대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일-4일

Multiple Multicast Tree Allocation Algorithm in Multicast Network

Chae Y. Lee and Hee K. Cho
Department of Industrial Engineering, KAIST,
373-1, Kusung Dong, Daejon, 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. However, there have been few researches about the relation between multiple and concurrent multicast groups. In this paper, the multiple multicast tree allocation algorithm to avoid congestion is proposed. The multicast group with different bandwidth requirement is also considered.

A two-phase algorithm is proposed. The first phase is for basic search and the second phase for further improvement.

The performance of the proposed algorithm is experimented with computational results. Computational results show that the proposed algorithm outperforms an existing algorithm.

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 many emerging applications are of multicast nature such as teleconferencing, remote education and collaborative applications.

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 these effort is concerned to a single multicast group. In this paper, we are interested in multiple and concurrent multicast groups.

Previous works about multiple multicasts are quite limited. In [2], a formulation and an elegant solution for accommodating two multicast streams (audio and video) on a small size network (8-nodes) is presented. A heuristic approach and a lower bound of optimal solution are provided for multiple multicasts by Yener et al [3]. However, in [3], the capacity of each link in the network is not considered. The heuristic proposed in [3] solves Steiner tree problem at each iteration. It requires very high complexity. Wang et al. [14] also considered the problem of multiple multicast and proposed heuristic algorithms. Heuristic algorithms in [4] find a set of multicast trees to have the minimized overall cost of these trees, while those in [3] find a set of multicast trees to minimize the network congestion. However, the required bandwidth for each multicast group in those papers is assumed to be all identical.

In this paper we propose an efficient heuristic with low complexity for multiple multicast tree allocation in which the link capacity is considered and the required bandwidth for each multicast group is different. The problem is to allocate several concurrent multicast traffic to the network such that bottlenecks are avoided on the links.

This paper is organized as follows. In section II, we explain the multiple multicast tree allocation problem and its formulation. We also propose efficient heuristic algorithms for multiple multicast tree allocation. The computational results are presented in section III, and we finally conclude our work in section IV.

II. Multiple Multicast Tree Allocation

The multiple multicast tree allocation problem is explained with formulation. The Yener's algorithm [3] is modified to consider the link capacity in the network. An efficient multiple multicast tree allocation algorithm is proposed that has lower complexity than the Yener's procedure. Above two algorithms are based on the assumption in which the required bandwidth for each multicast group is all identical. Thus we discuss the effect of the case in which each multicast group requires different required bandwidth in the remainder of this section.

1. The Multicast Tree Allocation Problem

Multicasting is an efficient scheme for transmitting 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 considered with the size of multicast tree. The number of links on the multicast tree is limited such that the delay threshold is satisfied implicitly. 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 is represented by a set of nodes $M \subset V$. A border router connected to a remote sender is also contained in M. In the network, we are interested in finding a subgraph G' that spans the multicast group M and satisfies a certain subgraph selection criterion. Yener et al [3], select the subgraph which minimizes the network congestion. The traffic load of the most congested link is minimized in the network. However, this criterion is not applicable 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 under the assumption that each link has different capacity. The residual capacity which is extra capacity of a link is another measure of congestion. High minimum residual capacity of a link represents better chance of other best effort unicast traffic transport.

Since our objective of the multiple multicast tree allocation is to maximize the minimum residual network capacity as opposed to minimizing individual multicast tree costs, the solution to the multiple multicast allocation may result in high-cost multicast trees. To maximize the minimum residual network capacity, some trees that include the congested links need to be reconstructed without including the congested links. In the process of reconstruction, the size of the tree may be increased. The increased number of links in the multicast tree clearly delays the real time internet traffic. We thus need to restrict the size of each multicast tree to a certain degree. Consider the least size of the multicast tree OPT^k for multicast group k for $k \in K$, 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^k_e for all $e \in E$ and $k \in K$. Let N be the set of nodes in multicast group k. If a link e is used for the tree for the multicast group k, then $x^k_e = 1$. The traffic load of multicast group k is denoted by t^k which is assumed discrete in traffic unit and the minimum residual capacity by r.

