1

MULTICASTING FOR MULTIMEDIA APPLICATIONS

Vachaspathi P. Kompella
Joseph C. Pasquale
George C. Polyzos

Computer Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114

Abstract

We investigate multicast routing for high-bandwidth delay-sensitive applications in a point-to-point network as an optimization problem. We associate an edge cost and an edge delay with each edge in the network. The problem is to construct a tree spanning the destination nodes, such that it has the least cost, and so that the delay on the path from source to each destination is bounded. Since the problem is computationally intractable, we present an efficient approximation algorithm. Experimental results through simulations show that the performance of the heuristic is near optimal.

I. Introduction

Recent advances in communication technology are making packet video and audio communication over computer networks a reality. With them come the expectation that multiparty communication will become a popular interactive mode. In order to support the high data rates and considerably stringent delay constraints imposed by these media, new routing algorithms must be designed that address these issues. In this paper, we define the problem of multicasting in this context, and present a routing algorithm that constructs near-optimal cost delay-bounded multicast routes. A good discussion of research on multicasting can be found in [1]. Recent work in providing multicast service on the Internet can be found in [2].

This work was funded by NSF, DEC, IBM, NCR, Loral Instrumentation, and the Powell Foundation.

Previous optimization techniques for multicast routing have considered two optimization goals, delay optimization and cost optimization, but as distinct problems. Delay optimization is defined as follows. In a graph $G = (V, E)$, with node set V, edge set E, and delay function $\mathcal{D} : E \rightarrow \mathbb{R}^+$, the optimal delay solution is such that the sum of delays on the edges along the path from source to each destination is minimum. Dijkstra's shortest path algorithm [3] can be used to generate the shortest paths from the source to the destination nodes in $O(n^2)$ time in a graph with n nodes. This provides the optimal solution for delay optimization.

Cost optimization is defined as follows. In a graph $G = (V, E)$, with cost function $\mathcal{C} : E \rightarrow \mathbb{R}^+$, a cost optimized multicast route is a tree spanning the destinations such that the sum of the costs on the edges of the tree is minimum. This problem is also known as the Steiner tree problem [4], and is known to be *NP*-complete [5]. However, some heuristics for the Steiner tree problem have been developed that take polynomial time [6, 7, 8], and produce near optimal results.

Multicast routing algorithms that perform cost optimization have been based on Steiner tree heuristics. Wall [9] studied how the Steiner tree algorithm of Kou, Markowsky and Berman could be used to generate multicast trees. Waxman [10] examined the problem of reconfiguration of the tree if nodes join or leave the tree dynamically. Chow [11] looked at the problem of multiparty connections from two points of view. Firstly, the problem of designing a good multicast route was considered in terms of cost optimality with least computation time. Secondly, the inverse problem of combining multiple communications into one route (concast) was addressed.

Of the previous work done on multicast route optimization, only Kadaba and Jaffe discuss optimization on both cost and delay [12]. However, they assume that the cost and delay functions are identical. The problem we address here considers two independent functions for cost and delay. The results of Kadaba and Jaffe do not hold if the two functions are different. The other point of difference is that we do not optimize on delay, but rather search for solutions with bounded delays.

Ferrari and Verma [13] describe a procedure for establishing routes that account for constrained delay for unicast connections. They do not attempt to optimize on the routing, assuming that routing is addressed by a lower protocol layer. The extension of constrained delay unicast routing to multicasting is non-trivial, and we show that the optimization problem is *NP*-complete.

In this paper, we shall consider multicast routing as a source routing problem, with each node having full knowledge of the network and its status. We shall also assume that connections will be virtual circuits, since they are the more natural mechanism for continuous media like audio and video.

The organization of the paper is as follows: section 2 contains a formal statement of the constrained multicast tree problem, and details a heuristic that we developed to solve the problem. Section 3 describes a set of experiments to evaluate the performance of the algorithm. Section 4 discusses the results of the experiments, and Section 5 presents some conclusions on the problem, the algorithm, and its applications.

