A Delay-Scaling Multicast Algorithm with Multiple QoS Criteria

Li Chen*   Guoliang Xue†   Byung S. Lee‡

Abstract

A delay-scaling multicast algorithm (DSMCA) is presented for minimum-cost multicast tree construction that also considers end-to-end delays along the paths from the source to each multicast group member. This problem is known to be NP-complete. Our heuristic has a time complexity of $O(D^2p^2n^2 \lg n)$, where n is the number of nodes in the network, p is the number of destinations in the multicast session, and D is the delay constraint. Our simulation results show that DSMCA on average produces trees with lower cost than other known heuristics.

1 Introduction

Multicasting consists of concurrently sending the same information from a source to multiple destinations in a computer network. Multicasting falls between unicasting and broadcasting. The most popular solutions to multicast routing involve tree construction. There are two reasons for basing efficient multicast routes on trees: (i) the data can be transmitted in parallel to the various destinations along the branches of the tree; and (ii) a minimum number of copies of the data are transmitted, with duplication of data being necessary only at forks in the tree [2]. Algorithms computing multicast trees are called multicast algorithms.

As multimedia applications such as video conferencing and voice over Internet are gaining popularity, multicast routing algorithms capable of satisfying multiple QoS requirements will be essential for future high-speed networks. Delay, delay jitter, bandwidth, and packet loss probability are types of QoS. Most QoS criteria fall into one of two categories. One is of *additive* type, such as delay, loss and jitter. Bandwidth, in turn, belongs to the class of *bottleneck* requirements, which are much easier to handle than "additive" requirements.

Given a multicast group and a set of optimization objective functions, a set of constraints in the form of end-to-end delay bound, minimum bandwidth, delay jitter bound, constraint-based multicasting routing is a process of constructing, based on the network topology and the network state, a multicast tree that optimizes the objective functions while satisfying the constraints. The objective function could be defined to minimize the delay of each path from the source to a multicast group member, which is important for delay-sensitive multimedia applications, such as real-time teleconferencing. We call this kind of objectives *performance-driven*. Or it could be defined to minimize the cost of a multicast tree, which is important in managing network resources efficiently. We call this kind of objectives *cost-driven* [5]. This is known as the Steiner tree problem and is NP-complete [7].

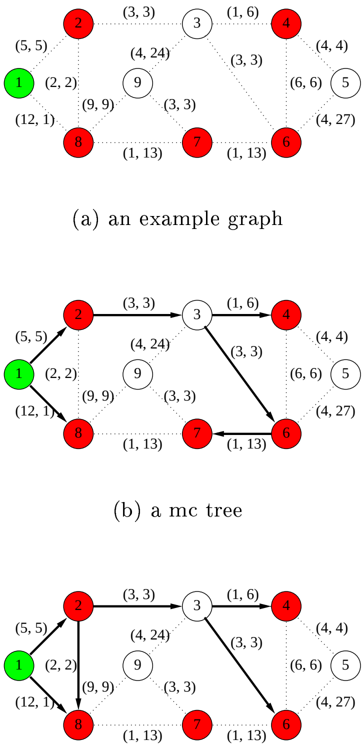
Figure 1(a) shows an example network where the edge labels are the (delay, cost) pair. Edge cost could be a measure of resource utilization, e.g., the amount of buffer space or channel bandwidth used, and edge delay could be a combination of propagation, transmission, and queuing delay. The multicast group is $\{1, 2, 4, 6, 7, 8\}$, where 1 is the source vertex. The delay constraint for all destinations is 12. Figure 1(b) shows a multicast tree with a cost of 26. This is not a delay-constrained multicast tree, since the 1–7 delay is 15, which is larger than the delay constraint 12. Figure 1(c) shows a delay-constrained multicast tree with a cost of 32.

Some multicast routing algorithms are based on building a least-delay source-to-destination tree. This approach does not always lead to efficient network resource utilization, since it does not necessarily optimize sharing of the network links. A number of new multicast routing algorithms designed specifically for real-time applications were proposed during the past few years [2, 3, 4, 8]. Parsa, Zhu and Garcia-Luna-Aceves [3] proposed a heuristic algorithm, called the bounded shortest multicast algorithm (BSMA), to construct minimum-cost multicast tree with delay constraints. BSMA starts by constructing an least-delay tree. Then it iteratively replaces superedges in the tree with lower cost superedges not in the tree, without violating the delay

*Li Chen is a graduate student in the Department of Computer Science, The University of Vermont, Burlington, VT 05405. Email: lchen@cs.uvm.edu. The research of this author was supported in part by ARO grant DAAD19-00-1-0377.

