Optimum Routing of Multicast Streams *

Ciro A. Noronha Jr. Fouad A. Tobagi

Department of Electrical Engineering
Stanford University
Stanford, CA 94305

Abstract

We show that the problem of optimally routing multicast streams can be formulated as an integer programming problem. We propose an efficient solution technique, composed of two parts: (i) an extension to the decomposition principle, to speed up the linear relaxation of the problem, and (ii) enhanced value-fixing rules, to prune the search space for the integer problem. We characterize the reduction in run time gained using these techniques. Finally, we compare the run times for the optimum multicast routing algorithm and for existing heuristic algorithms.

1 Introduction

The traffic underlying networked multimedia applications has different requirements than that underlying traditional data applications, as follows: (i) *Bandwidth* - multimedia streams use relatively high bandwidth on a continuous basis for long periods of time; (ii) *Multipoint Communications*; and (iii) *Low Latency*, required for interactive communication.

Multicast is the name applied to the unidirectional communication from one source to multiple destinations. The high bandwidths required to transmit real-time audio/video streams suggest that the routing of such streams should be performed as efficiently as possible, to maximize the number of multicasts the network is able to carry. The objective of this work is to propose and characterize an algorithm for optimum multicast routing, addressing the specific requirements in terms of bandwidth, multicasting, and latency underlying multimedia traffic.

*This work was in part supported by NASA under grant NAGW-419, by NSF under grant NCR-9016032, by Pacific Bell and by AT&T. Ciro Noronha was also supported by a graduate scholarship from FAPESP under grant 89/1658.

2 Elements of the Problem

A *stream* is a continuous flow of information (i.e., video frames and/or audio samples) that have to be delivered in a timely fashion. Several streams can be multiplexed in time and sent through a channel, each stream using a fraction of the channel's bandwidth. Given the network and the traffic requirements, the *multicast routing algorithm* is responsible for finding a route in the network that satisfies the requirements, while optimizing a certain objective function, which can be the delay, the cost or a combination thereof.

2.1 The Network

The network is a collection of *nodes*, interconnected by *links* subject to a certain *topology*. We consider all links to be point-to-point and directed (i.e., information can flow in only one direction). Full-duplex connections between pairs of nodes correspond to two independent links, one in each direction. Each link is characterized by: (i) *Capacity* - link bandwidth, in bits/second; (ii) *Cost* - monetary cost of using the link, in \$/bit; and (iii) *Delay* - time between the instant a bit of information becomes ready to be transmitted at the source of the link, and the instant it is received at the other end; it is assumed to be constant. Formally, a network with N nodes and K links is denoted by $G(\boldsymbol{A}, \boldsymbol{W})$, where $\boldsymbol{A}$ is the $N \times K$ topology matrix (its (i,j) element is 1 if node i is the origin of link j, -1 if it is the destination, and 0 otherwise) and $\boldsymbol{W}$ is the link parameters vector, containing the capacity, cost and delay for each of the links.

2.2 The Traffic

Multicast streams arrive at the network in *sessions*. A session is a group of multicasts which are logically related. One example is a video-conference, which can be established as a set of simultaneous multicasts. We will denote the number of multicasts in the session by T. Each of these multicasts is characterized by its addressing parameters (source and set of destinations), its bandwidth requirement, and its latency constraint.

We denote by *multicast path* the route used to establish the multicast stream on the network. Formally, the multicast path MP for a certain request is a tree, formed by merging the paths from its source to each of its destinations. Given a multicast path for a certain multicast stream, we define the *delay of the multicast* to be the maximum of the delays between the source and each of the destinations.

3 Existing Algorithms for Multicast Routing

When a multicast session arrives at the network, the multicast routing algorithm is responsible for finding routes for each of its components; each route should have enough free bandwidth to support the stream, and should not exceed the latency constraint. If multiple routes exist for a given multicast, the routing algorithm will choose one so as to optimize a certain objective function. Existing algorithms are able to route only a single multicast, without directly taking into account the bandwidth and latency constraints. To compute the routes for a multiple-multicast session, these algorithms have to be applied sequentially to each multicast in the session, in a given order. For each multicast in the session, in order to take the bandwidth requirements into account, the network topology must be pruned of the links that do not have enough free bandwidth to support the stream, prior to routing it. Finally, after the route has been computed, its delay has to be checked against the latency constraint; if it is not satisfied, the session has to be blocked.