II. The Optimal Constrained Multicast Tree Problem

We describe the optimal constrained multicast tree as follows. We have a point-to-point network represented by a graph $G = (V, E)$, with

$$\mathcal{C} : E \rightarrow \mathbb{R}^+, \text{a real edge cost function}$$

and

$$\mathcal{D} : E \rightarrow \mathbb{Z}^+, \text{an integer edge delay function}$$

A multicast on this graph is defined by three parameters: a source node s, a destination node set S, and a delay constraint D. A constrained multicast tree (CMT) is a tree rooted at s that spans the set S, such that the sum of the delays along the path from s to each destination $v \in S$ is bounded above by D. An optimal CMT is a CMT with the least cost, *i.e.*, with the least sum of costs on the edges of the tree.

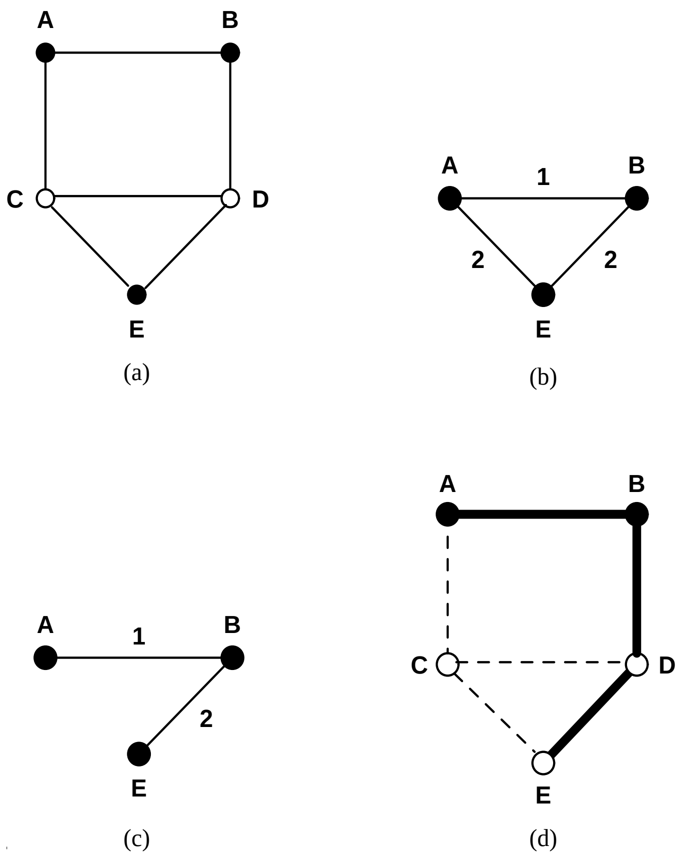

Figure 1: The KMB algorithm. (a) The graph. (b) The closure. (c) The MST. (d) The Steiner tree.

This problem is *NP*-complete (as shown in Appendix 2), but we have developed a heuristic to construct a near optimal CMT based on a Steiner tree approximation algorithm due to Kou, Markowsy, and Berman [6]. We briefly describe the KMB algorithm, and demonstrate it on the graph in Fig. 1. Assume that all the edges have unit cost and unit delay. The set of nodes to be spanned is {B, E}, and A is the source.

1. The first step of the KMB algorithm is to construct a closure graph G' on the set $S \cup \{s\}$. The closure is a complete graph on the nodes in S, with the cost of an edge (u, v) being the least cost path between u and v in G (Fig. 1b). The path costs are shown

along the edges of the closure graph.

2. The second step is to construct a minimum spanning tree T of G' (Fig. 1c).

3. Finally, the edges in T are expanded to the edges in G that make up the least cost paths of G' (Fig. 1d).

Note that this algorithm does not necessarily generate a CMT. However, by adding constraints to the various steps of the algorithm, we can ensure that the solution is a CMT.

Before we can describe the CMT algorithm, we need to define the *shortest constrained path* between two nodes. The shortest constrained path between u and v is defined as the path with least cost, subject to the delay along the path being less than D, the delay constraint. We can construct the shortest constrained path as follows. Let $\mathcal{C}(u,w)$ be the cost on edge (u,w), and $\mathcal{D}(u,w)$ be its delay. Define $\mathcal{W}(u,w)$ to be the cost of the shortest constrained path between u and w. Define $\mathcal{W}_d(u,w)$ to be the cost of the shortest path between u and w with delay equal to d. Then,

$$\begin{aligned} \mathcal{W}_d(u,w) &\triangleq \min_{v \in V} \{\mathcal{W}_{d-\mathcal{D}(v,w)}(u,v) + \mathcal{C}(v,w)\} \\ \mathcal{W}(u,w) &\triangleq \min_{0 \le d < D} \mathcal{W}_d(u,w) \end{aligned}$$

The CMT algorithm follows the steps of the KMB algorithm, being careful not to violate the delay constraint. We describe the CMT algorithm next, and apply it to the same graph as in Fig. 1.

