Application-Layer Conference Trees for Multimedia Multipoint Conferences Using Megaco/H.248

Wanjiun Liao, *Member, IEEE*, Jen-Chun Chang, and Victor O. K. Li, *Fellow, IEEE*

***Abstract*—In this paper, we propose a new approach to establishing application layer conference trees for multimedia multipoint conferences on the Internet using the Megaco/H.248 protocol, a Voice over IP (VoIP) media gateway control protocol. In existing VoIP protocols (and also legacy telephone networks), a multipoint conference takes place through an MCU, and forms a star topology centered at the MCU. This paper suggests to establishing shared, cost effective conference trees for VoIP conferences. Each tree is rooted at the conference initiator, who initiates the conference, and spans over all the conference members. Tree branches grow or are trimmed dynamically and adaptively, in a way to avoid the growth of a skewed tree. We develop a simplified analytical model and conduct simulations to evaluate the performance of the proposed approach. The results show that our approach enjoys the advantage of lower join latency and better bandwidth efficiency compared to the traditional MCU approach, and is cost effective compared to a near optimal Steiner tree.**

***Index Terms*—Conference tree, MCU, Megaco/H.248, multipoint conference.**

I. Introduction

INTERNET telephony, also known as Voice over IP (VoIP), promises to deliver real time, two-way, synchronous voice traffic over the Internet or corporate Intranets [1], [2]. The basic concept behind IP telephony is simple: segmenting voice into a series of packets and transmitting them across the IP network to be reassembled at the receiving end. Many standardization organizations have been involved in developing VoIP standards. Important VoIP standards include the International Telecommunication Union—Telecommunication Standardization Sector (ITU-T) Recommendation H.323 [3], [4], the Internet Engineering Task Force (IETF) Session Initiation Protocol (SIP) [5], [6], and the IETF Megaco/ ITU-T H.248 [7], [8]. To date, VoIP is considered one of the most important services in the market, and is expected to generate much revenue for the industry. Major players of VoIP include all major router/switches manufactures, network service providers, and telecommunications, computer, and consumer electronics companies.

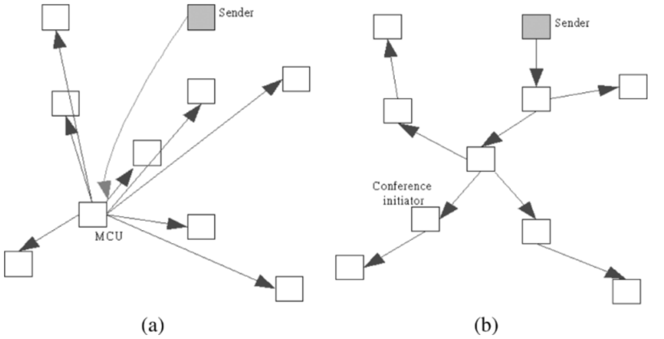


Fig. 1. Two approaches to providing multipoint conferences. (a) Traditional MCU approach. (b) Proposed conference-tree approach.

Multipoint conferences are calls in which three or more conference members are involved. In existing VoIP protocols (and also legacy telephone networks), this service is provided with a multipoint control unit (MCU), which centralizes the control of conference calls. To make a conference call, each member is connected to the MCU via a separate connection, as shown in Fig. 1(a). An MCU takes the responsibility of floor control, media multiplexing, and transcoding for conferences. Data sent to a conference first goes to the MCU, from where the data is mixed and then unicast to other authenticated members. The MCU approach enjoys the advantage of flexible control for conferences, but suffers from the problem of duplicating multiple identical copies of data, each in a point-to-point connection.

In this paper, a new approach is proposed to construct cost effective conference trees for multipoint multimedia conferences over the Internet. With our approach, multipoint conferences take place with a tree topology, rather than a star topology. Each conference tree is rooted at the conference initiator (i.e., who initiates the conference), and spans over all the conference members. Conference data is sent to the tree directly from the branch at which the sender is located, as shown in Fig. 1(b). Thus, it does not transmit duplicate data as in the MCU approach. Our conference tree is established branch-by-branch at the application layer, taking advantage of existing VoIP protocols. Meanwhile, we allow members to freely join or leave a conference, without interrupting on-going conference activities. The contribution of this paper is to propose a feasible approach to enabling multipoint Megaco conferences held via a tree topology. We describe detailed procedures of how a conference is initiated and terminated, and of how users join or leave the conference without disturbing the others. Note that while we describe the tree approach with Megaco/H.248, the approach is applicable to other VoIP protocols.

Manuscript received March 1, 2001; revised November 13, 2003. This work was supported in part by the MOE program for Promoting Academic Excellence of Universities, Taiwan, R.O.C., under Grant 89E-FA06-2-4-7, and in part by the Research Grants Council of Hong Kong, under Grant HKU7044/02E. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chung-Sheng Li.

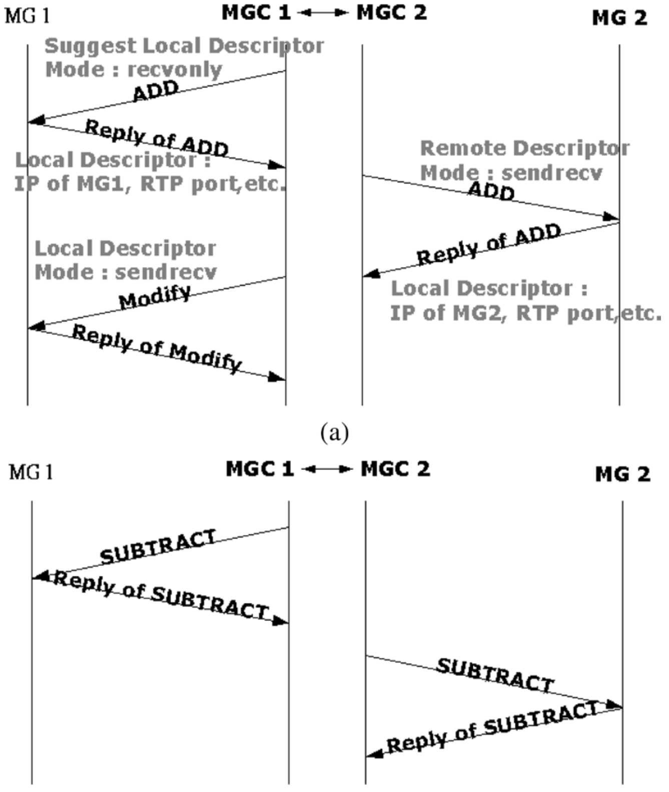
W. Liao and J.-C. Chang are with the Department of Electrical Engineering and the Graduate Institute of Communications Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: wjliao@ntu.edu.tw).

