed manner similar to the IP multicast routing protocols. Initially, the tree

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 $\phi_{\text{XOM}} - \left[\sum_{p \in M} \sum_{(u,v) \in A_p} \pi_{u,v}^p + \sum_{(p,q) \in A_c} \sum_{m \in R} \sum_{(u,q) \in A_p} \alpha_{p,q}^m \times (\chi_{p,q}^m - \tau_{u,q}^p) \right] \geq \theta$ and $\omega \leq N$, then
let $\mu \leftarrow \sum_{(p,q) \in A_c} \sum_{m \in R} \sum_{(u,q) \in A_p} (\chi_{p,q}^m - \tau_{u,q}^p)^2$, $\omega \leftarrow \omega + 1$;
let $\alpha_{p,q}^m \leftarrow \max \left\{ 0, \alpha_{p,q}^m + \sigma \times (\chi_{p,q}^m - \tau_{u,q}^p) / \mu \right\}$, $\forall m \in R$, $\forall (p,q) \in A_c$, $(u,q) \in A_p$;
if any Lagrange multiplier is adjusted, then return to step 2;
end if;
end if.

Fig. 4. The algorithm based on Lagrangean relaxation.

contains only the sender, and each member sequentially is connected to the tree with the shortest path to the sender. Note that the MOM tree is different from the IP multicast tree because the cost $\alpha_{p,q}^m$ of each arc (p,q) can be different for each receiver m . In other words, the routing of 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 first assign unit cost to each arc and then iteratively adjust the cost according to the subgradient algorithm [錯誤! 找不到參照來源](#), which is an algorithm based on the theory of Lagrangean relaxation to reduce the total bandwidth consumption. Let $W(\bar{\alpha})$ denote the objective function of LRP in Section III-A, where $\bar{\alpha} = (\alpha_{p,q}^m, \forall m \in R, \forall (p,q) \in A_c)$. The sub-gradient corresponding to the optimal solution of LRP is denoted by

$\nabla W(\bar{\alpha}) = (\partial W(\bar{\alpha}) / \partial \alpha_{p,q}^m, \forall m \in R, \forall (p,q) \in A_c)$, 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, $\forall m \in R, \forall (p,q) \in A_c$. At each iteration, our algorithm increases or decreases the value of each cost according to the solutions to the two sub-problems 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. Therefore, arc (p,q) is not used in the shortest path for m , but q is selected in LSP for p in this case.

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.