As far as the objective function is concerned, existing algorithms can be classified into: (i) Shortest-Path Algorithms; and (ii) Minimum-Cost Algorithms.

3.1 Shortest-Path Algorithms

Given a multicast, the routes are computed independently from the source to each destination, using the shortest path; the paths are then merged into a single tree. There are several exact algorithms available to compute the shortest path [1]. If the link labels used in the routing are the link delays, this approach will yield the minimum delay routing from the source to each of the destinations. Other measures, such as cost, can also be used for link labels; in this case, the routes have no special properties. This algorithm is used in [2] and in the Multicast OSPF routing protocol [3], where the link weights can be set arbitrarily by the network administrator.

3.2 Minimum-Cost Algorithms

Given a multicast, the routing is done so as to minimize the sum of the labels of the links used; if the link labels are set to the link costs, this approach will lead to the minimum cost multicast routing. This is the well-known problem of finding minimum-cost Steiner trees in graphs, which is known to be NP-complete [4]. Many exact solutions (with exponential worst-case run times) and heuristics have been proposed to address this problem; see [5] for a comprehensive survey of the field. An important heuristic solution is the algorithm by Kou, Markowsky and Berman [6] (known as KMB in the literature). Kompella et al [7] proposed a variation of the KMB heuristic to compute minimum-cost routes with a delay constraint; the link delays are assumed to belong to a discrete set.

4 A General Formulation for the Optimum Multicast Routing

In this section, we present an integer programming formulation for the optimum multicast routing problem.

4.1 Problem Definition

Formally, the problem can be stated as: "Given a network $G(\boldsymbol{A}, \boldsymbol{W})$ and a set of T multicast streams, with multicast i, $i = 1, \ldots, T$, being characterized by its source s_i, its set of n_i destinations $\{d_{i1}, \ldots, d_{in_i}\}$, its maximum delay constraint L_i and its bandwidth request r_i, find the set of multicast paths that satisfies the bandwidth and delay constraints for each stream and minimizes an arbitrary linear combination of the multicast costs and delays."

4.2 Mathematical Formulation

Definitions:

$\boldsymbol{N}$	:	Number of nodes in the network.
$\boldsymbol{K}$	:	Number of (directed) links in the network.
$\boldsymbol{A}$	:	$N \times K$ network topology matrix.
$\boldsymbol{C}$	:	$1 \times K$ cost vector.
$\boldsymbol{D}$	:	$1 \times K$ delay vector.

$\boldsymbol{V}$: $K \times 1$ capacity vector.
T : Number of multicast streams.
s_i : Source node for multicast i
n_i : Number of destinations for multicast i
$\{d_{ik}\}$: Set of destinations for multicast i, $k = 1, \ldots, n_i$
r_i : Bandwidth requirement for multicast i
$\boldsymbol{X}^i$: $K \times n_i$ multicast routing matrix for multicast stream i. $X^i_{jk} = 1$ if link j is used in the multicast path for stream i to reach destination d_{ik}, otherwise $X^i_{jk} = 0$, $k = 1, \ldots, n_i$.
$\boldsymbol{Y}^i$: $K \times 1$ multicast path vector for stream i. $Y^i_j = 1$ if link j is in the multicast path for stream i, otherwise $Y^i_j = 0$.
M_i : Delay for multicast request i.
L_i : Latency constraint for multicast request i.
$\boldsymbol{B}^i$: $N \times n_i$ source-destination matrix for multicast stream i; $B^i_{jk} = 1$ if $j = s_i$, $B^i_{jk} = -1$ if $j = d_{ik}$, and $B^i_{jk} = 0$ otherwise, $k = 1, \ldots, n_i$.
β_c : Weight of the cost in the optimization.
β_d : Weight of the delay in the optimization.