V. O. K. Li is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, China (e-mail: vli@eee.hku.hk).

Digital Object Identifier 10.1109/TMM.2005.854387

The Steiner tree is theoretically optimal as a multicast tree. There are some heuristics such as KMB [9] to construct near optimal Steiner trees as conference trees. However, these heuristics are not practical solutions in real networks, as they perform well only when membership is rather static. When allowing members to dynamically join or leave a conference, these approaches become too expensive because they require the tree to be entirely re-built after each update. To solve this problem, some heuristic algorithms were proposed. The dynamic greedy algorithm [10] is one of the examples. It constructs multicast trees with tree size comparable to near optimal Steiner trees as in KMB. The dynamic greedy algorithm works as follows. To join a tree, a node is connected to the nearest node on the tree using the shortest path. To depart a tree, a node is first marked as a "deleted" node. If the departing node itself is a leaf on the tree, the node is directly removed and its branch originally connected to the tree is pruned. Otherwise, the node is still connected until when all its descendant nodes have left. While being very simple in implementation, this approach might grow skewed trees after a series of departure operations. In [11], the authors proposed an approach that modifies the dynamic greedy algorithm with a new join procedure to reduce the effect of a series of departure. However, if a tree is reshaped in the Join process, new members may suffer from long join latency, an undesirable property for applications like attending conferences. To avoid this problem, we reshape conference trees in the Departing process, and only locally (i.e., not the entire tree). Thus, our conference tree grows dynamically and adaptively to achieve performance better than that in [11].

The rest of the paper is organized as follows. Section II overviews the Megaco/H.248 protocol. Section III describes the proposed approach in details. Section IV evaluates the performance of the proposed approach. Finally the paper is concluded in Section V.



(a)

(b)

Fig. 2. Call Processing in Megaco/H.248. (a) Connection creation. (b) Connection termination.

II. MEGACO/H.248: AN OVERVIEW

Megaco/H.248 is a promising VoIP protocol. It separates the gateway control function from the media transcoding function in media gateways (MG), and places the responsibility of call control outside the MG to the media gateway controller (MGC). Essentially, Megaco is a master-slave protocol in which MGCs instruct MGs on how to perform call processing.

In Megaco/H.248, a new connection model based on the concept of Termination and Context is defined to establish end-to-end voice paths. A Termination is the source or destination of one or more streams, within which media stream parameters and bearer network parameters are encapsulated. A Termination may be established permanently or ephemerally.

A Context is an association with a collection of Terminations. A null Context is a special type of Contexts inside which each Termination is separated, without being associated with one another. Terminations can detect Events (such as off hook) from the connected users, and can apply Signals (such as dial tone) to the users. Each MG may contain many Terminations, each of which may be in different or the same types. An MGC can request to be notified by MGs when certain events are detected by Terminations, and can request MGs to generate certain signals to Terminations. Each Termination has a number of properties, some of which are Termination-specific and some of which are media stream specific. Each property is described as a Descriptor using the modified Session Description Protocol (SDP) [12]. Descriptors are encapsulated in Commands exchanged between MGCs and MGs. Each command may carry a number of Descriptors. An MGC may suggest MGs how to configure Descriptors for connections. The MGs may provide their local connection parameters, such as IP addresses, UDP ports and RTP profiles, in rely, so as to proceed to connection establishment.

Megaco/H.248 has defined eight important gateway control commands, including Audit capabilities, Modify, Subtract, Move, Audit value, Add, Notify, and Service change. The Notify command is to report the detection of a certain event by MGs to their MGCs. The Add, Subtract, and Modify commands are used to add, remove, and modify Terminations, respectively, to Contexts.

Fig. 2(a) summarizes the creation procedure of a point-to-point, two-party call. Suppose that MGC1 and MGC2 are the MGCs of MG1 and MG2, respectively. Upon being notified by MG1 of a connection being created, MGC1 sends an Add command to create a calling Termination at the requesting gateway (MG1). The command carries a Local Descriptor suggestion from MGC1. It also carries an operation mode of "recvonly" from the calling side. Thus the calling Termination can only receive streams from the connected user. MG1 allocates resources to the connection, followed by a Reply sent by the calling Termination to MGC1. The Reply message indicates the modifications to the original Local Descriptor, including the IP address of MG1, the RTP port, and session description using SDP. After the Reply, MGC1, through a routing table lookup, determines the outgoing gateway, say MG2, at which the called party is located. It then notifies the affiliated MGC (i.e., MGC2) of a connection being created. MGC2 repeats what MGC1 does. It sends an Add along with the description of the calling Termination received from MGC1 to create the called Termination at MG2. The called Termination responds with its description to MGC2, again, using SDP. The response is then relayed to MGC1. MGC1 forwards the description of the called Termination to the calling Termination using a Modify command, changing the operation mode at the calling side to be "sendrecv." The calling Termination then sends an immediate Reply to confirm the establishment of a bi-directional connection. Fig. 2(b) summarizes the termination procedure of a point-to-point call. Again, upon being notified of tearing down a connection, the MGCs send a Subtract command to the corresponding MGs, clearing the call and releasing the resources previously allocated to the Terminations.

III. Multipoint Conference Trees for MEGACO/H.248 Conferences

In this section, a tree-based solution for multipoint conferences over the Internet is proposed. Each conference tree is rooted at the root-MG where the conference initiator is located, and spans over MGs connected to all conference members. Conference data from any member is sent directly to the tree in one copy only. Thus, it effectively reduces the usage of system resources as compared to the MCU approach, as the latter sends multiple identical copies of the data, each in a point-to-point connection. From the system perspective, a multipoint conference tree grows or is trimmed when members join or leave the conference, respectively. How a tree grows determines how system resources are used. Members participate in a conference through the use of such operations as "creating," "terminating," "joining," or "leaving" conferences, which in turn determine how the conference tree grows. In this section, we describe these four conference operations in detail. In the next section, we will examine the performance of our approach in terms of join latency, bandwidth usage, and tree size. Note that since Megaco is a hard-state mechanism, all callers dropped by accident need to dial to re-connect to the conference.

A. Conference Creation and Termination

A conference is initiated when a member (i.e., the initiator) starts a Termination in a null Context on its connected MG. The root-MG is the MG where the initiator is located. Each conference is associated with a conference number with which users can identify. The initiator dials the conference number and connects to the root-MG. On receiving a notified number, an MGC performs an authentication checking, and informs the root-MG of a success in conference creation.

