

Multiple Multicast Tree Allocation in IP Network

Chae Y. Lee and Hee K. Cho

Department of Industrial Engineering, KAIST,

373-1, Kusung Dong, Taejon, 305-701, Korea

Abstract

The multicasting is defined as the distribution of the same information stream from one to many nodes concurrently. There has been an intensive research effort to design protocols and construct multicast routing graphs for a single multicast group. In this paper, the multiple multicast tree allocation problem is discussed and algorithms are proposed to solve the congestion problem in the IP network. As the congestion measure the minimum residual capacity is considered. Two phase algorithm MMTA is investigated for multiple multicast tree allocation both for identical and different bandwidth requirement by the multicast groups. The central and distributed implementation of the multiple multicast tree is discussed for the deployment in the real IP network. The performance of the proposed MMTA is compared with other procedures. Computational results show that the two-phase MMTA outperforms other procedures. Approximately 3-7% improvement in the residual capacity is obtained by the MMTA. The solution gap from the upper bound by the well-known branch and bound is within 2-11% depending on the problem size.

I. Introduction

The multicasting is defined as the distribution of the same information stream from one to many nodes concurrently. In the last few years, multicast routing has attracted a great attention from network community due to emerging applications such as teleconferencing, remote education and collaborative applications [1].

Future networks will carry multiple multicast communications with different QoS requirements that will lead to a competition for the network resources. Therefore, bottlenecks need to be avoided to support as many applications as possible. There has been an intensive research effort to design protocols and to construct multicast routing graphs such as DVMRP [5], PIM-DM [6], MOSPF [7], CBT [8], and PIM-SM [9]. However, most of the effort is concerned to a single multicast group. In this paper, we are interested in multiple and concurrent multicast groups.

Many researchers have examined deploying IP multicast. Boivie et al. [11] propose Small Group Multicast (SGM) to deploy IP multicast in current IP network. In the SGM IP packet header is modified to support multicast. The SGM eliminates the need for multicast routing protocol and employs standards unicast IP routing for multicasting. However, the SGM requires extra bytes in IP header and is not suitable for huge broadcast-like multicast. The Multicast Source Discovery Protocol (MSDP) operates over a TCP connection between Rendezvous Points (RPs) in different domains in which PIM-SM is implemented for multicast [12]. The MSDP advertises active per-group sources and forwards packets across domain boundaries. The MSDP solves the problem of supporting large distribution trees across domain boundaries. However, it requires significant configuration and coordination efforts between ISPs. In addition, all routers in a distribution tree must control, process, and store multicast routing information. These requirements cause scalability problems and increase administrative complexity if there

are a number of multicast groups. Park et al. [4], propose a realistic transport scheme for Internet multicast applications which can easily be deployed. Their multicast transport scheme is based on the unicast transport from a remote sender to a local subnet and the multicast forwarding to receivers within the subnet. Since the scheme is not dependent on the inter-domain multicast, it can easily be deployed in the IP network.

In this paper, we describe a multicast network architecture based on [4] and discuss multiple multicast allocation problem. More detailed network architecture is assumed in the next section.

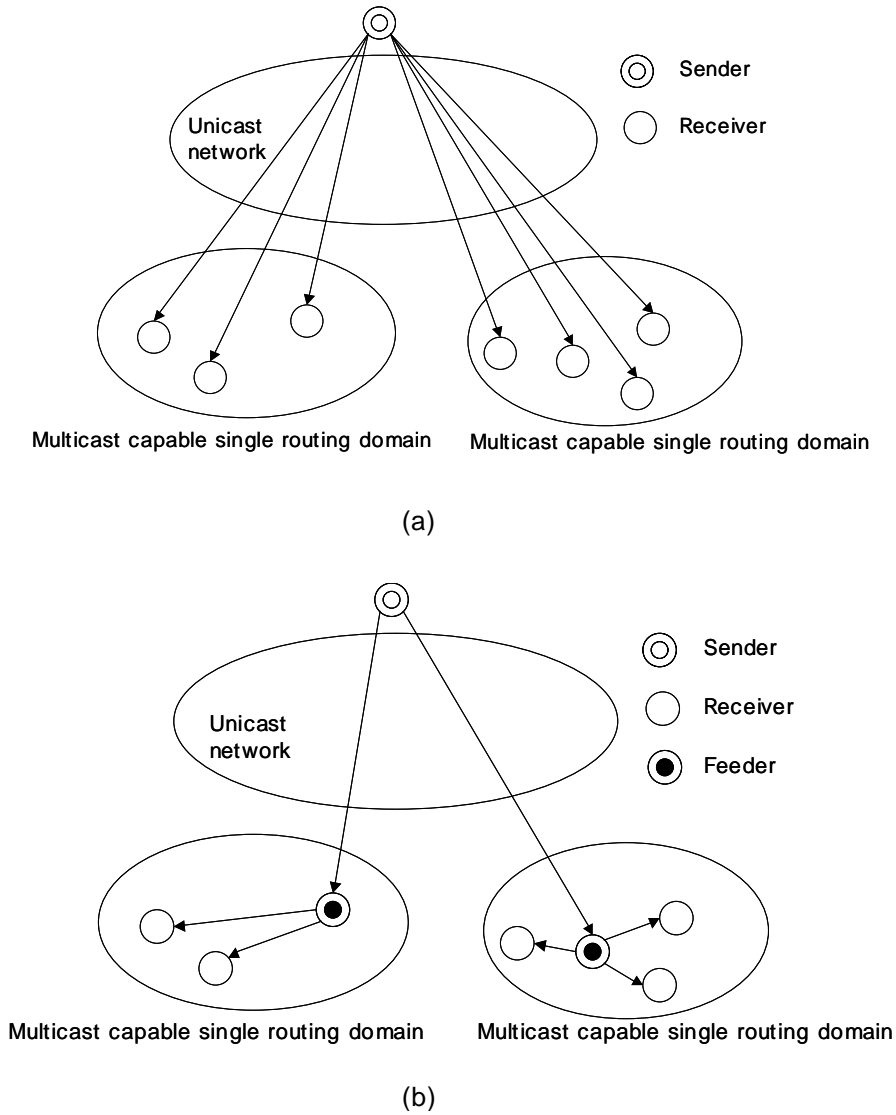


Figure 1. Unicast only and hybrid multicast scheme. (a) Unicast only. (b) Hybrid

Previous works about multiple multicasts are quite limited. A formulation and an elegant solution for accommodating two multicast streams (audio and video) on a small size network (8-nodes) is presented by Noronha and Tobagi [2]. Chen et al. [3] provide a heuristic approach and a lower bound of optimal solution for multiple multicasts that minimizes the network congestion without the capacity limit of each link in the network. Since the heuristic solves Steiner tree problem at each iteration, it requires very high complexity. Wang et al. [14] also consider the problem of multiple multicast and propose heuristic algorithms. The procedures find a set of multicast trees that minimizes overall link cost of the trees. All these multiple multicast allocation problems assume that the required bandwidth for each multicast group is identical.

In this paper by assuming a proper multicast network architecture we propose an efficient heuristic with low complexity for multiple multicast tree allocation. The problem is to allocate several concurrent multicast traffics to the network such that bottlenecks are avoided in the network.

This paper is organized as follows. Section II concerns the network environment that we consider in this paper. In section III, we explain the multiple multicast tree allocation problem and its formulation. Efficient heuristic algorithms for multiple multicast tree allocation are proposed both for the identical and different bandwidth requirements. The implementation issue of the proposed algorithm is discussed in section IV. The computational results are demonstrated in section V and the conclusion in section VI.