The optimization problem can be formulated as follows:

GIVEN: $\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{V}, N, K, T, \boldsymbol{B}^i, r_i, \boldsymbol{L}, \beta_c, \beta_d$

MINIMIZE:

$$\sum_{i=1}^{T} r_i \left(\beta_c \boldsymbol{C} \boldsymbol{Y}^i + \beta_d M_i \right) \tag{1}$$

WITH RESPECT TO: $\boldsymbol{X}^i, \boldsymbol{Y}^i, M_i, \quad i = 1, \ldots, T$

UNDER CONSTRAINTS:

$(k = 1, \ldots, n_i, j = 1, \ldots, K, i = 1, \ldots, T)$

1. There must be a path from the source to each of the destinations for all the streams:

$$\boldsymbol{A}\boldsymbol{X}^i = \boldsymbol{B}^i \tag{2}$$

2. The multicast path is formed by merging the unicast routes:

$$X^i_{jk} \le Y^i_j \le 1 \tag{3}$$

3. The delay for a multicast is the delay to the farthest destination:

$$M_i - \sum_{j=1}^{K} D_j X^i_{jk} \ge 0 \tag{4}$$

4. There is a maximum delay constraint for each of the multicast streams:

$$M_i \le L_i \tag{5}$$

5. The total flow through a link cannot exceed its bandwidth:

$$\sum_{i=1}^{T} r_i \boldsymbol{Y}^i \le \boldsymbol{V}; \tag{6}$$

6. No bifurcation of flow:

$$\boldsymbol{X}, \boldsymbol{Y} \quad \text{are binary.} \tag{7}$$

Constraints (1) to (6) define a linear programming problem, which can be solved by the simplex method. When constraint (7) is included, the problem becomes an integer programming problem[1].

5 Solution of the Optimum Multicast Routing Problem

In this section, we present an efficient solution technique for the integer programming problem presented in section 4.2. The technique is based on the well-known branch-and-bound method [8], which has two parts: (i) the linear relaxation (where the integer constraints are relaxed, and the problem is solved as a linear problem) and (ii) the branch and bound phase. We discuss each of the phases separately.

5.1 Solution to the Linear Relaxation

Although the linear relaxation could be solved using the traditional simplex method, it is possible to apply an extension of the decomposition procedure [9] to obtain a solution more efficiently. In other words, it is possible to decompose the problem of routing T multicast streams into T single-multicast routing problems; each of these single-multicast problems can be further decomposed into n_i unicast routing problems. The required extension is shown in the Appendix.

[1] In a packet-switched network, it is conceivable that bifurcation of flow could be allowed. In this case, constraint (7) does not apply, and the problem is no longer NP-complete.

5.1.1 Decomposing the Main Problem

Observe that equations (2), (3), (4) and (5) apply to each multicast in isolation, while equation (6) is the only connection between flows belonging to different multicasts. Rewriting these equations, and using positive slack variables to turn inequalities into equalities, we find:

GENERAL CONSTRAINTS (valid for $i = 1, \ldots, T$):

$$\begin{aligned} \boldsymbol{A}\boldsymbol{X}^i &= \boldsymbol{B}^i \\ X^i_{jk} - Y^i_j + Z^i_{jk} &= 0 \\ Y^i_j + S_{Yj} &= 1 \\ \sum_{j=1}^{K} D_j X^i_{jk} - M_i + S^i_{ak} &= 0 \\ M_i + S_{Li} &= L_i \end{aligned}$$

$$k = 1, \ldots, n_i, \quad j = 1, \ldots, K$$

(Z^i_{jk}, S_{Y_j}, S^i_{ak} and S_{L_i} are positive slack variables)