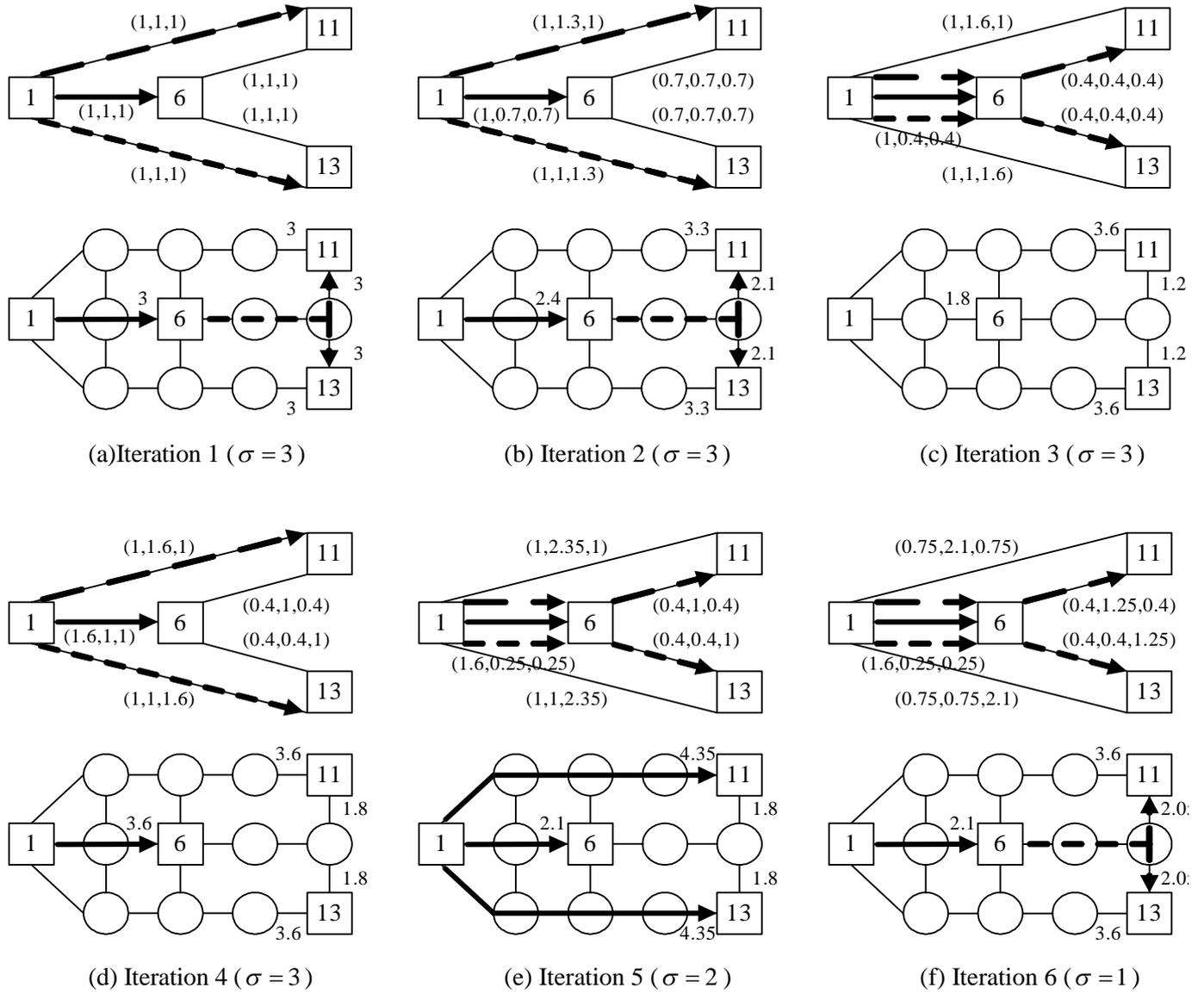


Fig. 5. An example of our algorithm with δ equal to two.

Therefore, if p does not relay data to q in the MOM tree, 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} , 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 optimal MOM tree 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. With a larger σ , the MOM tree improves faster, but the obtained MOM tree tends to consume more bandwidth than

the obtained tree with smaller σ . Therefore, in this paper, 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. 1 (a). At each iteration, the three numbers in the parenthesis beside each arc in the overlay network are the costs of the arcs 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, while 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 are larger than the total bandwidth consumption in T_6 to serve the two receivers. The solution for LSP suggests that member 6 should relay 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 suggests that sender 1 should not deliver 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, therefore, member 6 begins to relay 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, while 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.

We design a protocol which supports dynamic group membership to implement the proposed algorithm in a distributed manner. We briefly describe our protocol in the following. Please refer to [36] for more details. 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 the sender can thereby control the speed of the tree rerouting 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.

IV. 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 (ST), the shortest-path tree (SPT), the optimal overlay multicast tree (MST), and the MOM tree generated by our algorithm (MOM-LAG). In addition, we compare the above with the optimal MOM tree (MOM-OPT) in small networks or with the lower bound on the number of packets in the optimal MOM tree (MOM-LB) in large networks. Please note that both the lower bound and the MOM tree generated by our algorithm are obtained by the Lagrangean relaxation described in Section III, where the lower bound is the optimal objective value of LRP, and our MOM tree is modified from the solution to LRP. In addition, the MOM tree, when not explicitly stated as the optimal solution, refers to the solution obtain by our algorithm in the rest of this section.

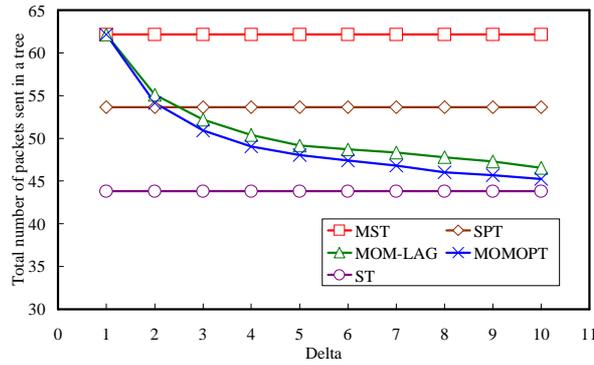


Fig. 6. Comparison of the bandwidth consumption of different multicast trees in the small network.

We do not compare the MOM tree with the overlay multicast trees proposed in previous work since the minimum spanning tree is the optimal overlay multicast tree. In other words, there is no overlay multicast tree in the previous work that can consume less bandwidth than the minimum spanning tree. In addition, it is not suitable to compare the MOM tree with the overlay multicast trees in the previous work since these overlay multicast trees are designed for different applications. Note that the minimum spanning tree is the same as the optimal MOM tree with δ equal to one.