1. The first step is to construct the constrained closure G'. Here, instead of setting the cost of an edge in G' to the cost of the shortest path, we set it equal to the cost of the shortest constrained path (Fig. 2b). The path delays and path costs are shown along the edges of the closure graph.

2. In the second step, we construct a constrained minimum spanning tree T of G' (Fig. 2c). We use Prim's technique of constructing a minimum spanning tree [14], with the following critical measure, c_m, that determines which edge to add to the subtree at each iteration:

$$c_m = \begin{cases} \frac{\mathcal{W}(u,v)}{D-\mathcal{D}_P(u)-\mathcal{D}'(u,v)} & \text{if } \mathcal{D}_P(u) + \mathcal{D}'(u,v) < D \\ \infty & \text{otherwise} \end{cases}$$

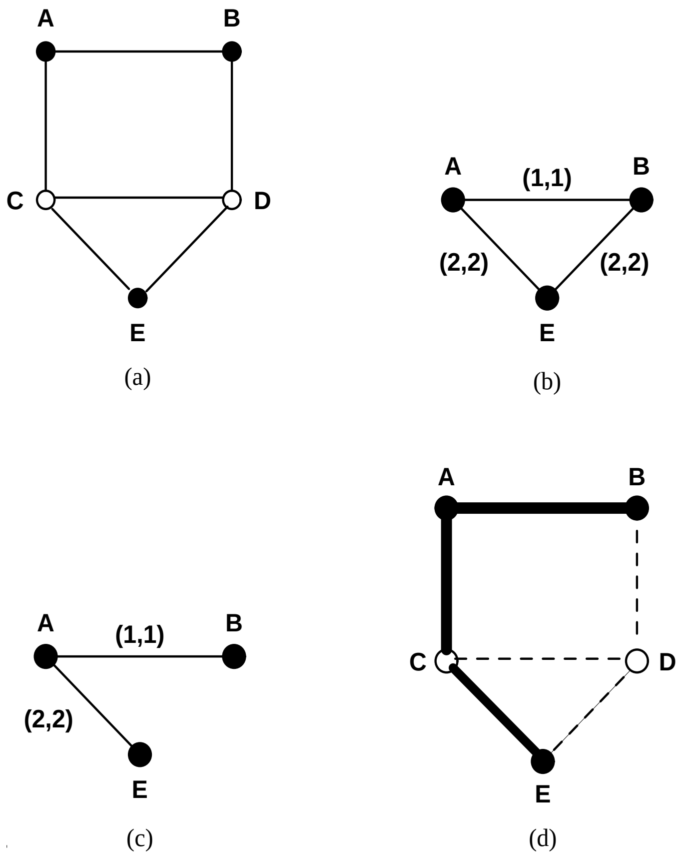

Figure 2: The MCT algorithm. (a) The graph. (b) The closure. (c) The MST. (d) The constrained multicast tree.

where

$$\mathcal{D}_P(u) = \text{delay along the path from source } s \text{ to } u$$

and

$$\mathcal{D}'(u,v) = \text{delay along the shortest constrained path from } u \text{ to } v$$

3. The last step is to expand the edges of the tree T into the edges that constitute the shortest constrained paths (Fig.2d).

The critical measure balances the greedy choice of the least cost edge (u,v) against the residual delay, *i.e.*, the delay left over from the path from s to v. The higher the residual delay, the better the chances are that the same route can be used to send data to another destination without exceeding the delay constraint.

For the example in Fig. 1, the Steiner tree produced by the KMB algorithm violates the delay constraint $D = 3$. Fig. 2 shows the steps involved in constructing a constrained multicast tree, with A as the source and {B, E} as the destination set, given $D = 3$. Pseudo-code for the CMT algorithm is listed in Appendix 1.

We now show that the CMT algorithm always finds a solution if one exists. Then we demonstrate the worst case performance of the CMT algorithm in terms of the cost of the constrained multicast tree, when compared with the optimal solutions.