COMPLICATING CONSTRAINTS:

$$\sum_{i=1}^{T} r_i \boldsymbol{Y}^i + \boldsymbol{S} = \boldsymbol{V}$$

Let $\boldsymbol{I}_K$ denote the $K \times K$ identity matrix; we define:

$$\mathcal{X}_i = \left[\boldsymbol{X}^{i\prime}_1 \cdots \boldsymbol{X}^{i\prime}_{n_i} | \boldsymbol{Y}^{i\prime} | M_i | \boldsymbol{Z}^{i\prime}_1 \cdots \boldsymbol{Z}^{i\prime}_{ni} | \boldsymbol{S}^{\prime}_Y | \boldsymbol{S}^{i\prime}_a | S_{Li}\right]' \tag{8}$$

$$\mathcal{A}_i = \left[\begin{array}{ccc|c|c|c|c|c|c}
\boldsymbol{A} & \cdots & 0 & & & & & & \\
\vdots & & \vdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \cdots & \boldsymbol{A} & & & & & & \\
\hline
\boldsymbol{I}_K & \cdots & 0 & -\boldsymbol{I}_K & & & & & \\
\vdots & & \vdots & \vdots & 0 & \boldsymbol{I}_{Kn_i} & 0 & 0 & 0 \\
0 & \cdots & \boldsymbol{I}_K & -\boldsymbol{I}_K & & & & & \\
\hline
\multicolumn{3}{c|}{0} & \boldsymbol{I}_K & 0 & 0 & \boldsymbol{I}_K & 0 & 0 \\
\hline
\boldsymbol{D} & \cdots & 0 & & -1 & & & & \\
\vdots & & \vdots & 0 & \vdots & 0 & 0 & \boldsymbol{I}_{n_i} & 0 \\
0 & \cdots & \boldsymbol{D} & & -1 & & & & \\
\hline
\multicolumn{3}{c|}{0} & 0 & 1 & 0 & 0 & 0 & 1
\end{array}\right] \tag{9}$$

$$\mathcal{B}_i = \left[\begin{array}{cccc|c|c|c|c} \boldsymbol{B}^{i\prime}_1 & \boldsymbol{B}^{i\prime}_2 & \cdots & \boldsymbol{B}^{i\prime}_{n_i} & 0 & \boldsymbol{1} & 0 & L_i \end{array}\right]' \tag{10}$$

$$\mathcal{C}_i = \left[\begin{array}{c|c|c|c|c|c|c} 0 & r_i\beta_c\boldsymbol{C} & r_i\beta_d & 0 & 0 & 0 & 0 \end{array}\right] \tag{11}$$

$$\mathcal{D}_i = \left[\begin{array}{c|c|c|c|c|c|c} 0 & r_i\boldsymbol{I}_K & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{12}$$

we can re-write the multicast optimization problem as:

MINIMIZE: $\sum_{i=1}^{T} \mathcal{C}_i\mathcal{X}_i$
WITH RESPECT TO: $\mathcal{X}_i$
SUBJECT TO:

$$\begin{aligned} \mathcal{A}_i\mathcal{X}_i &= \mathcal{B}_i \\ \sum_{i=1}^{T} \mathcal{D}_i\mathcal{X}_i + \boldsymbol{S} &= \boldsymbol{V} \end{aligned}$$

which is the same formulation as the one presented in equations (19), (20) and (21) in the Appendix[2], with $\boldsymbol{E} = 0$ and $\boldsymbol{F} = 0$. The condition $0 \leq \mathcal{X}_i \leq \boldsymbol{U}_i$ is always satisfied because $\boldsymbol{X}^i$, $\boldsymbol{Y}^i$ and $\boldsymbol{Z}^i$ are binary, and $\boldsymbol{M}_i$ and $\boldsymbol{S}_a$ are limited by the maximum delay in the network. Each of the resulting subproblems corresponds exactly to the problem of routing a single multicast request over an empty network, with arbitrary link labels.