A conference is terminated when the initiator leaves the conference. It happens when the initiator goes off hook. The initiator's MG notifies its MGC of the departure via a Notify message if the MG is a Residential Gateway, or via an REL message if the MG is an SS7 gateway. The MGC then forwards the conference termination message to all other MGCs involved in the conference. Each MGC then sends Subtract commands to the corresponding on-tree MGs to tear down connections and release the resources previously allocated.

B. Join a Multipoint Conference

To join a conference, a user dials the corresponding conference number. This makes the connected MG add a Termination to the Context associated with the conference, and notifies its MGC of a connection being created. To make a point-to-point call, there is a single callee and thus the work is to determine the outgoing gateway (denoted as Out-MG) to the callee. Based on the received dialed number, the MGC can determine the outgoing gateway (Out-MG) simply by a routing table lookup. However, to make a multipoint conference call, there may be several callees associated with a single number. Thus, to determine these Out-MGs, the MGC must first check if the notifying MG (denoted as In-MG) is connected to the conference tree. If it is on-tree, the upstream (toward the root) parent node of the In-MG in the tree is selected as the Out-MG. In this case, since both the In-MG and the Out-MG have been on-tree, the Join process is finished. If the In-MG is not on-tree, the In-MG must join the tree. The MGC determines an on-tree MG in the shortest path to the In-MG as a new Out-MG. Once the Out-MG is identified, the MGC can establish a point-to-point connection between the In-MG and the Out-MG. This can be done with the normal process of point-to-point connection creation described in Section II. In other words, the MGC sends an Add command to both the In-MG and the Out-MG, and adds their Terminations to the Contexts associated with the target conference. The MGs respond to the MGC with their session descriptions to do capability negotiation and to enable related association between the Terminations. Therefore, the new identified Out-MG, again, becomes the parent node of the In-MG in the resulting conference tree.

Fig. 3 shows an example of the Join operation in a conference. Assume that MGC-1 controls both MG-11 and MG12; MGC-2 is the MGC of MG-21. It is also assumed that all users participate in the conference through PSTN bearer channel Terminations. User 1 initiates a conference, as shown in Fig. 3(a). In Fig. 3(b), User 2 joins the conference through MG-12, which is not on-tree. MGC-1 instructs MG-11 to add an RTP Termination (i.e, T2) and a PSTN bearer channel Termination (i.e., T1) to Context 1. It also instructs MG-12 to add an RTP Termination (i.e., T3) and a PSTN bearer channel Termination (i.e., T4) to Context 2. MGC-1 associates T2 with T3 to establish a connection between Users 1 and 2. In Fig. 3(c), when User 3 joins the conference, both RTP Terminations T5 and T6, and PSTN bearer channel Termination T7 are added to the conference. Since MG-11 and MG-21 are controlled by two different MGCs, i.e., MGC-1 and MGC-2, these MGCs should exchange information to establish the connection. In Fig. 3(d), User 4 joins. Since the In-MG of User 4 (i.e., MG-21) has been on-tree, only PSTN bearer channel Termination T8 is added to Context 3.

C. Leave a Multipoint Conference

A user may dynamically join or leave a conference. Upon detecting a Termination going on-hook, either via a Residential

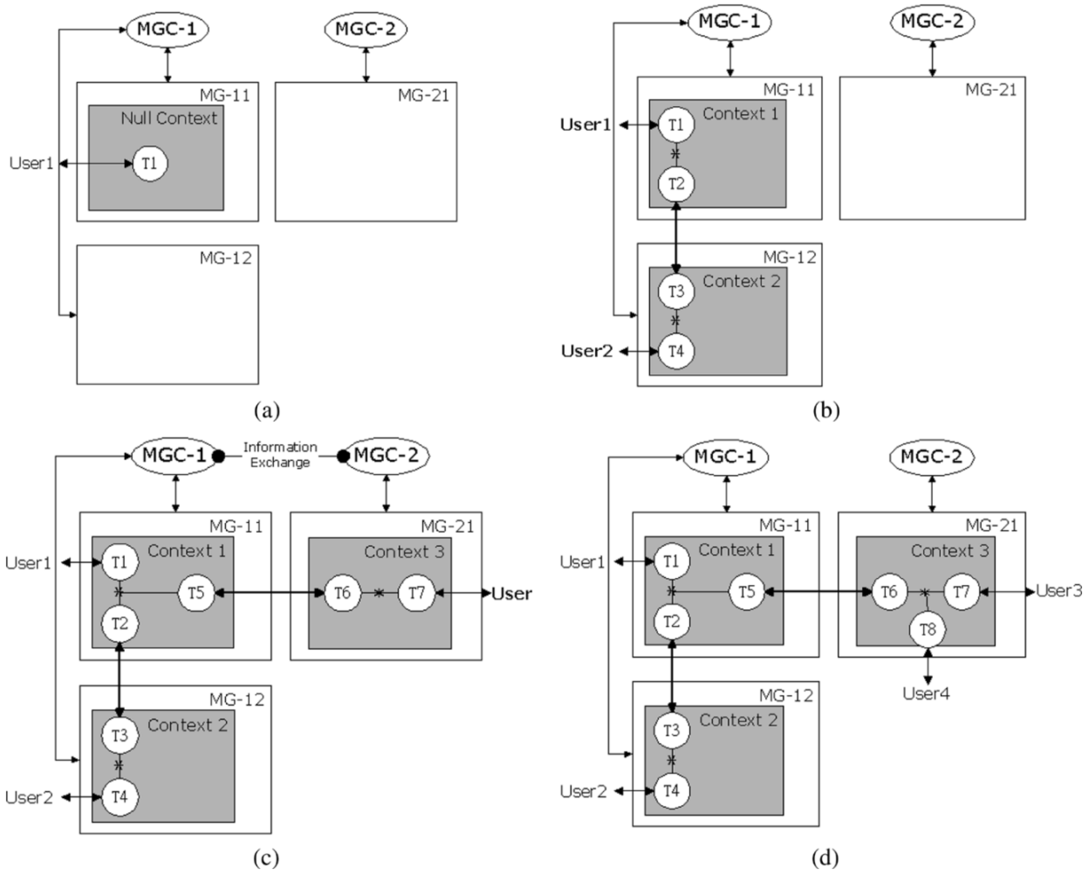


Fig. 3. Join a multipoint conference. (a) The initial state of the Conference. (b) Both Out-MG and In-MG are in the same MGC. (c) Out-MG and In-MG are with different MGCs. (d) The In-MG is on-tree.