†Guoliang Xue is an Associate Professor with the Department of Computer Science and Engineering at Arizona State University, Tempe, AZ 85287-5406. Email: xue@asu.edu. He is also an adjunct Associate Associate Professor with the Department of Computer Science at the University of Vermont, Burlington, VT 05405. The research of this author was supported in part by ARO grant DAAD19-00-1-0377 and by DOE grant DE-FG02-00ER45828.

‡Byung S. Lee is an Assistant Professor with the Department of Computer Science, The University of Vermont, Burlington, VT 05405. Email: bslee@cs.uvm.edu.



(a) an example graph

(b) a mc tree

(c) a delay-constrained mc tree

Figure 1: Example network, mc trees and delay-constrained mc trees

bound, until the total cost of the tree cannot be reduced any further. Kompella, Pasquale, and Polyzos [2] proposed a heuristic algorithm, called the KPP heuristic. KPP first computes a delay constrained closure graph over the multicast group. Then, KPP uses Prim's algorithm to obtain a minimum spanning tree of the closure graph. The edge selected each time is the one which does not violate the delay constraint and minimize a selection function. Observing that every multicast tree has a Steiner topology, Xue and Xiao [8] proposed a fully polynomial time approximation scheme for computing a minimum cost delay-constrained multicast tree under a Steiner topology. This paper presents a new algorithm for the construction of a minimum-cost multicast tree with delay bounds. Consistent with the assumptions made in all prior approaches to delay-constrained minimum-cost multicasting, throughout this paper we assume that a node running DSMCA has complete topology information.

The rest of this paper is organized as follows. In section 2, we define the problem model. In section 3, we review Steiner topologies and discuss the relationship between minimum cost delay-constrained multicast trees and minimum cost delay-constrained realizations of a Steiner topology. In section 4, we present our DSMCA algorithm. In section 5, we present simulation results.

2 Problem Definition

We model a communication network by an undirected edge weighted graph $N = G(V, E, c, l)$, where V is the set of n nodes, E is the set of m edges. Nodes represent routers or switches, and edges represent the communication link between them. $c(e) = c(u, v) = c(v, u) \geq 0$ is the cost of an edge $e = (u, v) = (v, u) \in E$. $l(e) = l(u, v) = l(v, u) \geq 0$ is the delay that packets experience on edge $e = (u, v) = (v, u) \in E$. Let G' be a subgraph of G. The cost of G', denoted by $c(G')$, is the sum of the edge costs over the edges in G'. Let π be a path in G. The delay of π, denoted by $l(\pi)$, is the sum of the edge delays over the edges on π.

A multicast request is given by a tuple $(s; \Delta; t_1, t_2, \ldots, t_p)$ where $s \in V$ is the source vertex, $t_1, t_2, \ldots, t_p$ are p destination vertices and Δ is the corresponding delay constraint. A multicast tree for the given multicast request is a tree subgraph of N which spans the source vertex and all the destination vertices. The multicast tree that we are interested in constructing is a delay-bounded minimum Steiner tree(DBMST) and the problem can be formally described as follows.

DBMST problem: Given a graph $N = G(V, E, c, l)$ and a multicast request, construct a delay-bounded minimum Steiner tree T spanning the source vertex and all the destination vertices, such that the cost of the tree is minimized while delay constraint is satisfied. If π_i is the unique s–t_i path in T,

$$\text{minimize} \quad \textstyle\sum_{e \in T} c(e) \qquad (2.1)$$
$$\text{s.t.} \quad \textstyle\sum_{e \in \pi_i} l(e) \leq \Delta \qquad (2.2)$$

For example, during a teleconference, it is important that the speaker be heard by all participants within a bounded time; otherwise the teleconference may lack the feel of an interactive discussion.