5.1.2 Decomposing the Single-Multicast Subproblems

The subproblem to be solved in the first decomposition is:
MAXIMIZE:

$$(\boldsymbol{\omega}\mathcal{D}_i - \mathcal{C}_i)\mathcal{X}_i + \alpha_i \tag{13}$$

WITH RESPECT TO: $\boldsymbol{\omega}$
SUBJECT TO:

$$\mathcal{A}_i\mathcal{X}_i = \mathcal{B}_i \tag{14}$$

Introducing (8), (11) and (12) in the objective function (13), it becomes:

$$(\boldsymbol{\omega}\mathcal{D}_i - \mathcal{C}_i)\mathcal{X}_i = r_i\left[(\boldsymbol{\omega} - \beta_c\boldsymbol{C})\boldsymbol{Y}^i - \beta_d\boldsymbol{M}\right] \tag{15}$$

We can now rewrite the problem defined by equations (13), (14) and (15) into the format presented in the Appendix; most of the matrices involved can be identified by inspection from equations (8) to (12):

$$\overline{\mathcal{X}}_j = \boldsymbol{X}^i_j \quad \overline{\mathcal{A}}_j = \boldsymbol{A} \quad \overline{\mathcal{B}}_j = \boldsymbol{B}^i_j \quad \overline{\mathcal{C}}_j = 0$$

$$\overline{\mathcal{Y}} = \left[\frac{\boldsymbol{Y}^i}{M_i}\right] \quad \overline{\mathcal{E}} = \left[\begin{array}{c|c} -r_i(\boldsymbol{\omega} - \beta_c\boldsymbol{C}) & r_i\beta_d \end{array}\right]$$

[2] This corresponds to the traditional decomposition applied to multicommodity flow problems.

$$\overline{\mathcal{V}} = \begin{bmatrix} 0_{n_iK\times 1} \\ \hline \mathbf{1}_{K\times 1} \\ \hline 0_{n_i\times 1} \\ \hline L_i \end{bmatrix}$$

$$\overline{\mathcal{D}}_j = \left[\, 0 \cdots \boldsymbol{I}_K \cdots 0 \middle| 0 \middle| 0 \cdots D_i' \cdots 0 \middle| 0 \,\right]'$$

$$\overline{\mathcal{F}} = \left[\begin{array}{c|c} -\boldsymbol{I}_K & \\ \vdots & 0 \\ -\boldsymbol{I}_K & \\ \hline \boldsymbol{I}_K & 0 \\ \hline & -1 \\ & -1 \\ 0 & \vdots \\ & -1 \\ \hline 0 & 1 \end{array}\right] \quad \overline{\mathcal{S}} = \begin{bmatrix} \boldsymbol{Z}_1^i \\ \vdots \\ \boldsymbol{Z}_{n_i}^i \\ \boldsymbol{S}_Y \\ \boldsymbol{S}_a^i \\ S_{Li} \end{bmatrix}$$

The problem can then be expressed as follows:
MINIMIZE:

$$\sum_{j=1}^{n_i} \overline{\mathcal{C}}_j \overline{\mathcal{X}}_j + \overline{\mathcal{E}\mathcal{Y}} = \overline{\mathcal{E}\mathcal{Y}} \tag{16}$$

WITH RESPECT TO: $\overline{\mathcal{X}}_j, \overline{\mathcal{Y}}$
SUBJECT TO:

$$\overline{\mathcal{A}}_j \overline{\mathcal{X}}_j = \overline{\mathcal{B}}_j \quad j = 1, \cdots, n_i \tag{17}$$

$$\sum_{j=1}^{n_i} \overline{\mathcal{D}}_j \overline{\mathcal{X}}_j + \overline{\mathcal{F}\mathcal{Y}} + \overline{\mathcal{S}} = \overline{\mathcal{V}} \tag{18}$$

whose decomposition corresponds to a set of unicast routing problems, which can be efficiently solved by methods such as network simplex.

Figure 1 illustrates the complete two-step decomposition of the optimum multicast routing problem.

5.2 Solution to the Integer Programming Problem

In this section, we describe the additions to the value-fixing phase of the branch-and-bound method to further prune the search space, thus making it run faster.