Gateway or an SS7 gateway, the Termination's MG notifies its MGC of the departure of the Termination. An MG at which a departing Termination is located is called a departing MG. If there are other Terminations in a departing MG or a departing MG itself is a leaf node in the conference tree, it just uses the normal Megaco's departing procedure via a Subtract command to tear down the connection. If either is not the case, the MGC applies a local reshaping algorithm to avoid the growth of a skewed tree. The basic reshaping idea is explained as follows. To retain on-going conference activities, the connection between the departing node and its parent node can be torn down only after those connections between the departing node and all its immediate child nodes have been glued back to the tree. This works only when the re-route path and the original path coexist in the system because we would like to re-shape the tree smoothly.

The local reshaping algorithm is described as follows.

Step 1: Mark a departing MG with no other Terminations as a "departure" node.

Step 2: Mark all the descendants of the departure node as "re-join" nodes.

Step 3: Define the part of the original tree except the departure node and the re-join nodes as a target tree.

Step 4: Glue a subtree rooted at an immediate child node of the departure node back to the target tree. The immediate child node joins the target tree in the shortest path using the normal Megaco's joining process of point-to-point connections.

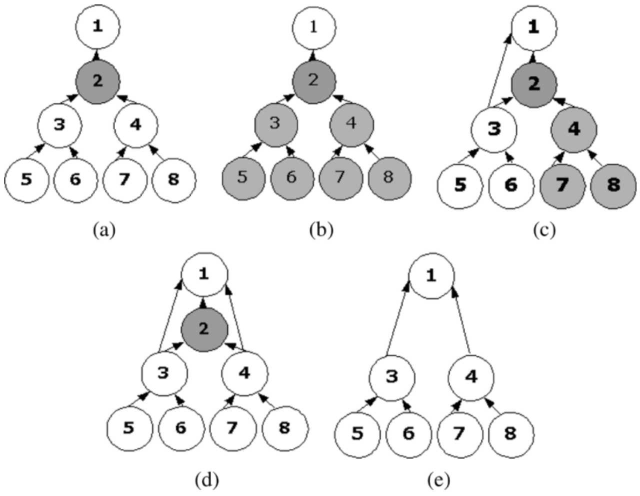
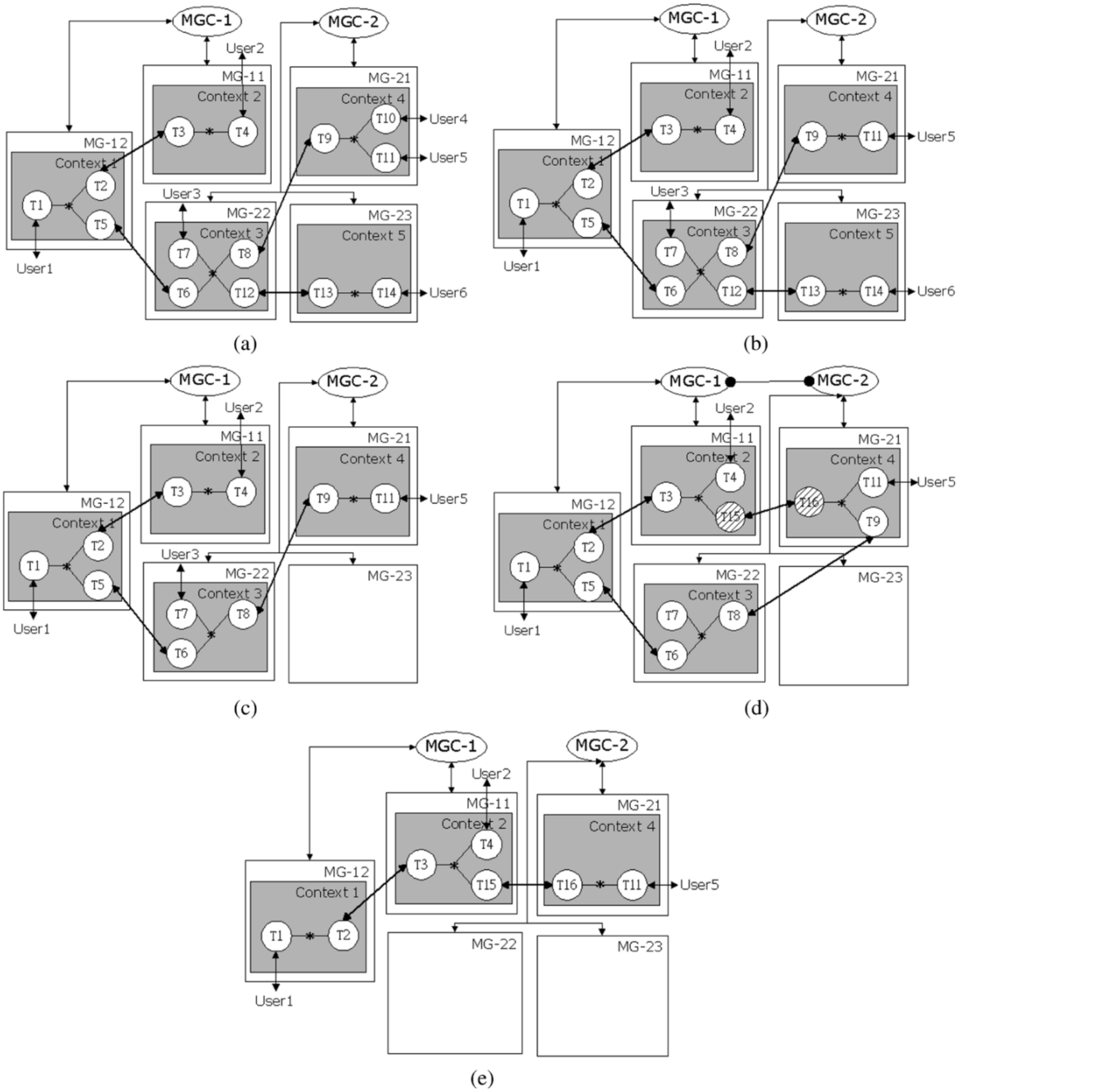


Fig. 4. Local reshaping algorithm.

Step 5: Unmark all the nodes at the subtree which have been successfully glued back to the target tree.

Step 6: Repeat Step 3 to 5 until all the subtrees, each of which is rooted at one of the immediate child nodes of the departure node, have been glued back to the tree.

Step 7: Delete the departure node and all the direct connections to its parent and to all the immediate child nodes.



(a)

(b)

(c)

(d)

(e)

Fig. 5. Leave a multipoint conference. (a) Conference topology. (b) Still other Terminations in a Context when departing. (c) Departure of a leaf MG. (d) Departure of an intermediate MG. (e) Resultant conference tree after a series of departures.