All members of a multicast group must be connected by a tree for delivering multicast packet from a sender. Eq. (1) shows that all members in a multicast group 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\S by d(S).

$$\Sigma_{e \in d(S)} x^k_e \geq 1 \quad \text{for all } k \in K \text{ and all } S \subset V, \text{ and } N \not\subset S \qquad (1)$$

Note that the multicast traffic has to satisfy the network bandwidth. Multicast traffics that pass through a link need to satisfy the link capacity. Since our objective is to avoid bottleneck in the network we consider the residual capacity in addition to the kink capacity constraint. The extra capacity r in Eq. (2) thus has the effect of systematically distributing traffics in congested links to relatively idle links.

$$C_e - \Sigma_{k \in K} t^k x^k_e \geq r \text{ for all } e \in E \qquad (2)$$

Now, in the process of distributing the congested traffic a multicast tree may experience delay due to the extended tree with relatively idle links. 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. (3) the OPT^k represents the number of links in the Steiner tree for nodes in multicast group k.

$$\Sigma_{e \in E} x_e^k \leq (\alpha OPT^k) \text{, for all } k \in K \qquad (3)$$

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:

$$\begin{aligned}
&\text{Max} \quad r \\
&\text{s.t.} \quad \Sigma_{e \in d(S)} x^k_e \geq 1 \quad \text{for all } k \in K \\
&\qquad\qquad \text{and all } S \subset V, \text{ such that } N \not\subset S \\
&C_e - \Sigma_{k \in K} t^k x^k_e \geq r \text{ for all } e \in E \\
&\Sigma_{e \in E} x_e^k \leq (\alpha OPT^k) \text{, for all } k \in K \\
&r \geq 0
\end{aligned}$$

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.

2. Modified Yener's Algorithm [3]

Basically, in multiple multicast tree allocation problem, each multicast tree is constructed independently for initial solution set. We thus focus on reconfiguring multiple multicast trees from the initial solution set to maximize the minimum residual

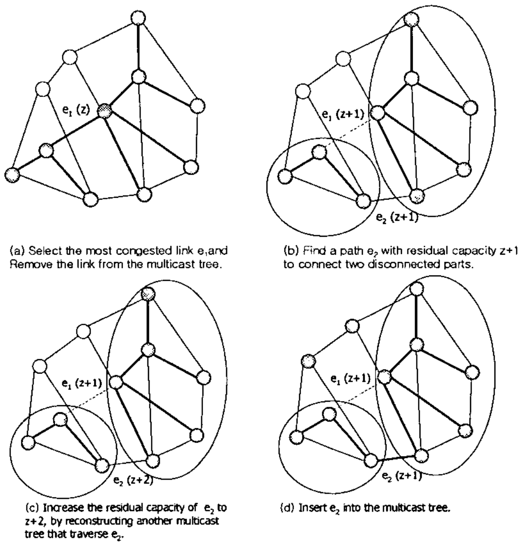

Fig. 1. An example of the second phase

network capacity.

Note that the heuristic algorithm by Yener el al [3] presents a solution to multicast tree allocation problem without the tree capacity constraint of each link. We modify the algorithm to consider the link capacity. In the modified algorithm, residual capacity is used as a measure of congestion and links are sorted by nonincreasing order of residual capacity. We also assume each multicast group has the same traffic unit, i.e., $t^k = 1$, in this modified heuristic. The algorithm at each iteration chooses the most congested link *e* and finds a tree which employs the link. The algorithm removes the congested link from a multicast tree T_k and reconstructs a tree using a Steiner tree algorithm. If the size of the newly constructed tree T is within the range of αOPT^k, then T_k is replaced with T and T becomes a new multicast tree for the multicast group *k*. If such a tree T is not found, the algorithm is terminated. Otherwise the algorithm continues by reordering the residual capacity.