3 Tree topologies and their realizations

We will need the following concepts introduced in Xue and Xiao [8]. **Definition 1** Let $\mathcal{V} = \{\sigma; \tau_1, \tau_2, \ldots, \tau_p; \omega_1, \omega_2, \ldots, \omega_q\}$ be a set of $1 + p + q$ vertices. A tree $\mathcal{T} = G(\mathcal{V}, \mathcal{E})$ is called a tree topology for source vertex s and destination vertices $t_1, t_2, \ldots, t_p$ if the degree of ω_j is at least 3 for $j = 1, 2, \ldots, q$. $\omega_1, \ldots, \omega_q$ are called Steiner vertices. A tree topology $\mathcal{T}$ is called a Steiner topology if

Table 1: *Two realization of the topology in Figure 2*

δ	τ_1	τ_2	τ_3	τ_4	τ_5	ω_1	ω_2	ω_3	ω_4
1	2	4	6	7	8	8	8	2	3
1	2	4	6	7	8	2	8	2	3

1. $q = p - 1$;
2. the degree of ω_j is exactly 3 for $j = 1, 2, \ldots, p-1$;
3. the degree of σ or τ_i is exactly 1 for $i = 1, 2, \ldots, p$.

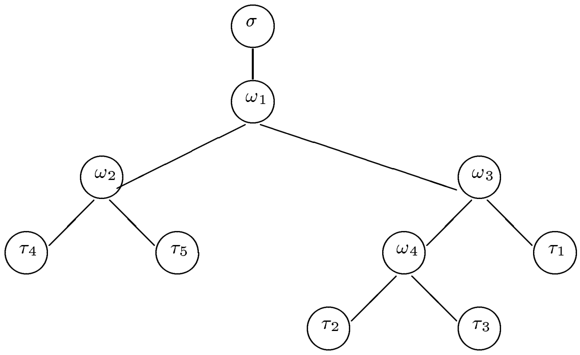


Figure 2: *A Steiner topology corresponding to the multicast tree in Figure1(c)*

Definition 2 Let $\mathcal{T}$ be a tree topology. A realization $\mathcal{R} = \mathcal{R}(\mathcal{T})$ of $\mathcal{T}$ is a pair of mappings $\mathcal{R}_\mathcal{V}$ and $\mathcal{R}_\mathcal{E}$ where $\mathcal{R}_\mathcal{V}$ maps $\mathcal{V}$ to V such that

1. $\mathcal{R}_\mathcal{V}(\sigma) = s$;
2. $\mathcal{R}_\mathcal{V}(\tau_i) = t_i$ for $i = 1, 2, \ldots, p$; and
3. $\mathcal{R}_\mathcal{E}$ maps each edge (μ, ν) in $\mathcal{T}$ to a $\mathcal{R}_\mathcal{V}(\mu) - \mathcal{R}_\mathcal{V}(\nu)$ path in G.

The delay of (μ, ν) in N induced by realization $\mathcal{R}$ and delay function l, denoted by $l(\mu, \nu; \mathcal{R})$, is the delay of the $\mathcal{R}_\mathcal{V}(\mu) - \mathcal{R}_\mathcal{V}(\nu)$ path in N. The cost of (μ, ν) in N induced by realization $\mathcal{R}$ and cost function c, denoted by $c(\mu, \nu; \mathcal{R})$, is the cost of the $\mathcal{R}_\mathcal{V}(\mu) - \mathcal{R}_\mathcal{V}(\nu)$ path in N.

Definition 3 A realization $\mathcal{R}$ of a tree topology $\mathcal{T}$ is a delay- constrained realization if the induced path delay from σ to τ_i is no more than Δ for $i = 1, 2, \ldots, p$. The cost of $\mathcal{R}$, denoted by $c(\mathcal{R})$, is the sum of the induced edge costs of all the edges in $\mathcal{T}$. The cost of $\mathcal{T}$, denoted by $c(\mathcal{T})$, is the minimum among the costs of all delay-constrained realizations of $\mathcal{T}$.

Table 1 shows two realizations of the Steiner topology in Figure 2. The first realization in Table 1 has a cost of 28 and does not satisfy the delay constraints. The second realization is a delay-constrained realization with a cost of 32.

Xue and Xiao[8] proved the following theorem.

Theorem 1 Let T be a minimum cost delay-constrained multicast tree. Let $\mathcal{T}$ be a minimum cost Steiner topology. Then $c(\mathcal{T}) = c(T)$, i.e., there exists a minimum cost delay-constrained realization $\mathcal{R}$ of $\mathcal{T}$ such that $c(\mathcal{R}) = c(T)$. □

Computing a minimum cost delay-constrained realization of a Steiner topology is NP-hard [8]. However, Xue and Xiao presented a polynomial time approximation scheme for this problem. In the following section, we will apply this technique to design a delay-scaling algorithm for computing a good Steiner topology and a good approximation to the minimum cost delay-constrained multicast tree. Computational results show that in all cases, the delay bound is satisfied and the cost of our tree is smaller than those produced by previous heuristics.