Fig. 4 shows an example to describe how the local reshaping algorithm works. Assume that node 2 is a departing MG. In Fig. 4(a), node 2 is marked as a departure node, and in Fig. 4(b), nodes 3 to 8 are marked as rejoin nodes. In Fig. 4(c), the subtree rooted at node 3 is glued back, and in Fig. 4(d), the subtree rooted at node 4 is glued back. Finally, in Fig. 4(e), node 2 and all its branches are deleted.

Fig. 5 shows how the protocol works when a user leaves a conference. Fig. 5(a) shows a conference topology. In Fig. 5(b), User 4 leaves the conference. Since User 5 is still in Context 4, MG-21 only removes T10 when receiving a Subtract command from MGC-2. In Fig. 5(c), User 6 leaves the conference. Since MG-23 is a leaf MG in the tree, both T13 and T14 (as well as Context 5), and T12 are removed. Then User3 leaves. Once User 3 has left, Context 3 should be removed. However, MG-21, the child MG of MG-22, would be disconnected from the conference tree if Context 3 were removed without doing anything. Thus, MGC-2 selects a new parent MG for MG-21 (i.e., MG-11) according to the local reshaping algorithm. Once MG-11 is determined, a new connection between MG-11 and MG-21 is established by adding T15 and T16 to Contexts 2 and 4, respectively [see Fig. 5(d)]. After the connection has been successfully built, Context 3 is removed and Terminations T5 and T9 are subtracted (see Fig. 5(e)).

IV. Performance Evaluation

This section evaluates the performance of the proposed conference tree approach in terms of join latency, bandwidth efficiency, and tree size. Each performance metric is defined as follows.

1) Join latency is normally defined as the time elapsed between when a user joins a conference and when the user has received data. Here we refine the Join latency of a new

member as the hop count from the member's MG to any on-tree gateway for a conference. The longer the distance to the conference tree, the higher the join latency the user may experience.

2) The bandwidth efficiency is defined as bandwidth usage per user to participate in a conference.
3) The cost of a conference tree is referred to as the tree size in terms of the number of tree branches.

A. Join Latency

Let $\{\mathrm{x_1},\ \mathrm{x_2},\ \ldots,\ \mathrm{x_K},\ \mathrm{x_{K+1}}, \ldots,\ \mathrm{x_N}\}$ be a set of existing media gateways (MG) in the system, where N is the total number of MGs and K is the number of MGs involved in a conference tree. Without loss of generality, we assume that $\forall \mathrm{x_i},\ \mathrm{i} \leq \mathrm{K}$, gateways $\mathrm{x_i}$ are on-tree. Let $\mathrm{d(x_i, x_j)}$ be the shortest distance between MGs $\mathrm{x_i}$ and $\mathrm{x_j}$ in terms of hop counts. Let $\hat{d}(x_i)$ be the shortest distance from MG $\mathrm{x_i}$ to the conference tree. Thus, $\hat{d}(x_i) = \min\limits_{\forall j}\{d(x_i, x_j)\}$, where $\mathrm{x_j}$ is any on-tree MG, and $\hat{d}(x_i) = 0$ if MG $\mathrm{x_i}$ is already on-tree. The farther the distance for a host to join a tree, the longer the join latency the host has. Thus, the join latency T is in proportion to the mean shortest distance to the conference tree, namely

$$T = \frac{\sum\limits_{i=1}^{N} \hat{d}(x_i)}{N} = \frac{\sum\limits_{i=K+1}^{N} \hat{d}(x_i)}{N} > 0.$$

Let $D_{\max}(K) = \max\{\hat{d}(x_i) | \mathrm{i} = \mathrm{K}+1, \mathrm{K}+2, \ldots, \mathrm{N}\}$, and $D_{\min}(K) = \min\{\hat{d}(x_i) | \mathrm{i} = \mathrm{K}+1, \mathrm{K}+2, \ldots, \mathrm{N}\}$. Thus

$$\frac{(N-K)D_{\min}(K)}{N} \leq \frac{\sum\limits_{i=K+1}^{N} \hat{d}(x_i)}{N} \leq \frac{(N-K)D_{\max}(K)}{N}.$$

The larger the value of K, the tighter the bound on the mean shortest distance to the conference tree.

Let $f(K) = ((N-K)D_{\max}(K)/N)$, and $g(K) = ((N-K)D_{\min}(K)/N)$.

$$f(K) - f(K+1) = \frac{(N-K)\left(D_{\max}(K) - D_{\max}(K+1)\right) + D_{\max}(K+1)}{N}.$$

Since no on-tree gateway leaves, distance from every off-tree node to the tree stays the same or decreases. i.e., $D_{\max}(K) \geq D_{\max}(K+1)$. Thus, $f(K)$ is decreasing. It can be derived that g(K) is decreasing in similar way. This concludes that the mean shortest distance, or the join latency, from a joining host to the tree is bounded by the two decreasing curves $f(K)$ and $g(K)$.

To validate these arguments, we conduct the following experiment to observe the join latency of the proposed approach. Fig. 6 shows the two network topologies in the simulation, generated by network topology generator GT-ITM [13] developed by Georgia Tech. The generator adopts the random graph model designed by Waxman [14] to reflect the structure of real internetworks. This model distributes nodes randomly in the plane, and uses an edge probability function given by

$$\beta \exp \frac{-d(u,v)}{Ld}$$

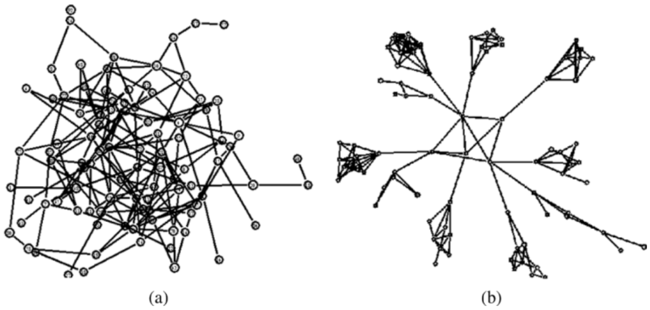


Fig. 6. Network topology in our simulations. (a) Flat graph model. (b) Hierarchical graph model.