Lemma 1: A constrained multicast tree exists *iff* there are k edges incident on s with finite cost in the constrained closure graph, where $|S| = k$.

Proof: Obvious, since there must exist at least one path under the delay constraint D from s to each of the k destinations for a solution.

Theorem 1: The CMT algorithm finds a solution *iff* a solution exists.

Proof: Follows from Lemma 1 and the observation that any edges chosen in forming the tree from the closure graph do not violate the delay constraint.

The complexity of the algorithm is dominated by the computation of the all-pairs shortest constrained paths, which takes $O(n^3\,D)$. The rest of the algorithm takes $O(k^3)$, where $|S| = k$. Although this algorithm is not polynomial, since it takes time $> 2^{\log D}$, it is close enough to a polynomial solution in practical situations, given that D is from a bounded set of well-known values of delay tolerance for video and audio transmission.

Let the optimal tree be T_O, and the CMT tree be T_{CMT} for some problem, and their costs be $|T_O|$ and $|T_{CMT}|$, respectively. The performance of the CMT algorithm can be measured by the ratio ρ, where

$$\rho = \frac{|T_O|}{|T_{CMT}|}$$

Theorem 2: For the CMT algorithm, the worst case ratio is given by

$$\rho_{max} = \frac{k\,l}{k+l}$$

Proof: The graph in Fig. 3 shows a graph with this ratio. This would occur if all the shortest constrained paths are disjoint, and the delays along them are just under D, so that it is not possible to concatenate any paths without violating the delay constraint. Thus, the star graph out of s would be the only possible solution from the closure, giving a cost of $k\,l$. In the graph in Fig. 3, $v_1, v_2, \ldots, v_k$ are the destination nodes, and s is the source. The shortest constrained paths from s to v_i are shown as edges, each of which is of length l. However, there is a path from s to v of cost l, and from v to v_i of cost 1, such that the path delay is under the delay constraint. These paths form the optimal CMT of cost $(k + l)$.

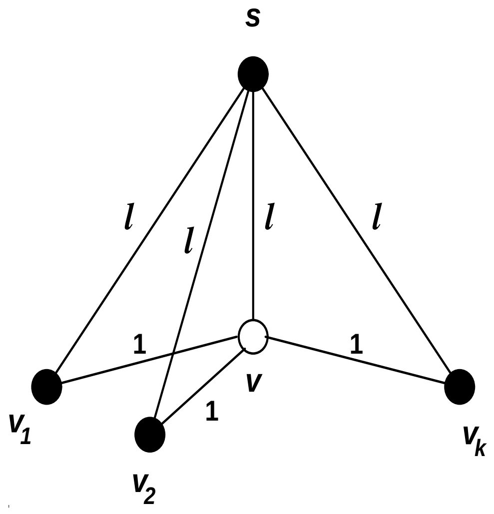

Figure 3: A worst case example for the CMT algorithm, with $\rho = \frac{k\,l}{k+l}$

III. Experiments

One of the important measures of the utility of the CMT algorithm is its average case performance. To determine this, we ran the algorithm on a number of randomly generated graphs. We generated a constrained multicast tree using the CMT heuristic. We also found the optimal solution by enumerating all constrained spanning trees, and picking the least cost tree. Finally, we produced a constrained spanning tree by randomly picking edges until a solution was found. We chose to consider that the tree generated randomly would be as bad a solution as we could find. We compared the costs of the different trees. We chose to measure two figures of merit. Firstly, we computed the average distance from the optimal solution normalized by the cost of the optimal solution. We call this the average normalized distance, denoted by δ, where

$$\delta = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{M_i - O_i}{O_i}\right)$$

This measure can be thought of as the percentage over the optimal cost incurred by using the CMT algorithm. It gives an indication of how efficient the algorithm is when the penalty of being sub-optimal is amortized over a number of multicasts.