The value-fixing phase consists of setting binary variables either to 0 or to 1. We can use our knowledge of the structure of the problem to reduce the number of free variables. The variables with integer constraints are $\boldsymbol{X}$ and $\boldsymbol{Y}$; however, if $\boldsymbol{X}$ is integer, the constraint set and the objective function will automatically ensure that $\boldsymbol{Y}$ is also integer; therefore, we can drop the

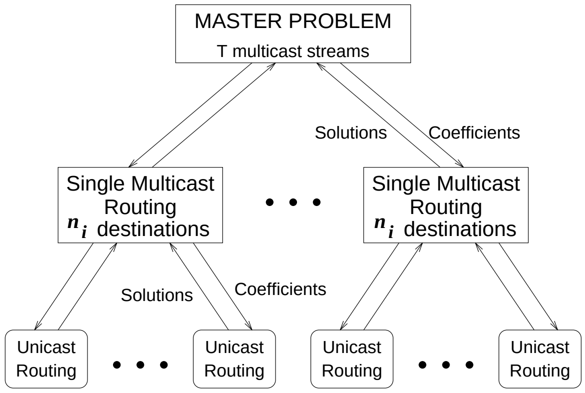

Figure 1: The two-step decomposition process

integer constraints on $\boldsymbol{Y}$. The following rules can be applied to reduce the number of free variables:

At initialization: Since the links directed to the source of a multicast will never carry any flow for that multicast, one can set:

$$X_{jk}^i = 0$$

$$i = 1, \ldots, T; \quad k = 1, \ldots, n_i; \quad j : A_{s_ij} = -1$$

Links originating from multicast destinations cannot be in the path to that destination:

$$X_{jk}^i = 0 \qquad i = 1, \ldots, T; \quad j, k : A_{d_{ik}j} = 1$$

When X_{kj}^i is set to 0: This means that request i, on its way to its k^{th} destination, will not use link j. Denoting by $\mathcal{N}$ the node from which link j originates, and assuming that l links originate from $\mathcal{N}$, if $\mathcal{N}$ is the origin of request i (or if one of the incoming links to $\mathcal{N}$ has been set to 1 as a result of previous value-fixing) and if the X_{kj}^i corresponding to $l-2$ branches out of the remaining $l-1$ branches have been fixed to 0, then the last remaining X_{kj}^i can be set to 1.

When X_{kj}^i is set to 1: This means that request i, on its way to its k^{th} destination, will use link j. Denoting by $\mathcal{N}$ the node from which link j originates, and assuming that l links originate from $\mathcal{N}$, the X_{kj}^i corresponding to the remaining $l-1$ links can be all set to 0.

Capacity Constraints: At any given step in the value-fixing process, there are some Y_j^i that have been set to 1. This means that a certain amount of bandwidth (given by r_i) has been reserved on that link already. Therefore, streams requiring more than the free bandwidth on that link should not use it. Formally:

$$\text{Define:}\quad U_j = V_j - \sum_{i:Y_j^i \text{ is set to } 1} r_i \quad j = 1, \ldots, K$$

$$\text{If } r_i > U_i, \text{ set } X_{jk}^i = 0$$

$$k = 1, \ldots, n_i, \quad i : Y_j^i \text{ is not set}$$

Note that these rules can be applied recursively, i.e., if setting a variable to 1 implies setting another variable to 0 (or vice-versa), the corresponding rule can also be invoked.

6 Numerical Evaluation

In this section, we give a numerical evaluation of the average run time of the algorithm presented in this paper. The evaluation consisted of routing a single multicast session in an empty network, and was carried out in three phases. In the first phase, we evaluated the improvement in run time gained when decomposition is used, compared with the traditional simplex method, for the linear relaxation of the routing problem. In the second phase, we evaluated the improvement in run time gained by the improved value-fixing rules of section 5.2. Finally, in the third phase, we compared the average run time for the optimum multicast routing algorithm presented in this paper and the shortest-path and KMB algorithms, used as described in section 3. When routing multiple-multicast sessions, the KMB and shortest-path algorithms were applied in a random order to each of the multicasts. The algorithms were implemented in a DEC 5000/240 workstation in C.