In this algorithm, Steiner tree problem is solved at each iteration. We use the well known KMB algorithm by Kou et al. [1] to solve Steiner tree problem. The time complexity of KMB is $O(nv^2)$. *n* is the number of member for a multicast group. *v* is the number of routers in a network. We assume *m* is the expected iteration of the algorithm. Then, overall complexity of modified Yener's algorithm is $O(mnv^2)$.

3. The Multiple Multicast Tree Allocation Algorithm

The modified Yener's algorithm is relatively complex because the Steiner tree needs to be searched every iteration, and hard to apply to real network. Thus we propose a more efficient multiple multicast tree allocation algorithm that can be applicable to real network. First, we propose an algorithm by assuming each multicast group has the same traffic unit, i.e., $t^k = 1$. After that, we propose a scheme in which each multicast group requires different bandwidth.

In the algorithm for $t^k = 1$, 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 algorithm 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 a shortest path is found. Note that to increase the residual capacity of the network each link that connects the two disconnected parts 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 has to satisfy the tree size limit 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, we search a path with z+1 residual value to connect the two disconnected parts within the tree size limit. After finding the path with z+1 residual capacity, we select a multicast tree T' that traverses the link *l* with z+1 residual capacity. The multicast tree T' is reconstructed after deleting the link by using the same procedure in phase 1. Since T' 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 path and the residual network capacity is improved to z+1.

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

If any path with z+1 residual value is not found in the second phase, the algorithm is terminated. Fig. 2 shows the procedure of the proposed algorithm. In the algorithm, the most congested link with minimum residual capacity is selected and one among the multicast groups which traverse the link is selected to be reconstructed as shown in step 2 of Fig. 2. 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 random selection.

Now, we consider each multicast group requires different bandwidth. Finding a solution in this case is more difficult than the case of all identical bandwidth. In step 2 of Fig. 2, a multicast group *k* is selected among groups that traverse the most congested link. The resultant minimum residual value

```
Preprocessing:
1. for each multicast group, construct a multicast tree independently.
2. for each link, compute residual capacity.
Input: set of multicast trees computed independently T={T_k: k = 1, ..., |K|} and a
bound on the tree size αOPT^k.
Output: Revised multicast trees

BEGIN
- Sort and Update Process
1. Sort all links e by z_e into array LE in increasing order.
2. Choose a link e' with maximum congestion z and a group k that traverses the link.
- Phase 1
3. Remove the link e' from tree T_k and partition tree into two parts
4. Find a path to connect two parts with at least z+2 residual capacity.
5. IF there exists such a path within the range αOPT^k for the group k.
        THEN insert the path into T_k and update z_e values in LE and goto step 1.
  ELSE
        IF Steiner tree problem is solved for all links and groups with maximum
          congestion z.
                THEN goto step 6.
        ELSE
                THEN choose another link e' and group k with maximum congestion z
                      and goto step 3.
- Phase 2
6. Find a path to connect two parts with z+1 residual capacity.
7. 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 in LE and goto step 1
  ELSE
        THEN stop
END
```

Fig. 2 The procedure of the proposed algorithm

is dependent on the multicast group selected. Clearly, selecting the multicast group that requires the highest bandwidth among multicast groups which traverse the most congested link leads to the highest residual value of the link. We call this scheme as the highest-bandwidth-first. However, this scheme has a disadvantage of decreasing the residual value of the path to connect two disconnected parts of multicast tree T_k selected in step 2. As a result, the highest-bandwidth-first scheme may not find better solution. The example of this case is shown in Fig. 3. Each number in the middle of link represents the link capacity in the figure. In Fig. 3 (a), there are two multicast groups. Group 1 consists of node 1 and 5, requires 4 bandwidth units, and has 1-4-5 multicast tree. Group 2 consists of node 5 and 6, requires 2 bandwidth units, and has 6-4-5 multicast tree. The most congested link is link (4,5) and the residual value is 1 (=7-4-2). Fig 3 (b) shows that group 1 is selected at link (4,5) by using the highest-bandwidth-