The second measure is a rating of the CMT algorithm between the performance of the optimal solution (best) and random solution (worst, in the sense described above). We call this measure the efficiency η, of the algorithm.

$$\eta = \frac{\sum_{i=1}^{N}(R_i - M_i)}{\sum_{i=1}^{N}(R_i - O_i)}$$

In the above,

$$\begin{aligned} M_i &= \text{cost of the CMT tree in the } i\text{th run} \\ O_i &= \text{cost of the optimal tree in the } i\text{th run} \\ R_i &= \text{cost of the random tree in the } i\text{th run} \end{aligned}$$

Graphs were generated randomly with the following characteristics. Every graph had to have at least one solution. A number of graphs of different sizes were tried. Every node had a degree between 1 and the maximum degree. Each edge had unit cost, and a randomly assigned delay, uniformly distributed over the set $\{1, \ldots, 8\}$. Various destination set sizes were tried, with the destinations randomly selected. Different delay constraints were also tried. An experiment generated N graphs with the same number of nodes n, maximum degree d, destination set size g, and delay constraint D. The experiments had the following parameters:

number of runs N	10 000
number of nodes n	10, 11, ..., 20
maximum node degree d	4, 5
size of destination set g	3, 4, 5
edge cost, $\mathcal{C}(e)$	1
edge delay, $\mathcal{D}(e)$	$1 \le \mathcal{D}(e) \le 8$, uniformly random
delay constraint D	15, 20, 25, 30

Table 1: Parameters for the experiments

IV. Results

The efficiency ratio, η, is a measure of how close to perfect we are on a scale of 0 to 1, with 0 being the worst, and 1 the best. From Fig. 4, we notice that the efficiency increases with an increase in the delay constraint.

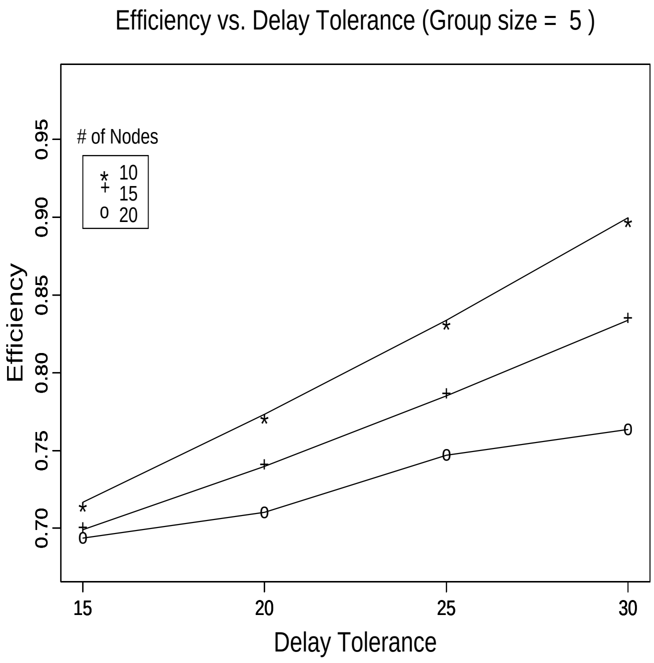

Figure 4: Efficiency as a function of delay tolerance for various group sizes

In fact, if D becomes infinite, then the algorithm reduces to the KMB algorithm, since the numerator is the only significant factor in c_m.

In Fig. 5, we notice that efficiency decreases with an increase in node size, but approaches a stable value. This is because the average delay on an edge is about 4. With a delay constraint of 25, the average number of hops on a path will be around 6 edges. Thus, since the nodes farther from the source will not be reachable under the delay constraint, the node size does not play a significant role beyond a certain point, for a given delay tolerance and maximum degree. The average normalized distance δ, can be seen from Fig. 6 to be quite low, ranging between 1 and 7%. This shows that the absolute performance of the algorithm is good. We notice that this measure also increases with node size, but also reaches a plateau for the same reason that the nodes at a distance will be unreachable under the delay constraint.

Another observation is that over the 2.6 million graphs generated through the experiments, the largest value for the ratio δ was 2. Note that the worst case value of Theorem 2 is applicable only if nodes have unbounded degree.