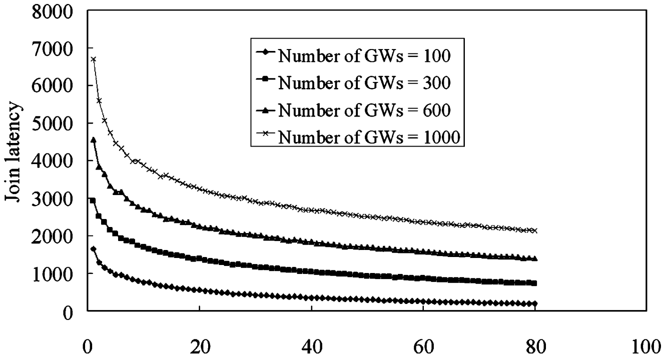


Fig. 7. Join latency.

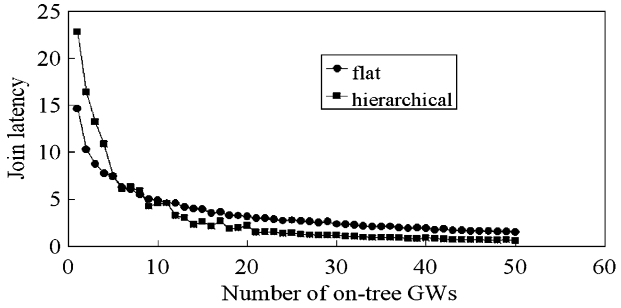


Fig. 8. Join latency versus network topology.

where $1 \geq \alpha, \beta > 0$ and $\mathrm{d}(u, v)$ is the Euclidean distance from nodes u to v. The average node degree of the graph increases as α increases, and the edge densities increases as β increases.

Fig. 7 shows the latency of a new user to join a conference using the flat network topology. We vary the number of on-tree MGs from 5 to 80. The four curves represent four different network sizes with the total number of nodes of 100, 300, 600, and 1000, respectively. Fig. 7 depicts that the join latency decreases with a concave decreasing curve as the number of on-tree MGs increases, agreeing with the analytical result we have done above. The join latency curves have a sharp drop at a point where the number of on-tree MGs is small (around 5), regardless of the network size. Therefore, the proposed approach can significantly reduce the join latency when the number of on-tree MGs is very small even in a large-scale network.

The advantage of which the join latency can be significantly reduced even with a small number of on-tree gateways is magnified in hierarchical networks. Fig. 8 plots the join latencies of the two topologies shown in Fig. 6: flat and hierarchical models.

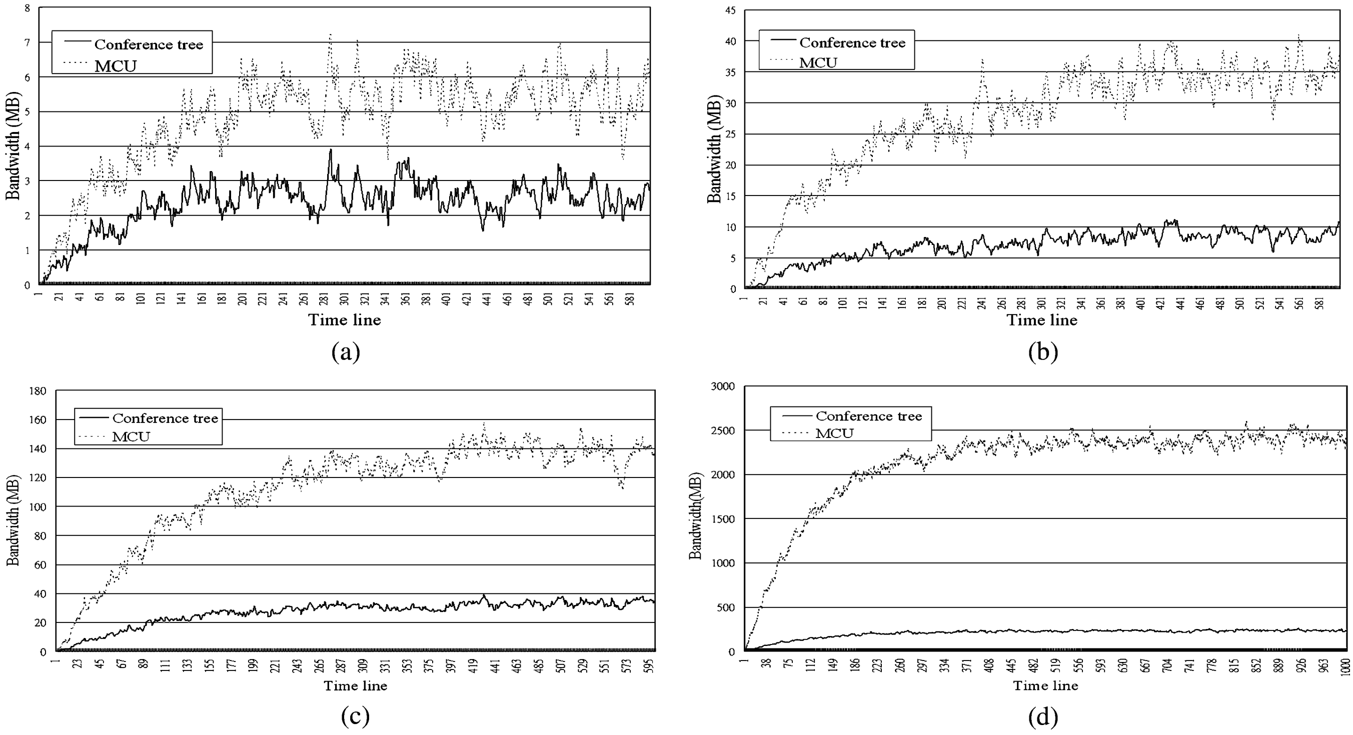


Fig. 9. Bandwidth efficiency. (a) 5 gateways. (b) 10 gateways. (c) 20 gateways. (d) 80 gateways.

It depicts that, as compared to the flat model, the mean join latency of a network organized hierarchically drops more rapidly when a small number of gateways is on-tree.