II. Network Environment for IP Multicasting

Although multicast is more efficient than the replicated one-to-one unicast transports, IP multicasting has not been widely deployed in the global Internet. One of the main reasons is that

many ISPs still have concerns for implosion of multicast traffic into the network. In particular, a large amount of investment on the existing network is required for the multicast deployment including multicast-capable routers and softwares. In fact, IP multicasting does not seem to be widely deployed in the near future. By this reason, it is hard to manage multiple multicasts in the global Internet.

In the unicast-only networks, the application sender has to send a data stream to each of the receivers by multiple unicast connections as shown in Figure 1 (a). It is well known that these replicated transmissions induce inefficiency in terms of the network resource utilization and management overhead at the sender [10].

In the hybrid multicast proposed by Park et al [4], only a single IP host of each single domain receives the application data stream from the remote sender by unicast. Such a receiver is called the feeder for the concerned application data stream. The other receivers in each single domain receive the application data stream from the feeder as illustrated in Figure 1 (b). The single domain in this paper represents a domain that is managed by a common multicasting protocol.

Such a multicast architecture is possible where the size of a single domain is relatively small and the domain manager has the topology information such as link capacity. It enables the domain manager to allocate the multiple multicasts such that the network resources are utilized efficiently.

In this paper, we focus our attention on the multiple multicast tree allocation within a single domain that typically contains a few hundred routers and possibly several thousand hosts. Figure 2 shows detailed network architecture of a single domain. In the figure, there are two multicast sessions each with its own feeder and receivers. The feeder forwards the application data stream from the remote sender to the other receivers located in the same single domain. A single domain has a domain manager who has the topology information of the network. A

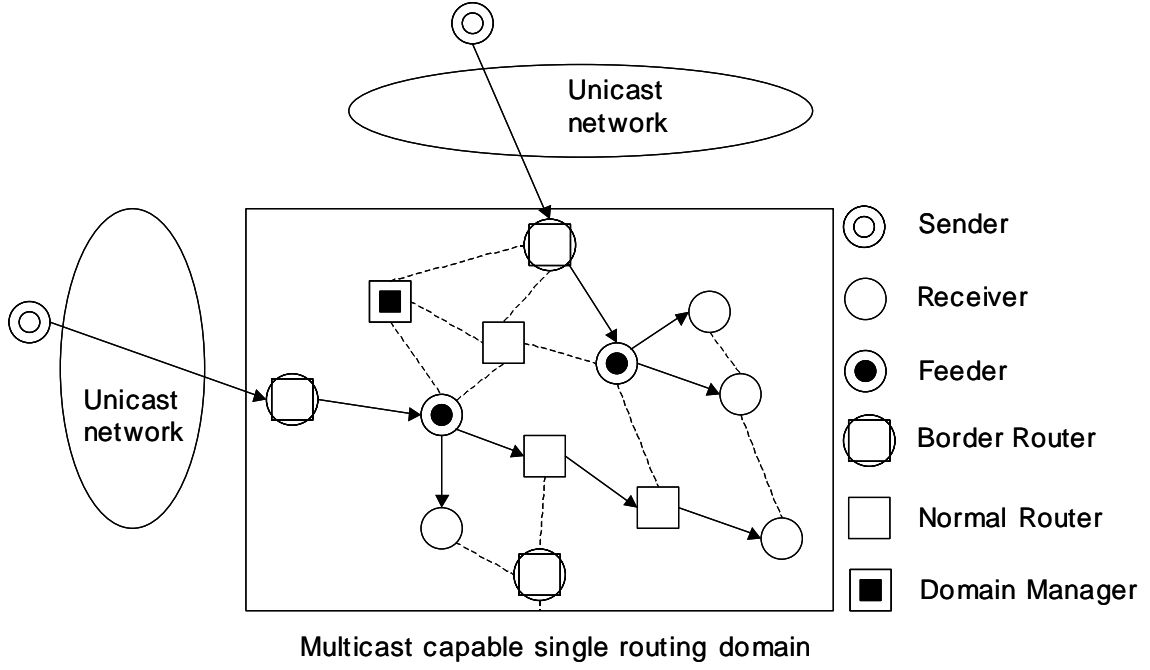


Figure 2. The multicast network architecture

domain manager implements feeder configuration and multiple multicast tree allocation. Note in the figure that proper placement of feeders and multiple multicast tree allocation are critical to the efficient utilization of network resources.

III. Multiple Multicast Tree Allocation

The multiple multicast tree allocation problem is discussed with integer formulation. The Chen's algorithm [3] is modified to consider the link capacity in the network. More efficient multiple multicast tree allocation algorithm MMTA is proposed both for identical and different bandwidth requirement.

1. The Multicast Tree Allocation Problem

Multicasting is an efficient scheme for transmitting packets from a sender to many receivers. A multicast protocol finds a multicast tree through which multicast packets are delivered. If there exist multiple multicast groups in a network, a tree for each group is required to deliver the

corresponding multicast traffic. In this case, we need to design the multiple multicast trees such that the network resources are utilized efficiently. To avoid bottleneck of a network is one important objective of the multicast tree allocation problem. Each multicast tree has to satisfy the network bandwidth and delay threshold. In this paper, the delay threshold is assumed to be satisfied by limiting the size of the multicast tree. By limiting the size of the multicast tree the distance between a sender and a receiver may be restricted. More detailed explanation for the notation and formulation of the problem is followed.

We denote the network by $G = (V, E)$, where V is the set of nodes and E is the set of links. Each link $e \in E$, has a capacity $C_e > 0$. A multicast group k is represented by a set of nodes $N_k \subset V$. A border router connected to a remote sender is also contained in N_k . In the network, we are interested in finding a subgraph G' that spans the multicast group N_k and satisfies a certain subgraph selection criterion. Chen et al [3], select the subgraph that minimizes the network congestion. The traffic load of the most congested link is minimized in the network. However, this criterion is not practical to real network where each link has different capacity. In this paper, as the subgraph selection criterion we consider the maximization of the minimum residual capacity. This criterion is more practical as the congestion measure in a network with different link capacities. High residual capacity which is extra capacity of the network allows better chance of other best effort unicast traffic transport.

Since our objective of the multiple multicast tree allocation is to maximize the minimum residual capacity as opposed to minimizing individual multicast tree costs, the solution to the multiple multicast tree allocation may result in high-cost multicast trees. To restrict the size of each multicast tree and to limit the delays of the real time internet traffic we consider the least size of the multicast tree OPT^k for multicast group $k \in K$, where K is the set of multicast groups. The least size can be obtained by the Steiner tree [3] that includes the nodes in a multicast group.

By restricting the size of the reconstructed tree within αOPT^k ($\alpha \geq 1$) in multiple multicast tree, we can limit the traffic delay in the network.

In the formulation we use binary variables x_e^k for all $e \in E$ and $k \in K$. If link e is used for the tree of the multicast group k , then $x_e^k = 1$. We also define binary variables v_i^k for all $i \in V$ and $k \in K$. If node i is included in the tree of the multicast group k , then $v_i^k = 1$. The traffic load of multicast group k is denoted by t^k which is assumed discrete in traffic unit. Residual capacity of link e is denoted by $z_e = C_e - \sum_{k \in K} t^k x_e^k$ and the minimum residual capacity is denoted by z .