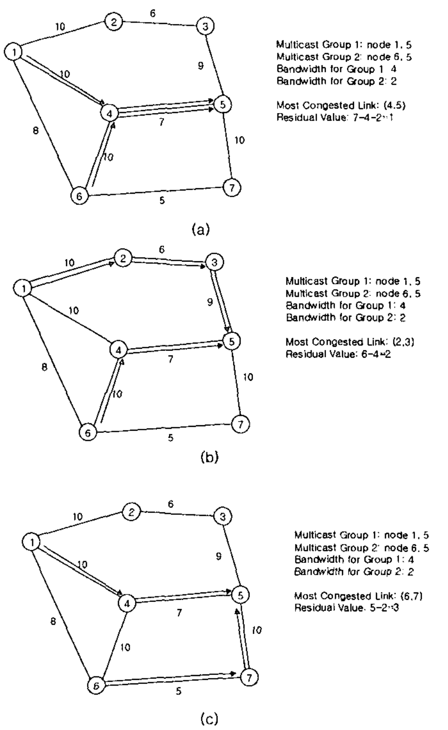

Fig. 3. Comparison of max required bandwidth first and max residual alternative first

first. In this case, the most congested link (2,3) has residual value 2 (=6-4). However, better solution is found by selecting group 2 instead of group 1 at the most congested link in Fig.3 (a). The result of selecting group 2 is shown in Fig. 3 (c). This example shows that the highest-bandwidth-first scheme does not always result in the best solution.

Let's define *alternative gain* (k) as the difference of residual value of the network before and after reconstructing a multicast group k. From Fig. 3, it is clear that selecting the group k to have the largest *alternative gain* (k) is better than the highest-bandwidth-first. We call this scheme as highest-residual-gain-first.

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 a shortest path between two partitioned parts. The time complexity of finding a shortest path is $O(v^2)$, where v is the number of routers in the network. Thus, by assuming m as the expected iteration of the algorithm, the overall complexity of the algorithm becomes $O(mv^2)$ in the case of $t^k = 1$. Considering that each multicast group requires different bandwidth, the time complexity of max required bandwidth first is also $O(mv^2)$ due to the computing of the shortest path once at each iteration. The complexity of max residual alternative first is $O(kmv^2)$ because computing the shortest path is computed k times in the worst case, where k is the number of multicast group.

III. Computational Results

The performance of the proposed algorithm is experimented with computational results. Ten random networks with N = 800 nodes and |E| = 2N are generated. The value of parameter α is set to two.

Fig. 4 shows the performance of the algorithms under the assumption of $t^k = 1$ for each group k. The number of group members varies from 5 to 25 with 10 and 20 groups. Fig. 4 shows that tree allocation scheme without any adjustment (separate multicast) has the worst performance. The proposed algorithm using only phase 1 without phase 2 has worse performance than Yener's algorithm. However, the figure shows that the proposed algorithm with two phases have 3-8% better minimum residual value than Yener's procedure. This shows that phase 2 of the proposed algorithm improves the solutions significantly.

Fig. 5 shows the impact of multicast tree selection scheme among trees share the most congested link. In view of packet delivery delay, small-group-first scheme has the better performance than random selection. Delay is measured by hop counts from sender to receiver.

Next, we investigate the performance of the proposed algorithm with tree selection schemes (highest- bandwidth-first and highest-residual-gain-first) in problems where t^k varies from 1 to 5 for each multicast group k. In Fig. 6, the highest-residual-gain-first has higher minimum residual value than the highest-bandwidth-first. The figure shows the gap of the solution between two schemes increases as the group size increases.

Ⅳ. Conclusion

In this paper, the multiple multicast tree allocation algorithm to avoid congestion is proposed. The multicast group with different bandwidth requirement is also considered..