6.1 Evaluation Scenarios

For the network scenario, we chose the following: (i) *topology* - a simplified version of the NSFNet T3 backbone, with 12 nodes and 15 full-duplex links, depicted in figure 2; we also considered topologies with 6 nodes and 8 full-duplex links (half of the size of the NSFNet), and 6 nodes and 15 full-duplex links (a fully-connected topology), generated at random; (ii) *link costs* set to 1; (iii) *link delays* set to the propagation delays (NSFNet) or generated at random; (iv) *link capacities* are all equal.

For the traffic scenario, we chose the following: sessions with 1, 2 or 4 multicasts. For single-multicast sessions, the bandwidth requirement is irrelevant as

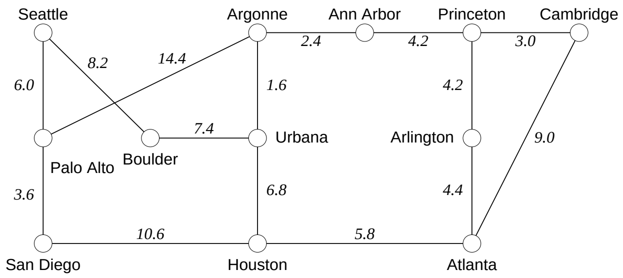

Figure 2: The simplified NSFNet T3 backbone (numbers next to the links indicate the delays).

long as it is less than the link bandwidth. For 2-multicast sessions, we set the bandwidth requirement to 60%; this value was chosen to make the capacity constraints active (if the bandwidth requirement were 50% or less, the problem could be divided into two decoupled single-multicast sessions). For 4-multicast sessions, we set the bandwidth requirement to 30%. No latency constraint was imposed.

We considered the following objective functions for the optimum multicast routing algorithm:

- minimum cost ($\beta_c = 1$, $\beta_d = 0$ in equation (1)); this solution will be referred to as "Cost" in the discussion;
- minimum cost, with delay as a secondary objective ($\beta_c \gg \beta_d > 0$); this solution will be referred to as "Cost/delay"; and
- minimum delay, with cost as a secondary objective ($\beta_d \gg \beta_c > 0$); this will be referred to as "Delay/cost".

6.2 Evaluation of the Decomposition Procedure

The run time for the linear relaxation of the routing problem is a function of the following parameters: (i) number of links in the topology; (ii) number of multicasts in the session; and (ii) number of destinations in each multicast.

The general conclusion is that the decomposition significantly reduces the time needed to compute the optimum. The difference in performance is a function of the size of the problem and the objective function, going all the way from *increase* in run time by a factor of 2 (6 nodes, 15 links, single multicast, minimum cost) when using decomposition, to an improvement by a factor of 100 or more; for example, to compute the routes for a 4-multicast session in the NSFNet, each multicast having 5 destinations, the traditional simplex method took an average of 542 seconds, while using decomposition the time was reduced to 4.8 seconds. The bigger the problem, the larger the advantage. Decomposition will increase the run time only on very small problems, with single-multicast sessions, where the overhead associated with it is not compensated by a decrease in run time.

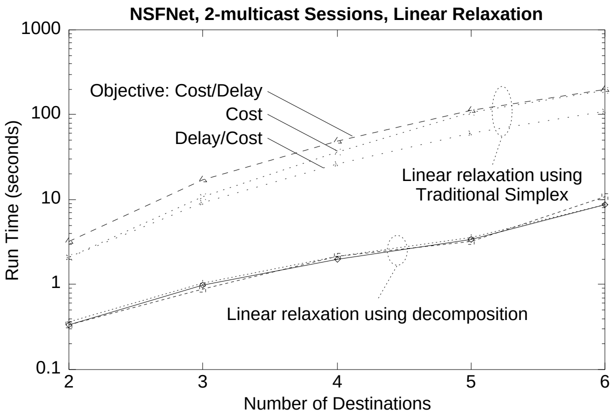

Figure 3: Observed run times for the NSFNet topology, 2-multicast sessions, linear relaxation only

In figure 3, we plot the run time as a function of the number of destinations in the multicast, for 2-multicast sessions in the NSFNet, for the various objective functions. For this scenario, an improvement of 10 times is observed. Figure 3 also shows that the run time for the linear relaxation is largely independent of the objective function when using decomposition.