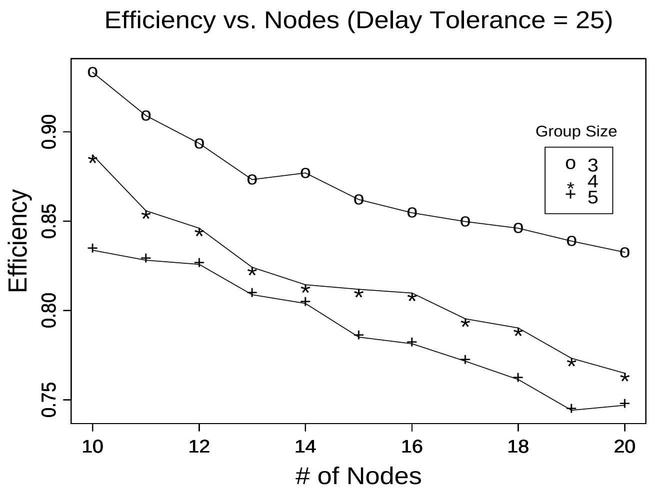

Figure 5: Efficiency as a function of number of nodes for various group sizes

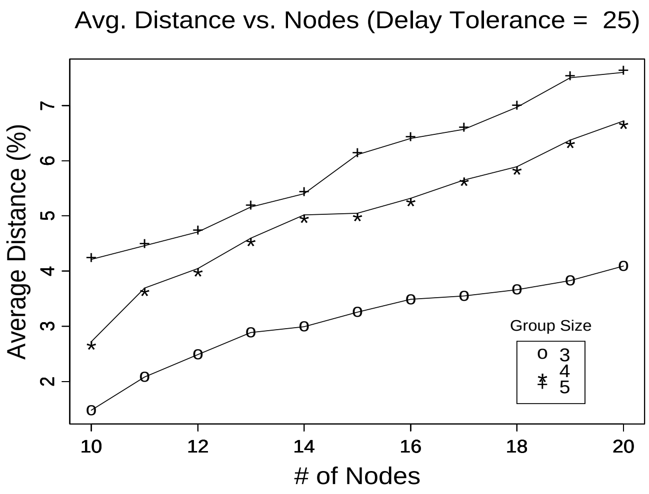

Figure 6: Normalized distance of multicast solution from optimal for various numbers of nodes

Finally, the time taken to run these experiments on a DECstation 5000 was about 12.5 days. Most of the time was needed to compute the optimal solution. The random solution also took considerable time because the tree cost for the random solution was actually an average of the cost of 50 randomly constructed trees.

V. Conclusions

We have considered the problem of multicasting for delay sensitive data. In this paper we have assumed a source routing based solution. Our CMT routing algorithm minimizes the cost of a multicast while ensuring that the delay between the source and any destination is bounded by a specified value. We have presented an $O(n^3\, D)$ solution. Since in practice the value of D is chosen from a finite set, this algorithm should be regarded as polynomial. Empirical results show that this heuristic has good performance in an absolute sense (i.e., when compared with the optimal). In addition, comparisons with random solutions satisfying the delay constraint show that the optimization is worthwhile.

The heart of the algorithm operates on a condensed graph and builds a tree by selecting edges based on a metric which considers both the cost and the residual delay. We have also been considering other critical measures that may give rise to improved performance in the case of source routing [15]. That paper also presents distributed routing algorithms for solving this problem and discusses their performance. These distributed algorithms use information about adjacent nodes only,rather than requiring knowledge of the entire network topology.

We believe that the CMT algorithm and similar approaches are called for in carrying audio and video data over the network because of the kind of traffic characteristics these media have. Typically, audio data will need bandwidths between 64Kbits/sec and 1Mbit/sec, depending on the sound quality, and require an end-to-end delay of less than 100 msec [16]. Video data has an order of magnitude higher data rate, with about 30 to 50 msec delay tolerance [17]. Taking the cost on an edge to be a function of the bandwidth utilized, and the delay to be a sum of the switching and propagation delays, the CMT algorithm could be applied to generate multiparty connections for multimedia applications.

Acknowledgements

We would like to thank Mike Saks for his help in the *NP*-completeness proof, and suggestions on approaches to solving the constrained multicast tree problem. We would also like to thank Ted Billard for his remarks on the original statement of the problem.