All members of a multicast group must be connected to a tree for delivering multicast packet from a sender. Eq. (1) shows that all nodes in N_k are connected. For any proper subset S of V , we denote the collection of links with one endpoint in S and the other in $V \setminus S$ by $d(S)$.

$$\begin{aligned} \sum_{e \in d(S)} x_e^k &\geq 1 && \text{for all } k \in K \\ &&& \text{and for all } S \subset V \text{ such that } S \cap N_k \neq N_k \end{aligned} \quad (1)$$

To guarantee a tree for each multicast group k , Eq. (2) – (4) are required. Let $E(i)$ be the set of links that are connect to node i . Then the node i in the multicast tree k has to satisfy Eq. (2) and (3) with $v_i^k = 1$. Thus Eq. (4) guarantees a tree for each multicast group k .

$$v_i^k \geq x_e^k \quad \text{for all } i \in V, \text{ all } k \in K \text{ and all } e \in E(i) \quad (2)$$

$$v_i^k \leq \sum_{e \in E(i)} x_e^k \quad \text{for all } i \in V \text{ and all } k \in K \quad (3)$$

$$\sum_{i \in V} v_i^k = 1 + \sum_{e \in E} x_e^k \quad \text{for all } k \in K \quad (4)$$

Also the multicast traffic has to satisfy the network bandwidth. In other words, multicast traffics that pass through a link has to satisfy the link capacity. Considering the minimum residual capacity z of the network, each link has to satisfy the following constraint.

$$C_e - \sum_{k \in K} t^k x_e^k \geq z \quad \text{for all } e \in E \quad (5)$$

Now, in the process of distributing the congested traffic a multicast tree may experience delay due to the extended tree. Thus, we need to limit the number of links in a tree such that the size

of the tree does not exceed the minimum Steiner tree by a factor of α . In Eq. (5) the OPT^k represents the number of links in the Steiner tree for nodes in multicast group k .

$$\sum_{e \in E} x_e^k \leq \alpha \text{OPT}^k \quad \text{for all } k \in K \quad (6)$$

Note that objective of the problem is to maximize the minimum residual capacity. Thus the formulation for the multiple multicast tree allocation is given as follows:

$$\begin{aligned} \text{Max} \quad & z \\ \text{s.t.} \quad & \sum_{e \in d(S)} x_e^k \geq 1 && \text{for all } k \in K \\ & && \text{and for all } S \subset V \text{ such that } S \cap N_k \neq N_k \\ & v_i^k \geq x_e^k && \text{for all } i \in V, \text{ all } k \in K \text{ and all } e \in E(i) \\ & v_i^k \leq \sum_{e \in E(i)} x_e^k && \text{for all } i \in V \text{ and all } k \in K \\ & \sum_{i \in V} v_i^k = 1 + \sum_{e \in E} x_e^k && \text{for all } k \in K \\ & C_e - \sum_{k \in K} t^k x_e^k \geq z && \text{for all } e \in E \\ & \sum_{e \in E} x_e^k \leq \alpha \text{OPT}^k && \text{for all } k \in K \\ & z \geq 0 \\ & v_i^k, x_e^k \in \{0, 1\} && \text{for all } e \in E, i \in V \text{ and all } k \in K \end{aligned}$$

Solving the multiple multicast tree allocation is significantly more difficult than to solve one multicast tree design problem which is known as NP-hard [3]. This is due to the max-min nature of the objective function in the multiple multicast tree problem. We propose heuristics for the multiple multicast tree allocation problem first by assuming each multicast group has the same traffic unit, i.e., $t^k = 1$. Allocation of multicast trees with different bandwidth is also considered to improve the residual capacity.

2. Improvement of Chen's Algorithm

Basically, in multiple multicast tree allocation problem, each multicast tree is constructed

Preprocessing:

1. For each multicast group k , construct a multicast tree T_k independently.
2. For each link e , compute residual capacity z_e .

Input: set of multicast trees computed independently $T = \{T_k: k = 1, \dots, |K|\}$ and a bound on the tree size αOPT^*

Output: Revised multicast trees

BEGIN

1. Choose a link e with the minimum residual capacity z and a group k that traverses the link. A random selection is applied to break ties in selecting the link e .
2. Construct an auxiliary graph, G' which is obtained from the union of the links whose residual capacity is at least $z+2$.
3. Solve the Steiner tree problem on G' for the group k .
4. IF the size of tree found at step 3 is within the range αOPT^* for the group k ,
 THEN update z_e values and goto step 1.
 ELSE
 IF another pair of link and group exists with the minimum residual capacity z ,
 THEN choose the link and the group and goto step 2.
 ELSE
 THEN stop.

Figure 3. Modified Chen's Algorithm

independently for initial solution set. We thus focus on reconfiguring multiple multicast trees from the initial solution to maximize the minimum residual network capacity.

Note that the heuristic algorithm by Chen et al. [3] presents a solution to multicast tree allocation problem without the capacity constraint of each link. To take the link capacity into account the residual capacity is employed. The algorithm at each iteration chooses a link e with the minimum residual capacity and finds a tree T_k which employs the link. The algorithm removes the link with the minimum residual capacity from a multicast tree T_k and reconstructs a tree using a Steiner tree algorithm. If the size of the newly constructed tree T_k' is within the range of αOPT^k , then T_k is replaced with T_k' . T_k' becomes a new multicast tree for the multicast

group k and the algorithm repeats the procedure. If such a tree T_k' is not found, the tree construction is repeated with another pair of link and group whose minimum residual capacity is in tie with the link e and group k . If no such pairs exist, the algorithm is terminated.

In the procedure, the Steiner tree problem is solved at each iteration by employing the well-known KMB algorithm by Kou et al. [13]. The time complexity of KMB is $O(nv^2)$, where n is the number of member for a multicast group and v is the number of routers in the network. Since each multicast group has one tree, there are k trees and for each tree we may have to process each link which makes total $k|E|$ and for each link and each tree combination we call procedure rebuild with complexity $O(nv^2)$. Then, overall complexity of modified Chen's algorithm is $O(k|E|nv^2)$.

3. Allocation of Multiple Multicast Tree with Identical Bandwidth

The modified Chen's algorithm is relatively complex compared to the KMB because the Steiner tree needs to be solved every iteration. Thus we propose a more efficient multiple multicast tree allocation algorithm MMTA that can be applicable to real network. In this section, we propose the MMTA by assuming each multicast group has the same traffic unit, i.e., $t^k = 1$.