4 Description of DSMCA

In this section, we present our algorithm, DSMCA (Delay Scaling Multicast Algorithm). The name comes from the fact that the algorithm works by scaling delays. Given a delay constraint D, we can compute single source delay-constrained minimum cost s–v paths for all $v \in V$ whose delay is at most $d \in \{1, 2, \ldots, D\}$. This can be accomplished in $O(mD + nD\log(nD))$ time by a modification of Dijkstra's algorithm [8]. We will use $L(u, v, d)$ to denote the minimum cost of a u–v path whose delay is bounded by d. In [8], Xue and Xiao presented a dynamic programming algorithm for computing a minimum cost delay constrained multicast tree under a Steiner topology and delay constraint D. The time complexity of that algorithm is $O(D^2 n^2 \lg p)$. Due to space limitation, we will not describe that algorithm here but will use that as a subroutine in the design of our algorithm.

DSMCA constructs a Steiner topology by starting from the source node and iteratively adding one multicast destination at a time until the topology is completed. At each iteration, the least-cost topology of all possible different topologies is selected. For Steiner topology $\mathcal{T}_p$ for the case of p-destinations, there are $2p - 1$ edges in $\mathcal{T}_p$. We can obtain $2p - 1$ different Steiner topologies $\mathcal{T}_{p+1}$ for the case of $(p+1)$-destinations, obtained by joining the new destination to one of the $2p-1$ existing edges in $\mathcal{T}_p$. Therefore, at the ith iteration, there are $2i - 1$ different topologies. The least-cost one is selected as $\mathcal{T}_{i+1}$. The basic steps of the DSMCA heuristic are described in Algorithm

1. The Running time is $O(n^4 p^2 \lg p)$, where p is the size of the multicast group and n is the number of nodes in the network.

Algorithm 1 DSMCA

step_1 Scale the delay of the links by $\frac{n}{\Delta}$.

step_2 Let $\mathcal{T}$ be the unique Steiner topology for source s and destinations t_1 and t_2. Using the algorithm of [8] to compute an low cost delay constrained multicast tree T under this topology. Set $k := 2$.

step_3 while $k < p$ do
  for each of the $2k - 1$ edges in $\mathcal{T}$ do
    connect t_{k+1} to that edge to form a new Steiner topology;
    compute the minimum cost delay-constrained multicast tree under this topology;
  endfor
  Let $\mathcal{T}$ be one of the $2k - 1$ topologies that yields the minimum cost;
  $k := k + 1$;
endwhile

step_4 Compute and output an approximation to the minimum cost delay constrained multicast tree under topology $\mathcal{T}$.

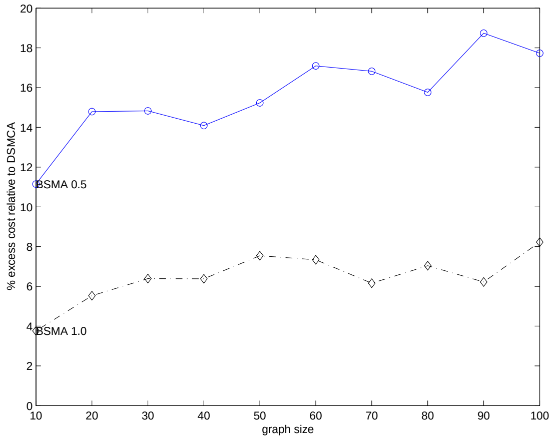
5 Simulation results

DSMCA has been implemented in C++. A C++ class library, LEDA is used in software implementation. The experiments were carried out on a PIII desktop with 512M of memory and running RedHat Linux 7.0. We used random graphs of a given node size n and edge size m. The link cost and delay are also random numbers uniformly distributed in the interval [1, 100]. Group members were picked uniformly from the set of nodes in the graph, excluding the nodes already selected for the group.