Appendix 1

```
/*
 * Let G = (V, E) describe the network topology
 * s = source node
 * S = multicast group
 * D = delay constraint
 */
Multicast (G(V, E), s, S, D)
begin
   /* assume the cheapest constrained paths have
    * been found between all nodes in S ∪ {s}
    */
   V' ← S ∪ {s}
   for each v, w ∈ V' do
   begin
      W[v, w] ← cost of cheapest constrained
                path from v to w
      D[v, w] ← path delay along cheapest
                constrained path from v to w
   end
   /* C = set of nodes already visited */
   /* P[v] = path delay from s to v in the tree */
   C = {s}
   P[s] = 0
   /* until all nodes in V' have been spanned */
   while (C ≠ V') do
   begin
      min = ∞
      for each v ∈ C do
          for each w ∈ V' \ C do
          begin
             /* if delay from s to w is within limits,
              * consider edge (v, w) as a candidate
              */
             if (c_m(v, w) < min and
                  D[v, w] + P[v] < D) then
             begin
                nextedge = (v, w)
                min = c_m(v, w)
                u = w
             end /* if */
          end /* for */
          C = C ∪ {u}
          P[u] = P[v] + D[v, u]
      end /* for */
   end /* while */
end /* Multicast */
```

Appendix 2

Theorem: Given a graph $G = (V, E)$, with edge weights and edge delays, destination set S, source s, and delay constraint D, finding the optimal constrained multicast tree problem is NP-complete.

Proof: The problem is in NP, since a non-deterministic "guess" can list a set of edges that form the tree, and in deterministic time, it is possible to check:

a) the edges do form a tree

b) the nodes of S are all covered

c) the path delay from s to each node in S is under the delay constraint D.

The problem is NP-hard. Assume there exists a deterministic polynomial time algorithm A for the problem. Then given a Steiner tree problem, we can construct a multicast tree problem as follows. Take any node in the set S to be the source node s. Let the delay constraint be the longest simple path delay. A solution given by A is exactly the solution for the Steiner tree problem. Thus, the optimal constrained multicast tree problem is NP-complete. This proof holds even for edge weights = 1, since the corresponding Steiner tree problem is NP-complete [18].

References

[1] A. Frank, L. Wittie, and A. Bernstein, "Multicast communication in network computers," *IEEE Software*, vol. 2, no. 3, pp. 49–61, 1985.

[2] S. Deering and D. Cheriton, "Multicast routing in internetworks and extended lans," *ACM Trans. on Comp. Sys.*, vol. 8, pp. 85–110, May 1990.

[3] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[4] S. Hakimi, "Steiner's problem in graphs and its implications," *Networks*, vol. 1, pp. 113–133, 1971.

[5] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations* (R. Miller and J. Thatcher, eds.), pp. 85–103, New York: Plenum Press, 1972.

[6] L. Kou, G. Markowsky, and L. Berman, “A fast algorithm for Steiner trees,” *Acta Informatica*, no. 15, pp. 141–145, 1981.

[7] V. Rayward-Smith, “The computation of nearly minimal Steiner trees in graphs,” *Intl. J. Math. Educ. Sci. Tech.*, vol. 14, no. 1, pp. 15–23, 1983.

[8] H. Takahashi and A. Matsuyama, “An approximate solution for the Steiner problem in graphs,” *Math. Japonica*, vol. 6, pp. 573–577, 1980.

[9] D. Wall, *Mechanisms for Broadcast and Selective Broadcast*. PhD thesis, Elect. Engin. Dept., Stanford University, Jun 1980.

[10] B. Waxman, “Routing of multipoint connections,” *IEEE J. on Selected Areas in Comm.*, vol. 6, pp. 1617–1622, Dec 1988.

[11] C.-H. Chow, “On multicast path finding algorithms,” *Proc. of IEEE INFOCOMM*, pp. 1274–1283, 1990.

[12] B. Kadaba and J. Jaffe, “Routing to multiple destinations in computer networks,” *IEEE Trans. on Comm.*, vol. COM-31, pp. 343–351, Mar. 1983.

[13] D. Ferrari and D. Verma, “A scheme for real-time channel establishment in wide-area networks,” *IEEE J. on Selected Areas in Comm.*, vol. 8, pp. 368–379, Apr. 1990.

[14] R. Prim, “Shortest connection networks and some generalizations,” *Bell Systems Tech. J.*, vol. 36, pp. 1389–1401, 1957.

[15] V. Kompella, J. Pasquale, and G. Polyzos, “Two techniques for multicast routing for multimedia networking,” Tech. Report CSL-1005-91, Computer Systems Laboratory, University of California, San Diego, Dec 1991.

[16] “Engineering and operations in the Bell System,” AT&T Bell Laboratories, 1984.

[17] D. Anderson, S.-Y. Tzou, R. Wahbe, R. Govindan, and M. Andrews, “Support for continuous media in the DASH system,” *ICDCS 10*, 1990.

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