In the MMTA, a sorted list of links is maintained to order the links according to their residual capacity as in the algorithm of Section 2. The proposed MMTA has two phases. In the first phase, the most congested link with minimum residual capacity z is selected and the link is removed from the corresponding multicast tree T_k . Removing the most congested link partitions the multicast tree into two disconnected parts. To connect the two disconnected parts an alternate path need to be found. Note that to increase the residual capacity of the network each link in the alternate path has at least $z+2$ residual capacity, which leads to improved $z+1$ residual capacity after adopting the traffic of the multicast group k . Here, the addition of new links in the

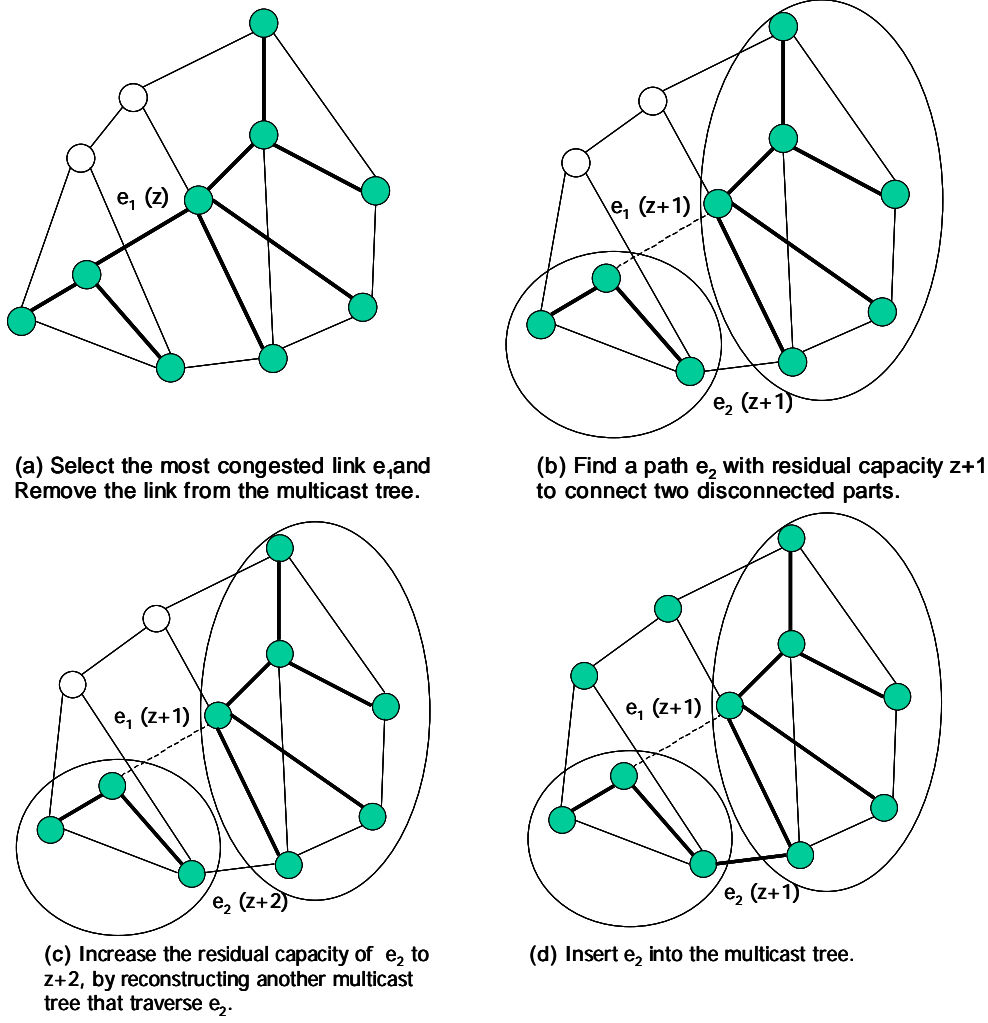


Figure 4. An example of the second phase in MMTA

alternate path has to satisfy the limit of tree size αOPT^k .

The second phase is implemented when no improved path is found to connect the two disconnected parts of T_k in the first phase. In the second phase, an alternate path with $z+1$ residual value is selected to connect the two disconnected parts within the limit of tree size. After finding the path with $z+1$ residual capacity, a multicast tree T_k' is selected that traverses link l in the path with $z+1$ residual capacity. The multicast tree T_k' is reconstructed after deleting the link l by using the same procedure as in phase 1. Since T_k' is reconstructed without the link l , the residual capacity of the path including the link l is improved from $z+1$ to $z+2$. Finally the

two disconnected parts of T_k is connected with the alternate path and the residual network capacity is improved to $z+1$.

Figure 4 explains the second phase of the proposed MMTA algorithm. In phase 1, the most congested link e_1 is deleted from the tree (Figure 4-a) and e_2 is selected as a path to connect the disconnected parts (Figure 4-b). In phase 2, another multicast tree that traverses link e_2 is reconstructed (Figure 4-c) and the multicast tree is connected via path e_2 with improved residual

```

Preprocessing:
1. For each multicast group  $k$ , construct a multicast tree  $T_k$  independently.
2. For each link  $e$ , compute residual capacity  $z_e$ .
Input: set of multicast trees computed independently  $T = \{T_k: k = 1, \dots, |K|\}$  and
a bound on the tree size  $\alpha OPT^k$ 
Output: Revised multicast trees

BEGIN
1. Choose a link  $e$  with the minimum residual capacity  $z$  and a group  $k$  that
traverses the link. A random selection is applied to break ties in selecting the
link  $e$ .
- Phase 1
2. Remove the link  $e$  from tree  $T_k$  and partition tree into two parts.
3. Find a shortest path to connect two parts with at least  $z+2$  residual capacity.
4. IF there exists such a path within the range  $\alpha OPT^k$  for the group  $k$ 
    THEN insert the path into  $T_k$  and update  $z_e$  values and goto step 1.
    ELSE
        IF another pair of link and group exists with the minimum residual
        capacity  $z$ ,
            THEN choose the link and the group and goto step 2.
        ELSE
            THEN goto step 5.
- Phase 2
5. Find a path to connect two parts with  $z+1$  residual capacity.
6. IF it is possible to increase the residual capacity of the path,
    THEN increase the residual capacity of the path and insert the path into
     $T_k$  and update  $z_e$  values and goto step 1.
    ELSE
        THEN stop.

```

Figure 5. The procedure of MMTA

network capacity.

If a path with $z+1$ residual value is not found in the second phase, the algorithm is terminated. Figure 5 shows the procedure of the proposed algorithm. In the algorithm, the most congested link with minimum residual capacity is selected. Then a multicast group k that traverses the link is selected to be reconstructed as shown in step 1 of the main process. The selection of the multicast group k is based on the small-group-first, which has the smallest number of group members. The small-group-first has an advantage in delay aspect compared to the other selection criterion.

In the proposed multiple multicast tree allocation algorithm, we find a path to connect two disconnected parts at each iteration. This is transformed to finding the shortest path between two partitioned parts. The time complexity of finding the shortest path is $O(v^2)$, where v is the number of routers in the network. Since we call tree rebuild procedure $k|E|$ times in the worst case, the overall complexity of proposed algorithm is $O(k|E|/v^2)$.

4. Allocation of Multiple Multicast Tree with Different Bandwidth

Allocation of multicast groups with different bandwidth is more complex than the case with all identical bandwidth. As in the case of identical bandwidth, a multicast group k needs to be selected among groups that traverse the most congested link. Clearly, selecting the multicast group with the highest bandwidth (maximum t^k) leads to the highest residual capacity of that link. However, this highest-bandwidth-first method has a disadvantage of decreasing the residual value of other links that connects the two disconnected parts of the multicast tree.