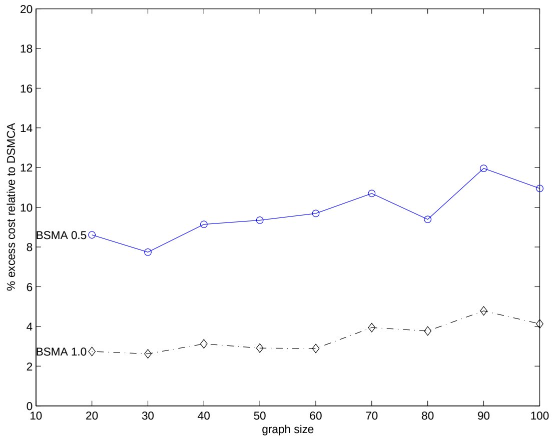
In the experiment, the average degree of the random graphs is 4. The size of the graph was varied between 10 and 100 nodes. A total of more than 40,000 runs were performed. Two different sizes were used for the number of destinations in the runs: 5 and 10. The cost of each multicast tree produced by BSMA is normalized by the cost of our DSMCA algorithm for the same group instance. The resulting cost ratio ρ is averaged over the number of groups K, i.e.

$$\rho = \frac{1}{K} \sum_{i=1}^{K} \left(\frac{cost(BSMA)}{cost(DSMCA)} - 1 \right) \qquad (5.1)$$

The results are shown in Figure 3. The label BSMA1.0 is given to the cost ratio of BSMA with the delay bound equal to the delay of minimum-cost solution obtained using Dijkstra's algorithm. The label BSMA0.5 is given to the cost ratio of BSMA with the delay bound halfway between the two extremes of the minimum-cost and minimum-delay solutions.



(a) 5 destinations



(b) 10 destinations

Figure 3: Normalized reduction in cost.

From Figure 3, we observe that when the delay is tight, on average BSMA's cost was higher than DSMCA's cost by at least 10%. When the delay is increased, the difference becomes smaller because there are fewer delay-bound violations.

The processing time was computed for each of the simulated heuristic. The overall average running times of each heuristic are shown in Table 2.

Table 2: Average processing time in seconds for multicast group size of 6

No. of nodes	BSMA	DSMCA
10	0.02	0.12
20	0.07	0.44
30	0.12	0.99
40	0.28	1.74
50	0.38	2.67
60	0.50	3.76
70	1.02	5.35
80	1.22	6.96
90	1.56	8.76
100	1.76	11.04

6 Conclusions

In this paper, a new delay-bounded minimum-cost multicast heuristic is presented. DSMCA constructs a Steiner topology by starting from the source node and iteratively adding one multicast destination at a time until the topology is completed. At each iteration, the least-cost topology of all possible different topologies is selected. Our simulation results show that DSMCA on average produces lower cost trees than other known heuristics.

References

[1] J. Walrand, P. Varaiya, *High-performance communication networks*, Morgan Kaufmann Publishers, 2000.

[2] V. P. Kompella, J. C. Pasquale, and G. C. Polyzo, Multicast routing for multimedia communication, *IEEE/ACM Transactions on Networking*, pages 286-292, 1993

[3] M. Parsa, Q. Zhu and J. J. Garcia-Luna-Aceves, An iterative algorithm for delay-constrained minimum-cost multicasting *IEEE/ACM Transactions on Networking*, Vol. 6(1998), pp. 461–474.

[4] A. Goel, K. G. Ramakrishnan, D. Kataria and D. Logothetis, Efficient computation of delay-sensitive routes from one source to all destinations, *IEEE INFOCOM'2001.*

[5] G.L. Xue, Provably Good Approximations to Minimum Cost Delay-Constrained Multicast Tree, *IEEE international Conference on Computer and Communications and Networks*, pp. 610–614.

[6] H. F. Salama, D. S. Reeves, and Y. Viniotis, Evaluation of multicast routing algorithms for real-time communications on high-speed networks, *IEEE J. Select. Areas Commun.*, Vol. 15, pp. 332–345, Apr. 1997

[7] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York:Freeman, 1979

[8] G.L. Xue, W. Xiao, *Approximations to minimum cost delay-constrained multicast tree under a steiner topology, with applications*, *IEEE/ACM Transactions on Networking*, under review.

[9] L. H. Sahasrabuddhe, B. Mukherjee Multicast routing algorithms and protocols: A tutorial, *IEEE Network*, pp. 90-102, Jan/Feb 2000