In the simulation, we use the flat graph with the Waxman distribution [錯誤! 找不到參照來源。](#) as the network topology to test our algorithm in networks with different graph characteristics. We also use MBone [錯誤! 找不到參照來源。](#) and the Internet topology with the power-law distribution generated by Inet [錯誤! 找不到參照來源。](#) [錯誤! 找不到參照來源。](#) 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 a larger α and β has a larger node degree and a smaller graph diameter [錯誤! 找不到參照來源。](#). Therefore, the multicast trees in different graphs lead to different characteristics. (2) Group size. The group size is the number of receivers in a multicast tree. (3) The maximum number of addresses that can be included

in each packet, i.e., δ . Since δ is an important parameter in our algorithm, we present our simulation results with different values of δ in most figures to find the proper δ in different scenarios. (4) Distribution of receivers. We use the affinity and disaffinity model [錯誤! 找不到參照來源。](#) 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 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. It represents the consumption of the interface bandwidth of each member. (3) Stress of each link. It is the number of packets sent in each link. (4) Number of addresses in the header of each packet. (5) Depth of a multicast tree. We measure the depth of a tree in terms of the number of members and the number of hops from the sender to each leaf member. The depth of a multicast tree is correlated with the end-to-end delay from the sender to a receiver. (6) Distance from each member to the parent member. It is the number of hops from the parent member to each member.

To find the optimal solutions, we first adopt a small network with 30 nodes in Fig. 6 because solving large ILP problems is computationally infeasible. We use CPLEX [錯誤! 找不到參照來源。](#) with the ILP formulation in [錯誤! 找不到參照來源。](#) and our proposed formulation to find the Steiner tree and the optimal MOM tree, respectively. Fig. 6 compares the total number of packets sent in the Steiner tree (ST), shortest-path tree (SPT), the

optimal overlay multicast tree (MST), the optimal MOM tree (MOM-OPT), and the MOM tree obtained by our algorithm (MOM-LAG) when the group size is 20. Parameters α and β are both set to 2.8 in the Waxman distribution, and the affinity index is zero. Fig. 6 shows that the optimal overlay multicast tree and the MOM tree with δ smaller than two consumes more bandwidth than the shortest-path tree because some links need to send identical packets multiple times. However, the total bandwidth consumption in the optimal MOM tree outperforms, i.e., is smaller than, the shortest-path tree and approaches the Steiner tree as δ increases, and the total bandwidth consumption in the MOM tree obtained by our algorithm approaches the optimal MOM tree. Moreover, Fig. 6 shows that the optimal MOM tree consumes about 20% and 30% less bandwidth than the shortest-path tree and optimal overlay multicast tree, respectively.

Fig. 7 shows the simulation results with different δ and group sizes in a large network with 100 nodes. Parameters α and β are both set to 2 in the Waxman distribution, and the affinity index is zero. Since finding the optimal MOM tree with our ILP formulation in the large network is computationally infeasible, we compare our MOM tree obtained by our algorithm (MOM-LAG) with the lower bound (MOM-LB) on the optimal solution. We find the lower bound by solving LRP defined in Section III. Fig. 7 (a) shows that the MOM tree requires less bandwidth than the optimal overlay multicast 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 in Fig. 7 (b). Fig 7 (b) also shows that MOM can save more bandwidth as the group size increases. Fig. 7 (c) compares the maximum number of packets sent in a link in different multicast trees, where the number in the parenthesis is the group size. Each link in the shortest-path tree, i.e., the IP multicast tree, must deliver exactly one packet. Fig. 7 (c) shows that each link in the MOM tree sends fewer packets than the link in the optimal overlay multicast tree. Besides that, a small δ can effectively reduce the maximum number of packets sent in a link. Fig. 7 (d) compares the average number of addresses in each packet. Each packet in overlay multicast and IP multicast must contain one receiver address. Although MOM requires more addresses in each packet, Fig. 7 (d) 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 , affinity indices, and (α, β) in the Waxman distribution with 100 nodes in the network. The values of (α, β) are set to (0.2, 0.2) and (0.3, 0.3) in the simulations. We vary the affinity index in Figs. 8 (a) and (b). With a larger affinity index, the members tend to cluster together, leading to a small multicast tree. Fig. 8 (a) 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. 8 (a) 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. 8 (b) shows that each member in MOM delivers fewer packets as δ increases. The average number of packets sent by a member may be smaller than one because the members that are the leaf nodes in the multicast tree send no packets. Figs. 8 (c) and (d) compare the depth 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 a 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 multicast tree has a smaller depth. Figs. 8 (c) and (d) show that the depth of an MOM tree approaches that of the shortest-path tree when δ increases. Since the end-to-end delay of a member is correlated with the depth of a multicast tree, we can reduce the end-to-end delay by assigning a suitable δ to the multicast group. Figs. 8 (c) and (d) show that a small δ can effectively reduce the depth of a tree.