An example is shown in Figure 6. Each number in the link represents the link capacity and the number in the parentheses represents the residual capacity. In Figure 6 (a), there are two multicast groups. Group 1 consists of node 1 and 5, requires 4 units of bandwidth, and has 1-4-5

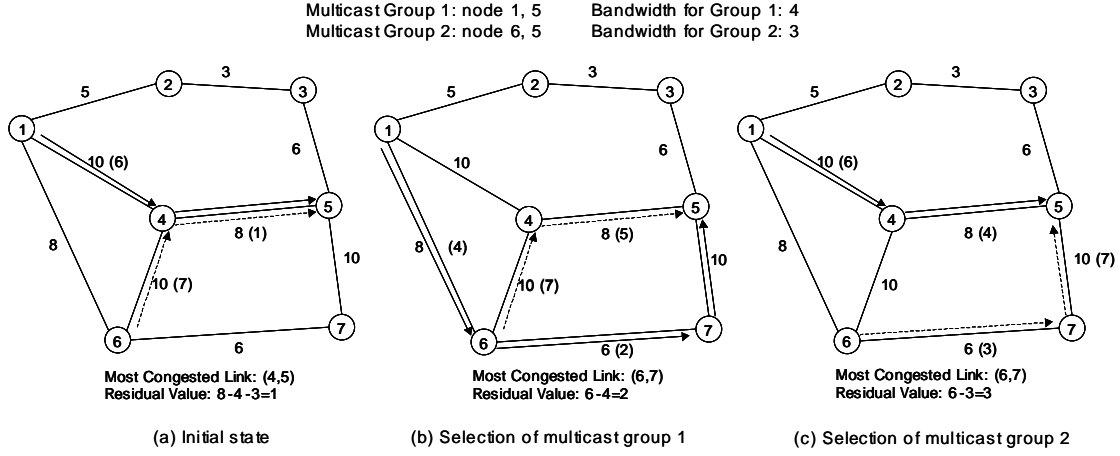


Figure 6. The highest-bandwidth-first selection scheme

multicast tree. Group 2 consists of node 5 and 6, requires 3 units of bandwidth, and has 6-4-5 multicast tree. The most congested link is link (4, 5) and the residual value is one ($=8-4-3$). Figure 6 (b) shows that group 1 is selected at link (4, 5) by the highest-bandwidth-first rule. By routing group 1 through the path of 1-6-7 the most congested link (6, 7) has residual value two. However, the solution can be improved by selecting group 2 instead of group 1 in the most congested link (4, 5). The result of selecting group 2 is shown in Figure 6 (c). This example shows that the highest-bandwidth-first scheme does not always provide the best solution.

To effectively increase the residual capacity of the network, we consider selecting a multicast group that maximizes the *alternative gain* (k) that is the difference of the residual value of the network before and after reconstructing a multicast group k . Let z_1 and z_2 be respectively the residual value of the most and second most congested link. Depending on the required bandwidth of the selected multicast group the improvement of the residual capacity results from one unit up to z_2-z_1 . Even if the improvement in the most congested link exceeds z_2-z_1 , the residual capacity in the overall network is limited by the second most congested link.

Figure 7 shows the alternative gain with regard to the required bandwidth of the selected multicast group. The solid line in the figure represents the best case where the *alternative gain*

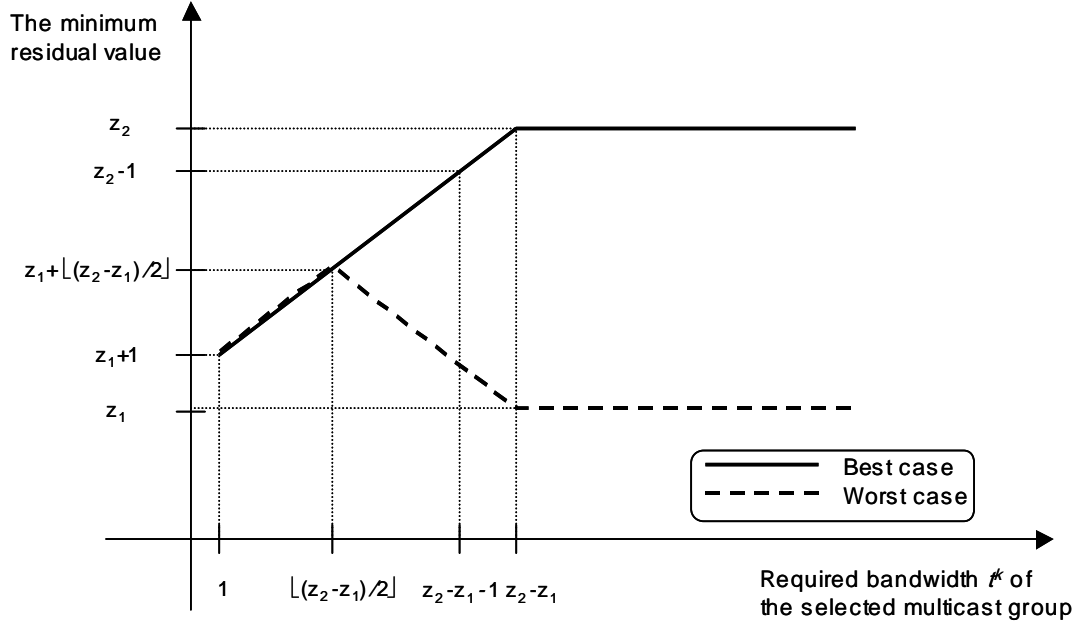


Figure 7. The minimum residual capacity vs. the required bandwidth of the selected

(k) is maximized. Clearly, the best case occurs when the second most congested link is not included in the alternate path. When the alternate path of the reconstructed multicast tree includes the second most congested link, the improvement results in the dashed line of Figure 7. If the bandwidth t^k of the selected multicast group k is smaller than $\lfloor (z_2 - z_1)/2 \rfloor$, the resultant minimum residual value is identical to the best case. However, if $\lfloor (z_2 - z_1)/2 \rfloor < t^k < z_2 - z_1$, the resultant minimum residual becomes $z_2 - t^k$ since $z_2 - t^k < z_1 + t^k$ in case of $t^k > \lfloor (z_2 - z_1)/2 \rfloor$. Thus the resultant minimum residual value in the worst case is limited by the most congested link, when $t^k < \lfloor (z_2 - z_1)/2 \rfloor$ and by the second most congested link otherwise.

Clearly, the best selection of a multicast group that maximizes the *alternative gain* may be obtained by evaluating all the candidates that traverses the most congested link. However, the procedure may seriously increase the computational effort as the network size increases. As a simple and efficient procedure we propose a closest-gap-first procedure that selects a multicast group whose bandwidth is closest to $z_2 - z_1$. The two multicast group selection procedures:

highest-bandwidth-first and closest-gap-first will be compared in the computational simulation of Section V.

IV. Implementation of the Multiple Multicast Allocation

The multiple multicast tree allocation algorithms presented in section III can be implemented at a domain manager that has the network topology information. A simple signaling procedure can be employed by the domain manager to periodically update list of multicast groups, multicast receivers in each group and the required bandwidth. Each router sends a control packet to the domain manager whenever a multicast allocation is required. The control packet contains the information of the multicast session at the router and the required bandwidth at the links connected to the router. Based on the periodically updated information of the residual capacity of each link, multicast trees are constructed such that the residual network capacity is maximized by the proposed procedure in Section III-3.

However, the implementation of the centralized control by the domain manager may not be practical. As an alternative we propose a distributed implementation of the multiple multicast tree allocation. In the distributed implementation, a router is assumed to have information about the multicast sessions passing through itself. The distributed implementation of the multiple multicast allocation is operated by control messages proposed in this section. The procedure of the distributed implementation is as follows.