B. Bandwidth Efficiency

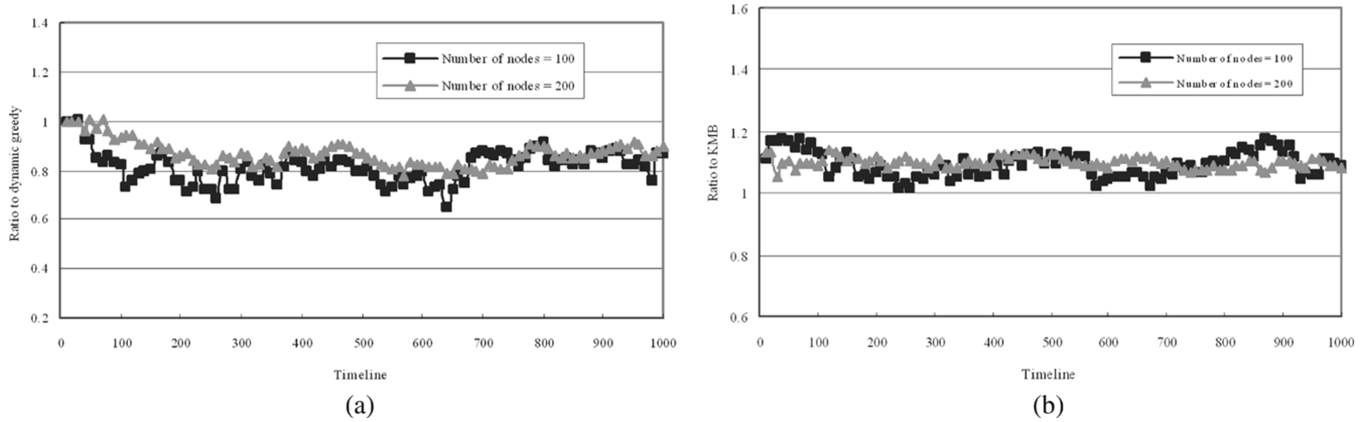
This section investigates the bandwidth usages of our proposed tree approach and the traditional MCU approach. We measure the required bandwidth in terms of the total number of links required per tree in the conference tree approach, or per star in the MCUs. In the traditional MCU approach, each host requires one point-to-point connection to be connected to the MCU. To send data to the conference, a connected MG first transmits the data to the MCU, from where they are duplicated and unicast to all the other MGs involved. Our proposed scheme establishes a shared multicast tree for each conference. The sending MG simply sends data to the tree, without transmitting duplicate data and thus conserving bandwidth.

Suppose there are N MGs involved in a conference, i.e., $\{\mathrm{x_1,\ x_2,\ \ldots,\ x_K,\ x_{K+1},\ldots,\ x_N}\}$. With the MCU approach, let the MCU be one of N MGs, and thus there are N-1 shortest path connections between the MCU and all the other MGs. Let $\mathrm{C_1, C_2, \ldots, C_{N-1}}$ be the cost of these N-1 connections to the MCU in terms of hop count in between. To send data with duration of s time units, MG $\mathrm{x_k}$ sends the data to the MCU first, which consumes a bandwidth of $\mathrm{sC_k}$. Then the MCU uses a $s\sum_{i=1,i\neq k}^{N-1} C_i$ to transmit duplicates to all the others. Therefore, the total bandwidth required is $s\sum_{i=1}^{N-1} C_i$. Our conference-tree approach also requires N-1 connections to connect the N MGs to a shared tree. Let $\mathrm{B_1, B_2, \ldots, B_{N-1}}$ be the cost of these N-1 connections. To join the conference, MG $\mathrm{x_i}$ determines the shortest path to the tree with a cost of Bi. If there is such a path, the cost of the path must be less than or equal to the one connecting $\mathrm{x_i}$ to the MCU in the MCU approach, i.e., $B_i \leq C_i$, $^{\forall}$ $\mathrm{x_i}$. Thus, to send data with s time units, it takes $s\sum_{i=1}^{N-1} B_i \leq s\sum_{i=1}^{N-1} C_i$.

To validate our argument, we conduct a simulation using the same simulation setup as in Fig. 6(a) to compare the bandwidth requirement of these two approaches. The following parameters are used in this experiment.

1) The process that an MG joins the conference is Poisson distributed with a mean inter-arrival time of 1.5 s. The time an MG stays in the conference is exponential distributed with a mean of 106 s.
2) Data sent by each host is distributed in a Poisson process with a mean inter-arrival time of 20 s. The data length is also exponentially distributed with a mean of 320 kB (5 s $*$ 64 kbps).
3) No users are blocked when joining a conference.
4) There are totally 100 nodes in the network, some of which are randomly selected as media gateways for a conference.

Fig. 9 depicts the bandwidth-time relationship between the conference-tree approach and the MCU one. As expected, the conference-tree approach saves much more bandwidth thanks to constructing trees instead of stars. In addition, as the number of MGs increases, the advantage of our proposed approach in bandwidth efficiency becomes more significant. For example, Fig. 9(a) shows that the bandwidth efficiency of our approach is twice better than the MCUs when five out of 100 routers become MGs. In Fig. 9(d), 80 out of 100 routers become MGs, and our approach is even nine times better.



(a) (b)

Fig. 10. Cost efficiency of the conference tree. (a) Ratio to dynamic greedy. (b) Ratio to KMB.

C. Cost Effectiveness

This experiment compares the tree size of the dynamic greedy algorithm and that of KMB to ours. The dynamic greedy algorithm is relatively simple and performs well in comparison to KMB. In this simulation, the network model is generated in the same method in part A of this section but with $\alpha = 0.2$, $\beta = 0.4$. We run a sequence of 1000 events. Each event may be a Join and Departing operation. The probability model which determines if an event is a Join is defined as follows [9], [15]: $P_C(k) = (\gamma(n-k))/(\gamma(n-k) + (1-\gamma)k)$, where n is the number of MGs in the network, k is the number of on-tree MGs connected to the group members, and γ is a real number parameter in the range of (0,1). We set γ to be 0.5 in this simulation. A random number m is chosen in an interval between 0 and 1. If m is smaller than Pc(k), a join event occurs. Otherwise, a departing event takes place. If a Join event occurrs, an off-tree (not on-tree) MG is randomly selected to connect to a group member so as to join the conference. If a Departing event occurs, an on-tree MG connected to a group member is randomly chosen to leave the conference. The measurement metric is defined as the ratio of our conference tree size to the trees constructed by the dynamic greedy algorithm and KMB.

Fig. 10 depicts the two curves of the simulations, each of which represents the one with the numbers of nodes being 100 and 200, respectively. We plot the curves with every ten modifications of events. Fig. 10(a) compares the ratio of our approach to the dynamic greedy algorithm. The initial state k of the simulation is set 1. Thus, the ratio is 1 when the simulation has just started. After a series of events, the system reaches the steady state. The ratio comes to around 0.8 for both curves, meaning that in the steady state, the cost of our trees is only 80% of the cost of the trees constructed by the dynamic greedy algorithm. This is because the dynamic greedy algorithm only concerns with minimizing the increasing cost of join events but not the entire tree. Our local reshaping algorithm, on the other hand, rearranges the tree locally when departing events occur, such that the cost of the tree is lowered. Since the rearrangement occurs only upon the departing events, new join members will not be affected with the reshaping operation. Thus, our approach does not increase the Join latency of new members, the problem suffered by [16]. Fig. 10(b) compares the ratio of our approach to KMB. It shows the ratio is bounded between 1 and 1.2 for the $\mathrm{n} = 100$ curve. For the $\mathrm{n} = 200$ curve, the ratio is about 1.1 in the steady state. Thus, the cost of our tree stays very close to KMB.