The proposed algorithm consists of two phases. The first phase is for basic search and the second phase for further improvement. In the first phase, the most congested link is found and the link from the corresponding multicast tree for partitioning the multicast group is removed. Then, the alternative path is inserted into the corresponding multicast tree to connect two partitioned parts of the multicast group. The second phase is implemented when no improved path is found to connect the two disconnected parts in the first phase.

The performance of the proposed algorithm with computational results is experimented. Computational results show that the proposed algorithm outperforms the existing algorithm.

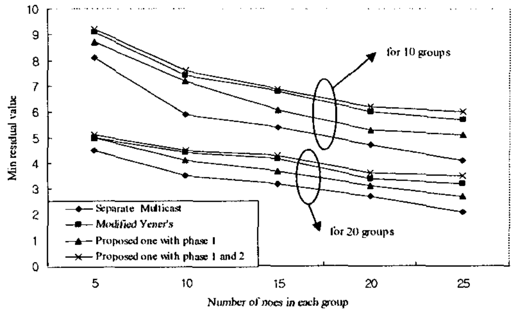

Fig. 4 The performance of the algorithms

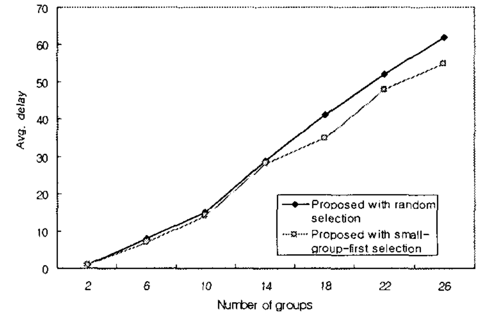

Fig. 5 The performance of the proposed algorithm as the selection scehem

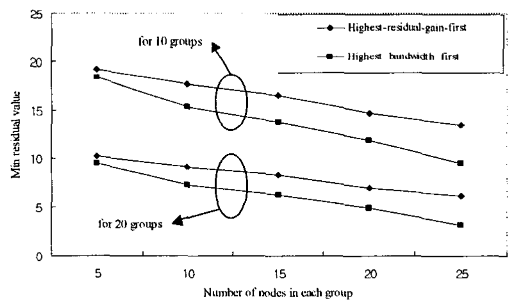

Fig.6 The performance of highest-bandwidth-first and highest-residual-gain-first

Reference

[1] L. Kou, G. Markowsky, and L. Berman, "A Fast Algorithm for Steiner Tree," *Acta Informatica 15*, 1981, pp.141-45.

[2] C. A. Noronha and F. A. Tobagi, "Optimum routing of multicast streams," in *IEEE INFOCOM*, pp. 865-873,1994.

[3] Shiwen Chen, Oktay Gunulk, and Bulent Yener, "The Multicast Packing Problem," *IEEE J. Select. Areas Commun.* Vol. 8, No.3, pp. 311-318, June, 2000.

[4] Chu-Fu Wang, Chun-Teng Liang, Rong-Hong Jan, "Heuristic Algorithms for Packing of Multiple-Group Multicating," Computers & Operation Research. No.

29, pp. 905-924, 2002.

[5] D. Waitzman, S. Deering and C. Partridge, "Distance Vector Multicast Routing Protocol," *RFC 1075*, November 1988.

[6] S.Deering et al, " Protocol Independent Multicast Version 2 Dense Mode Specification," *Internet Draft*, November, 1998.

[7] J. May, "Multicast Routing Extensions for OSPF," *Commun. ACM*, vol. 37, no. 8,, pp. 61-66, 114, August, 1994.

[8] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT) – An Architecture for Scalable Inter-Domain Multicast Routing," *Proc. ACM SIGCOMM '93, San Francisco, CA*, pp. 85-95, Oct. 1993.

[9] D. Estrin et al, "Protocol Independent Multicast - Sparse Mode (PIM SM): Protocol Specification," *RFC 2362*, June, 1998.

[10] D. Kosiur, *IP Multicasting*. New York: Wile, 1998