When the network residual capacity is below a predefined threshold value, a border router sends `ALLOCATION_INITIATE` messages to all receivers of the multicast sessions passing through the router. When a multicast receiver receives the `ALLOCATION_INITIATE` message, it sends `ALLOCATION_REQUEST` message to the border router. The `ALLOCATION_REQUEST` message contains the information of the receiver's address, required bandwidth, and

the most congested link in the path from the border router to the receiver. Thus, the border router identifies the most congested link when all `ALLOCATION_REQUEST` messages are received. Now to construct an alternate path that replaces the most congested link, the border router selects one of the multicast groups that pass the most congested link. The border router sends `REJOIN` messages to the multicast receivers connected through the congested link. When a multicast receiver receives the `REJOIN` message, it searches a new path to the border router. The new path has to satisfy the limit of tree size. If such a path is found, it rejoins the multicast session through the new path and sends an `ALLOCATION_CONTINUE` message to the border router. Otherwise, the multicast receiver sends an `ALLOCATION_TERMINATE` message to the border router. If the border router receives the `ALLOCATION_CONTINUE` messages from all the receivers to which the border router sends `REJOIN` messages, it updates the most congested link and sends `REJOIN` messages again to the selected multicast receivers sharing the most congested link. This rejoin procedure is repeated to update the residual network capacity. If the border router receives the `ALLOCATION_TERMINATE` message, it selects another multicast group that passes the most congested link and repeats the rejoin procedure. If the border router receives `ALLOCATION_TERMINATE` messages from all multicast groups to share the most congested link, this means that the minimum residual capacity cannot be increased. Thus the procedure is terminated and the multicast trees are determined.

V. Computational Results

The performance of the proposed algorithm MMTA is analyzed with experimental networks. Three types of networks are considered each with $|V| = 50, 100, \text{ and } 200$ nodes and $|E| = 2|V|$. In each type ten networks are generated. The capacity of each link has a uniform distribution over

Number of multicast groups	Algorithms	Number of members in each multicast group				
		4	6	8	10	12
0.1 V	Modified Chen's	18.1 (8.0)	17.6 (12.0)	17.2 (16.3)	16.7 (20.1)	16.6 (23.7)
	MMTA	18.1 (1.7)	17.7 (1.8)	17.3 (1.8)	17.1 (2.0)	16.8 (2.2)
	CPLEX	18.1 (735.3)	18.0 (2784.3)	17.7 (5210.2)	17.6 [*] (10000)	17.3 [*] (10000)
0.15 V	Modified Chen's	16.3 (12.1)	15.7 (19.0)	14.9 (25.5)	14.1 (32.3)	13.6 (38.1)
	MMTA	16.4 (2.5)	15.8 (2.7)	15.1 (2.6)	14.3 (2.8)	14.0 (2.9)
	CPLEX	16.7 (4265.3)	16.2 [*] (10000)	15.5 [*] (10000)	14.9 [*] (10000)	14.8 [*] (10000)
0.2 V	Modified Chen's	14.0 (23.0)	13.2 (31.3)	12.3 (32.3)	11.6 (40.1)	10.8 (47.7)
	MMTA	14.1 (2.7)	13.4 (2.8)	12.6 (2.9)	11.7 (3.0)	11.2 (3.1)
	CPLEX	14.5 [*] (10000)	13.9 [*] (10000)	13.1 [*] (10000)	12.3 [*] (10000)	11.8 [*] (10000)

* represents the upper bound of the solution

The number in the parenthesis represents the CPU seconds

Table 1. Computational results with $|V|=50$

Number of multicast groups	Algorithms	Number of members in each multicast group				
		4	6	8	10	12
0.1 V	Modified Chen's	16.7 (58.1)	16.6 (150.2)	16.0 (204.3)	15.7 (256.4)	15.5 (305.5)
	MMTA	16.9 (15.1)	16.8 (15.3)	16.3 (15.9)	16.0 (16.2)	15.8 (16.8)
	CPLEX	18.1 [*] (10000)	18.0 [*] (10000)	17.4 [*] (10000)	17.3 [*] (10000)	17.1 [*] (10000)
0.15 V	Modified Chen's	15.4 (732.0)	14.6 (230.4)	13.7 (305.4)	13.2 (381.5)	12.6 (454.6)
	MMTA	15.5 (46.3)	14.9 (46.5)	14.1 (47.1)	13.4 (47.4)	13.1 (48.1)
	CPLEX	16.7 [*] (10000)	16.0 [*] (10000)	15.3 [*] (10000)	14.7 [*] (10000)	14.3 [*] (10000)
0.2 V	Modified Chen's	13.1 (205.3)	12.3 (305.4)	11.3 (406.6)	10.6 (506.7)	9.9 (606.9)
	MMTA	13.2 (61.8)	12.4 (62.1)	11.5 (62.7)	10.8 (63.4)	10.3 (64.1)
	CPLEX	14.4 [*] (10000)	13.6 [*] (10000)	12.8 [*] (10000)	12.0 [*] (10000)	11.4 [*] (10000)

* represents the upper bound of the solution

The number in the parenthesis represents the CPU seconds

Table 2. Computational results with |V|=100

Number of multicast groups	Algorithms	Number of members in each multicast group				
		4	6	8	10	12
0.1 V	Modified Chen's	16.0 (823.3)	15.5 (1123.4)	15.1 (1632.6)	14.6 (1843.6)	14.3 (2133.7)
	MMTA	16.1 (142.8)	15.8 (158.6)	15.3 (182.3)	15.1 (197.5)	14.7 (235.6)
	CPLEX	17.3 [*] (10000)	16.8 [*] (10000)	16.7 [*] (10000)	16.2 [*] (10000)	15.9 [*] (10000)
0.15 V	Modified Chen's	14.3 (1532.4)	13.7 (1732.1)	12.8 (2231.8)	12.2 (2531.4)	11.7 (3129.2)
	MMTA	14.5 (235.3)	13.8 (262.2)	13.0 (283.7)	12.4 (297.4)	12.0 (321.9)
	CPLEX	15.6 [*] (10000)	15.1 [*] (10000)	14.2 [*] (10000)	13.2 [*] (10000)	13.1 [*] (10000)
0.2 V	Modified Chen's	12.0 (1421.5)	11.1 (2142.8)	10.2 (2843.0)	9.4 (3215.3)	8.6 (4386.6)
	MMTA	12.2 (281.1)	11.4 (315.9)	10.6 (321.1)	9.9 (336.5)	9.2 (391.1)
	CPLEX	13.3 [*] (10000)	12.6 [*] (10000)	11.6 [*] (10000)	10.8 [*] (10000)	10.4 [*] (10000)

* represents the upper bound of the solution

The number in the parenthesis represents the CPU seconds

Table 3. Computational results with |V|=200

the integers, ranging from 18 to 22 for identical bandwidth with $t^k=1$. For different bandwidth case the bandwidth t^k is assumed to vary from 1 to 5 and the link capacity is distributed over [55, 65]. The number of multicast groups is given proportionally to the size of the network such that $0.1|V|$, $0.15|V|$, and $0.2|V|$. Average number of group members varies from 4 to 12. To restrict the size of each multicast tree, the restriction parameter α is given by $\alpha=2$. In the procedure of the MMTA, the Steiner tree solution OPT^k is approximated with the well-known KMB algorithm [13]. All solution procedures are run on a Pentium II-660MHz PC.