V. Conclusion

In this paper, we have proposed a new protocol to construct conference trees for multipoint conferences. Unlike the MCU approach that requires one point-to-point connection per group member connected to the MCU, the proposed conference tree approach maintains a shared, cost effective conference tree for all members in a conference. The tree is rooted at the conference initiator, who initiates the conference. Tree branches are added, point-to-point, in the shortest path from newly joining members to the tree, and are trimmed, point-to-point, from departing members in a way of maintaining the tree more balanced. Data sent to the conference goes to the tree in one copy without duplicates, directly from the branch at which the sender is located. We have developed a simplified analytical model and conducted simulations to evaluate the performance of the proposed approach in terms of join latency and bandwidth efficiency. The results show that our approach enjoys the advantages of lower join latency and better bandwidth efficiency as compared to the traditional MCU approach. In addition, the dynamic multicast tree built with our join and depart operations is cost effective as compared to the tree constructed by dynamic greedy algorithms.

References

[1] "Special Issue on Internet Telephony," *IEEE Network*, vol. 13, no. 3, May/Jun. 1999.

[2] "Special Issue on Internet Telephony," *IEEE Commun. Mag.*, vol. 38, no. 3, Apr. 2000.

[3] ITU-T Rec. H.323v2, Packet Based Multimedia Communications Systems, Mar. 1997.

[4] G. A. Thom, "H.323: The multimedia comm. standard for local area networks," *IEEE Commun. Mag.*, vol. 43, pp. 52–56, Dec. 1996.

[5] M. Handley *et al.*, "SIP: session initiation protocol," in *IETF RFC 2543*, Mar. 1999.

[6] H. Schulzrinne and J. Rosenberg, "The IETF internet telephony architecture and protocols," *IEEE Network*, pp. 18–23, May/Jun. 1999.

[7] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers, "Megaco protocol version 1.0," in *IETF RFC 3015/ITU-T H.248*, Nov. 2000.

[8] T. Taylor, "Megaco/H.248: a new standard for media gateway control," *IEEE Commun. Mag.*, Oct. 2000.

[9] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, vol. 15, pp. 141–145, 1981.

[10] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Sep. 1988.

[11] Y. Ming and X. Xiren, "A optimal dynamic multicast routing algorithm," in *Proc. APCC/OECC '99—4th Optoelectronics and Communications Conf.*, vol. 2, 1999, pp. 1130–1133.

[12] M. Handley and V. Jacobson, "SDP: session description protocol," in *IETF RFC 2327*, Apr. 1998.

[13] K. Calvert and E. Zegura. GT Internetwork Topology Models (GT-ITM). [Online] Available: http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/

[14] M. Doar and I. Leslie, "How bad is naïve multicast routing?," in *Proc. IEEE INFOCOM '93*, San Francisco, CA, 1993, pp. 82–89.

[15] J. Kadirire and G. Knight, "Comparison of dynamic multicast routing algorithms for wide-area packet switched (asynchronous transfer mode) networks," in *Proc. IEEE INFOCOM '95*, 1995, pp. 212–219.

Wanjiun Liao (S'96–M'97) received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, R.O.C., in 1990 and 1992, respectively, and the Ph.D. 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, Taiwan, as an Assistant Professor in 1997. Since August 2005, she has been a Professor. Her research interests include wireless networks, broadband access networks, and Internet protocols.

Dr. Liao is an Associate Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. She has received many research awards. Two papers she co-authored with her students received the Best Student Paper Award at the First IEEE International Conferences on Multimedia and Expo in 2000, and the Best Paper Award at the First International Conference on Communication, Circuits and Systems in 2002. She was elected as one of Ten Distinguished Young Women in Taiwan in 2000, was listed in the *Marquis Who's Who* in 2001–2003, and the *Contemporary Who's Who* in 2005.

Jen-Chun Chang received the B.S. degree in mathematics and the M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1999 and 2001, respectively.

She joined the Benq Co., Taiwan, in 2001. Her research interests include multimedia transmission, wireless internet, and CDMA technology.

Victor O. K. Li (S'80–M'81–SM'86–F'92) received B.S., M.S., E.E., and D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1977, 1979, 1980, and 1981, respectively.

He joined the University of Southern California (USC), Los Angeles, in February 1981, and became a Professor of Electrical Engineering and Director of the USC Communication Sciences Institute. Since September 1997, he has been with the University of Hong Kong, Hong Kong, where he is Chair Professor of Information Engineering in the Department of Electrical and Electronic Engineering. He also served as Managing Director of Versitech Ltd., the technology transfer and commercial arm of USC, from September 1997 to June 2004. In addition, he serves on various corporate boards. His research interests include broadband networks, wireless networks, and Internet technologies and applications. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world. He serves on various government committees, including the Hong Kong Information Infrastructure Advisory Committee, the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Programme Committee, and the Engineering Panel of the Research Grants Council. He was a Distinguished Lecturer at the University of California at San Diego, at the National Science Council of Taiwan, and at the California Polytechnic Institute.

Dr. Li Chaired the Computer Communications Technical Committee of the IEEE Communications Society (1987–1989), and the Los Angeles Chapter of the IEEE Information Theory Group (1983–1985). He co-founded the International Conference on Computer Communications and Networks (IC3N), and Chaired its Steering Committee (1992–1997). He also Chaired various international workshops and conferences, including, most recently, IEEE INFOCOM 2004 and IEEE HPSR 2005. He has served as an Editor of *IEEE Network*, *IEEE JSAC Wireless Communications Series*, and *Telecommunication Systems*. He was also a Guest Editor for special issues of the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATION, *Computer Networks*, and *ISDN Systems*, and the KICS/IEEE *Journal of Communications and Networking*. He is now serving as an Editor for the ACM/Kluwer *Wireless Networks and IEEE Communication Surveys and Tutorials*. He has also delivered keynote speeches at many international conferences. He has received numerous awards, including, most recently, the United Kingdom Royal Academy of Engineering Senior Visiting Fellowship, the Outstanding Researcher Award of the University of Hong Kong, the KC Wong Education Foundation Lectureship, the Croucher Foundation Senior Research Fellowship, and the Bronze Bauhinia Star, Government of the Hong Kong Special Administrative Region, China. He is a Fellow of the HKIE and the IAE.