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 4177 nodes. In the Internet, there are a few nodes with a large number of degrees ~~錯誤! 找不到參照來源~~, and these nodes tend to act as the backbone routers. Therefore, the graph that

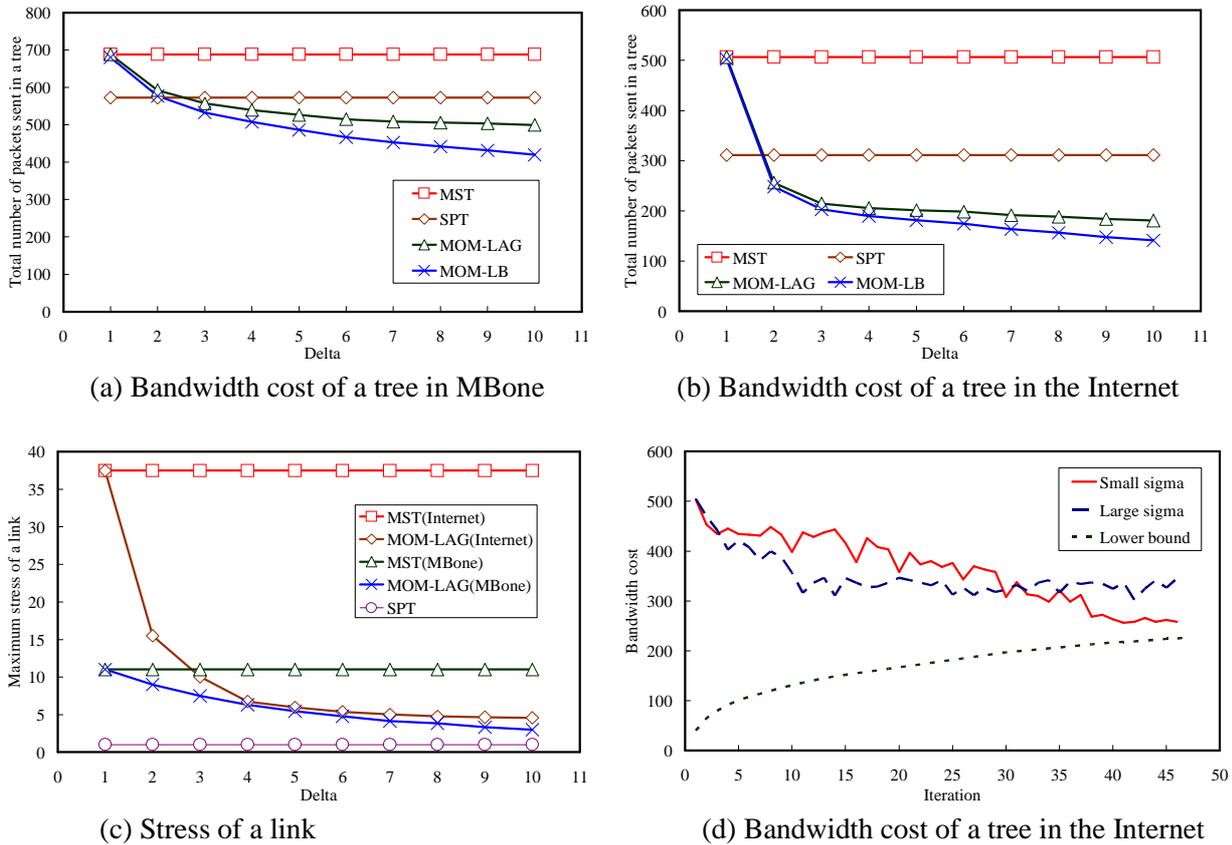


Fig. 9. Comparison of different multicast trees in MBone and the Internet topology generated by Inet, where the group size is 120.

represents the Internet has a smaller diameter as compared to MBone, and we have smaller multicast trees in the graph, as shown in Figs. 9 (a) and (b). 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. Therefore, Fig. 9 (c) 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. 9 (d) shows that the number of iterations in our algorithm is controllable. Note that σ corresponds to 1 or 10 in this simulation. Our algorithm with a large σ converges with about 11 iterations. Note that it is possible that at a certain 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. 9 (d) converges more slowly and obtains a 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 .