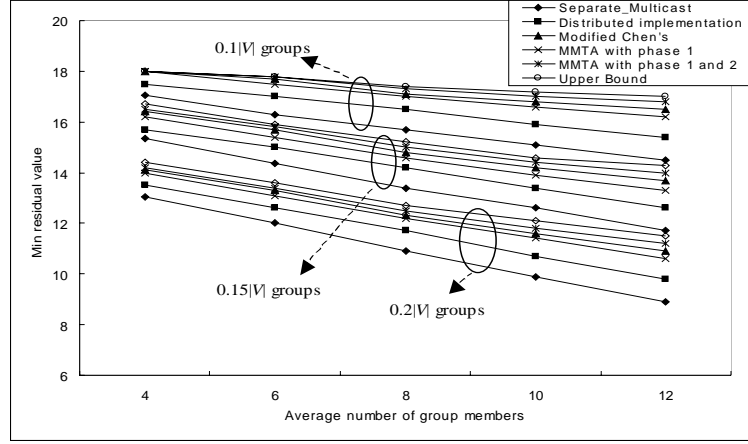
The performance of the proposed algorithm MMTA is tested with identical bandwidth ($t^k = 1$) for each multicast group. Table 1, 2 and 3 show the objective function values and computational times by the Modified Chen's algorithm, MMTA and the well-known branch and bound procedure by CPLEX [15]. Except some cases with $|V| = 50$, the CPLEX failed to provide optimal solutions within the CPU time limit of 10,000 seconds. Thus upper bounds are presented for the cases. Clearly, the MMTA presents 3-7% better residual capacity than the Modified Chen's procedure. Moreover, the computational time by the proposed MMTA is reduced by an order compared to the Modified Chen's algorithm. The solution gap between the MMTA and the upper bounds by the CPLEX is within 5% in problems with 50 nodes, 6-10% with 100 nodes, and 6-11% in 200 nodes.

Figure 8 shows the residual capacity by various procedures including the three algorithms in Tables 1-3. Figure 8 (a), (b), and (c) are the result of 50, 100, and 200 nodes respectively. In the figure, the proposed MMTA algorithm using only phase 1 falls behind the Modified Chen's algorithm. However, the MMTA with two phases has better minimum residual capacity than the Modified Chen's procedure. This shows that phase 2 of the proposed algorithm significantly improves the solutions by finding an alternate path more efficiently. The figure also shows that the distributed implementation has better minimum residual value than that of the Separate

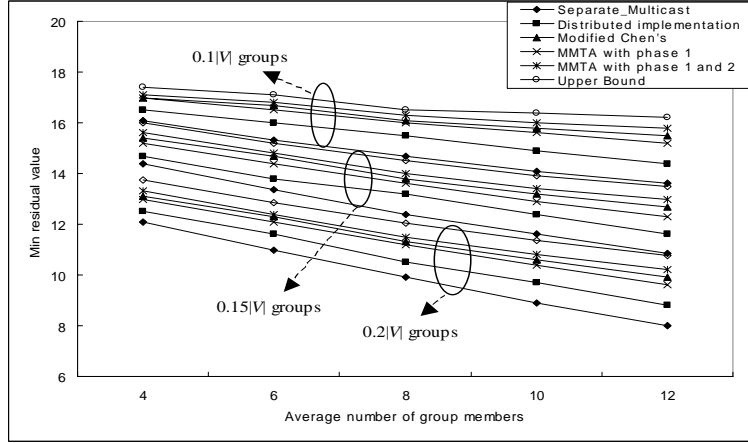
Multicast, where each multicast tree is constructed independently.

Figure 9 shows the delay performance of centralized and distributed implementation. A network with $|V|=100$ and ten multicast groups is experimented. Delay is measured by average hop counts from a sender to a receiver. In the proposed MMTA the minimum residual capacity is increased by increasing the multicast tree size, which results in increased delay by receivers in the multicast tree. In the distributed implementation, however, finding an alternate path that increases the residual capacity is not so efficient compared to the centralized implementation. In other words, the tree size in the distributed implementation is relatively small compared to the centralized procedure. As a result, the distributed implementation has less delay than the centralized method. Approximately 12% improvement is obtained in the packet delay by the distributed process.

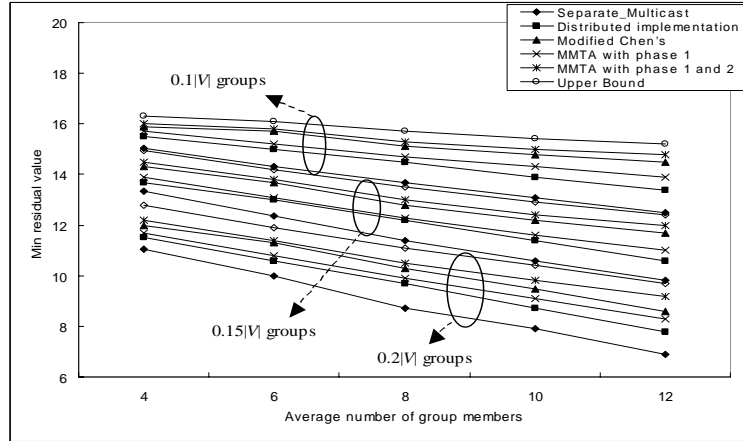
We now investigate the performance of the MMTA in problems with different bandwidth in each multicast group. The highest-bandwidth-first and closest-gap-first procedures are compared as the selection scheme of the multicast group that traverses the most congested link. The bandwidth t^k is assumed to vary from 1 to 5. Figure 10 (a), (b), and (c) demonstrates the results of 50, 100, and 200 nodes respectively. In the figure, the closest-gap-first has higher minimum residual value than the highest-bandwidth-first. The figure also shows that the solution gap between the closest-gap-first and the highest-bandwidth-first increases as the number of group members increases. Clearly, the closest-gap-first is more efficient than highest-bandwidth-first in problems with large number of multicast receivers.