V. CONCLUSION

In this paper, we propose a new multicast delivery mechanism, MOM, for bandwidth-demanding applications. Our mechanism use less bandwidth compared with both IP multicast and overlay multicast. The routing of our mechanism is more flexible than IP multicast, and our mechanism avoids the stress problem in overlay multicast. We model our Multiple-destination Overlay Multicast problem as an optimization problem, which is NP-hard. We prove that the optimal solutions to our problem use less bandwidth than the shortest-path trees and the optimal solutions to overlay multicast for all instances. We formulate the problem as an integer linear programming problem. We design an algorithm based on Lagrangean relaxation. We also

conduct simulations for all algorithms. In our simulation results, our mechanism uses less network bandwidth than IP multicast and overlay multicast. Moreover, our mechanism uses less interface bandwidth than overlay multicast because we include multiple addresses in each packet. Therefore, our mechanism is bandwidth-efficient for both network operators and end users.

會議報告

一 參加會議經過

本次出國開會主要是到香港參加 IEEE ICC 2007 的研討會，並發表論文，其題目分別為”Utility-Based Resource Allocation for Layer-Encoded IPTV Multicast in IEEE 802.16 (WiMAX) Wireless Networks。” IEEE ICC 為目前 IEEE 在無線通訊領域十分重要的國際會議之一，此會議將論文分為 8 個 symposia 及 1 general sessions 進行口頭報告及討論，並有 exhibition 同時進行；此會議將論文安排成三天，中間穿插著 panel discussions, business application sessions 及 幾個 keynote speeches。今年的主題主要包括 communications Theory、Communications QoS, Reliability and Performance Modeling、Multimedia Communications and Home Networking、Signal Processing for Communications、Wireless Communications、Ad Hoc and Sensor Networks。此外並參予 IEEE Globecom 2007 executive committee board meeting 討論。

除了專業報告外，會後更與多位對此協定有關及有興趣的人士進行討論。此外，除了自己的 session 之外，也參與其他相關的演講及討論，已確實掌握最新技術發展現況，並激發新的研究想法。由 IEEE ICC 2007 的論文發表來看，很明顯的目前的技術仍是以無線網路技術為發展重心。

二與會心得與建議

由 IEEE ICC 2007 的論文發表來看，很明顯的無線網路技術發展儼然已是目前最重要的趨勢，此外 cognitive ratio and collaboration networking 及 4G 的發展也十分值得注意。希望今後國內的研究及研究計劃也應繼續朝此方向進行，以其與世界一流研究同步。

此外，此次會議中明顯看出，大陸學者積極參予 IEEE ComSoc 相關活動，其企圖心與積極的態度更甚台灣的學者，這點實在值得注意。

三攜回資料名稱與內容

IEEE ICC 2007 之 conference proceedings 的 CD-ROM。

會議報告

一 參加會議經過

本次出國開會主要是到香港參加 IEEE ICC 2007 的研討會，並發表論文，其題目分別為”Utility-Based Resource Allocation for Layer-Encoded IPTV Multicast in IEEE 802.16 (WiMAX) Wireless Networks。” IEEE ICC 為目前 IEEE 在無線通訊領域十分重要的國際會議之一，此會議將論文分為 8 個 symposia 及 1 general sessions 進行口頭報告及討論，並有 exhibition 同時進行；此會議將論文安排成三天，中間穿插著 panel discussions, business application sessions 及 幾個 keynote speeches。今年的主題主要包括 communications Theory、Communications QoS, Reliability and Performance Modeling、Multimedia Communications and Home Networking、Signal Processing for Communications、Wireless Communications、Ad Hoc and Sensor Networks。此外並參予 IEEE Globecom 2007 executive committee board meeting 討論。

除了專業報告外，會後更與多位對此協定有關及有興趣的人士進行討論。此外，除了自己的 session 之外，也參與其他相關的演講及討論，已確實掌握最新技術發展現況，並激發新的研究想法。由 IEEE ICC 2007 的論文發表來看，很明顯的目前的技術仍是以無線網路技術為發展重心。

二與會心得與建議

由 IEEE ICC 2007 的論文發表來看，很明顯的無線網路技術發展儼然已是目前最重要的趨勢，此外 cognitive ratio and collaboration networking 及 4G 的發展也十分值得注意。希望今後國內的研究及研究計劃也應繼續朝此方向進行，以其與世界一流研究同步。

此外，此次會議中明顯看出，大陸學者積極參予 IEEE ComSoc 相關活動，其企圖心與積極的態度更甚台灣的學者，這點實在值得注意。

三攜回資料名稱與內容

IEEE ICC 2007 之 conference proceedings 的 CD-ROM。