(a) $|V|=50$



(b) $|V|=100$



(c) $|V|=200$

Figure 8. Selection of multicast trees with identical bandwidth

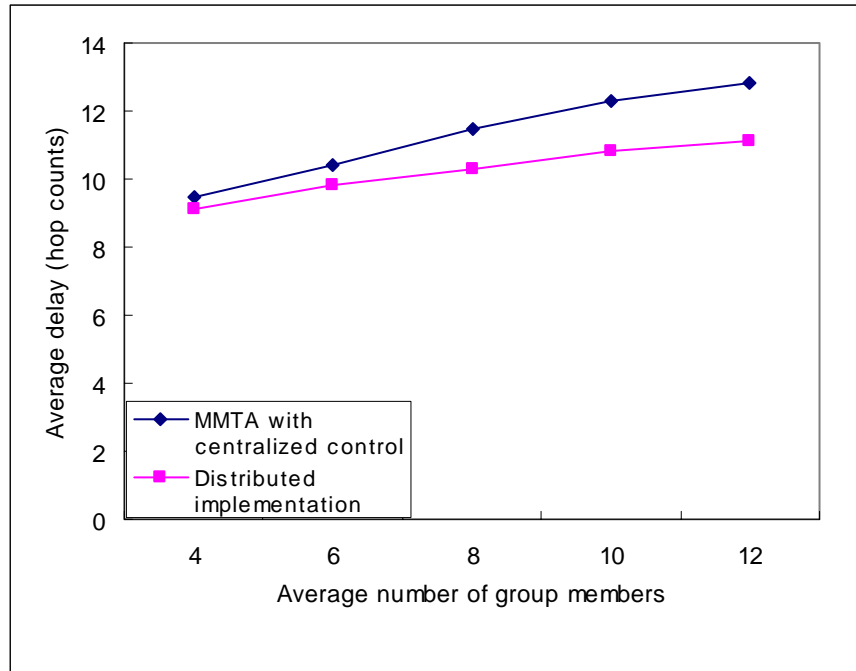
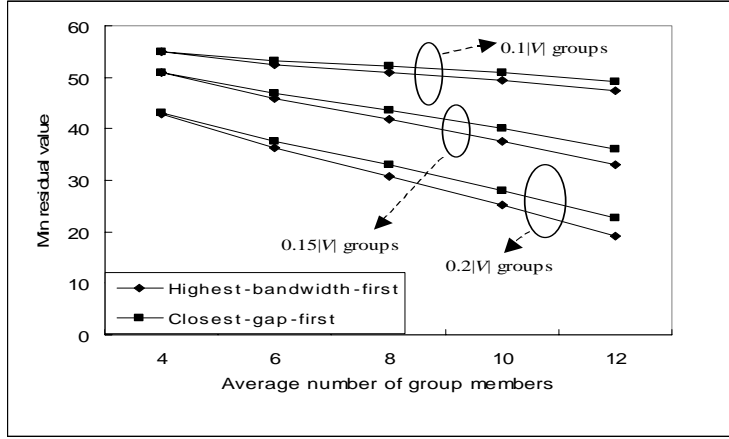
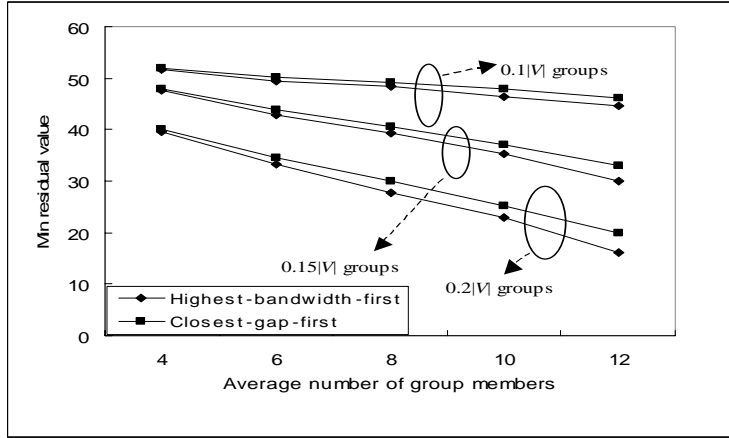


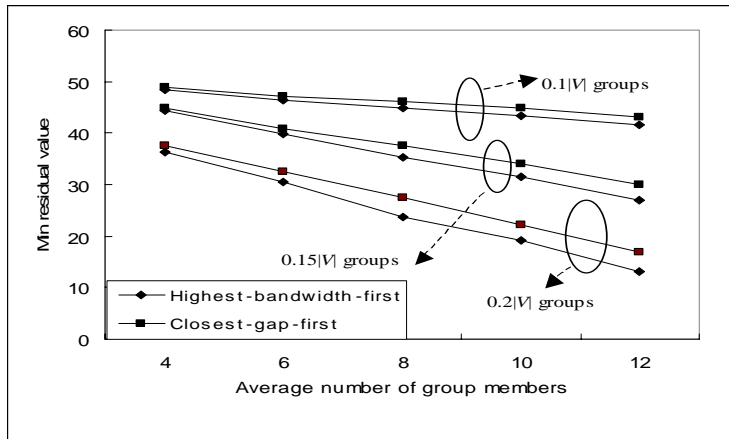
Figure 9. Delay performance of the MMTA: centralized vs. distributed implementation



(a) $|V|=50$



(b) $|V|=100$



(c) $|V|=200$

Figure 10. Selection of multicast trees with different bandwidth

V. Conclusion

In this paper, the multiple multicast tree allocation problem is examined. Linear integer formulation is presented to allocate multiple multicast trees that maximize the minimum residual capacity in the network. A two-phase MMTA is provided to solve the multiple multicast tree allocation.

The first phase replaces the link with the minimum residual capacity in a multicast tree with an alternate path that has higher extra capacity. When such an alternative path cannot be found, the second phase is applied. In the second phase the residual capacity is increased by swapping the most congested link with an alternate path in which a multicast tree is swapped again with other path. The two-phase MMTA is also applied to problems with different bandwidth requirement. To select a multicast group that traverses the most congested link the alternate gain is considered. The multicast group that maximizes the difference of the residual capacity before and after the reconstruction is considered by the closest-gap-first selection criterion. A distributed implementation of the multiple multicast tree is proposed and compared with the centralized implementation.

The performance of the proposed MMTA is analyzed with computational results. The two-phase MMTA demonstrates 3-7% higher residual capacity than the modified Chen's algorithm. The solution gap between the MMTA and the upper bound by the well-known CPLEX is also examined. The gap is within 5% in problems with 50 nodes and 6-11% in problems with 100 and 200 nodes. In problems with different bandwidth the closest-gap-first selection shows better performance than the other selection criterion. The distributed implementation is proved to reduce the packet delay compared to the centralized control of the multicast allocation.

Reference

- [1] Yi J, Pastor E. Specification of Communication Services for Multimedia Cooperative Applications Based on the Reference Model of ODP. ETRI Journal 1998; 20 (2): 149-170.
- [2] Noronha C, Tobagi F. Optimum routing of multicast streams. IEEE INFOCOM, 1994. p. 865-873.
- [3] Chen S, Gunulk O, Yener B. The Multicast Packing Problem. IEEE/ACM Transactions on Networking. 2000; 8 (3): 311-318.
- [4] Park J, Koh SJ, Kang SG, Kim DY. Multicast Delivery Based on Unicast and Subnet Multicast. IEEE communications letter 2001; 5 (4): 181-183.
- [5] Waitzman D, Deering S, Partridge C. Distance Vector Multicast Routing Protocol. RFC 1075. 1988.
- [6] Deering S, Waitzman D, Partridge C, Estrin D, Farinacci D, Jacobson V, Helmy A, Meyer D, Wei L. Protocol Independent Multicast Version 2 Dense Mode Specification. draft-ietf-pim-v2-dm-01.txt. 1998.
- [7] May J. Multicast Routing Extensions for OSPF. Commun. ACM 1994; 37 (8): 61-66.
- [8] Ballardie T, Francis P, Crowcroft J. Core Based Trees (CBT) – An Architecture for Scalable Inter-Domain Multicast Routing. Proc. ACM SIGCOMM '93, 1993. p. 85-95.
- [9] Estrin D, Deering S, Farinacci D, Jacobson V, Liu C, Wei L, Helmy A, Thaler D, Handley M. Protocol Independent Multicast - Sparse Mode (PIM SM): Protocol Specification. RFC 2362. 1998.
- [10] Kosiur D. IP Multicasting. New York: Wile, 1998.
- [11] Boivie R, Feldman N, Mets C. Small Group Multicast: A New Solution for Multicasting in the Internet. IEEE Internet Computing 2000. p. 75-79.
- [12] Meyer D, Fenner B. Multicast Source Discovery Protocol (MSDP). draft-ietf-msdp-spec-

10.txt. 2001.

[13] Kou L, Markowsky G, Berman L. A Fast Algorithm for Steiner Tree. Acta Informatica 15, 1981. p.141-45.

[14] Wang CF, Liang CT, Jan RH. Heuristic Algorithms for Packing of Multiple-Group Multicasting. Computers & Operation Research 2002, 29; 905-924.

[15] CPLEX 7.0, CPLEX Optimization Inc. 2002.