6.3 Evaluation of the Pruning Procedure

The main result is that, in general, the impact of the value-fixing (pruning) rules is much less dramatic than the effect of decomposition. Figure 4 shows the run times for 2-multicast sessions in the NSFNet, as a function of the number of destinations. The figure shows that, while there is little advantage if the objective is to minimize cost, there is a large advantage (2 to 4 times) if the objective is to minimize delay, with cost as a secondary objective. In summary, the more complex the objective function is, the larger the advantage.

6.4 The Heuristic Algorithms

We have found that, as expected, the heuristic algorithms run much faster than the optimum presented in this paper. The run times for the KMB heuristic and

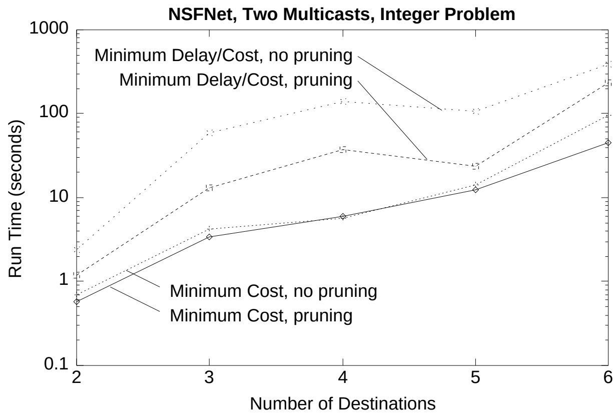

Figure 4: Observed run times for the NSFNet topology, 2-multicast sessions, integer problem

the shortest-path algorithm are shown in figure 5, for single-multicast sessions over the NSFNet; the heuristics are one to two orders of magnitude faster than the optimum.

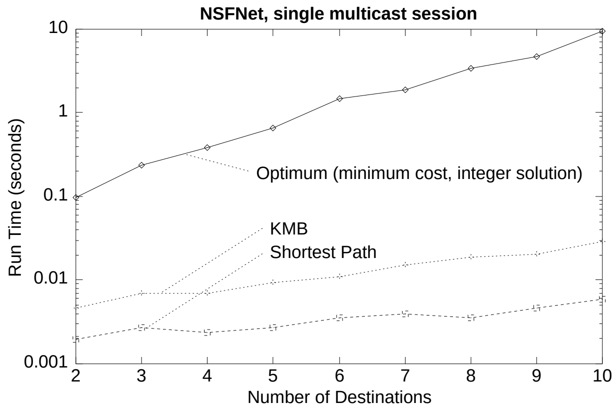

Figure 5: Observed run times for the NSFNet topology, single-multicast sessions

However, a problem with the heuristic solutions is that they might not find a route for multiple-multicast sessions, even though one exists. In contrast, if there a solution to the routing problem, the optimum will always find it. Table 1 shows the fraction of successful routes for 2-multicast sessions, routed through a 6-node, 8-link network, for the different algorithms. In all cases, there was at least one solution, and the optimum algorithm always found it; the heuristics not always were able to find a route. For example, for 5-destination multicasts, the shortest path failed in 50% of the cases, and the KMB failed in 40%. However, it should be stressed that these figures are typical of scenarios where the stream bandwidth requirements are a large fraction of the link bandwidth; if they are not, then the fraction of successful routes for the heuristic algorithms is close to the optimum.

Table 1: Fraction of successful routes for a 6-node, 8-link network under 2-multicast sessions

Dest.	KMB	Short. Path	Optimum
2	0.95	0.95	1.0
3	0.90	0.80	
4	0.75	0.70	
5	0.60	0.50	

7 Conclusions

In this paper, we showed that the optimum multicast routing problem for multimedia streams can be formulated as an integer programming problem, and proposed an efficient solution technique. We have shown that the proposed techniques vastly reduce the run times, when compared with traditional methods. We also compared the run time of the optimum solution with some heuristic algorithms. The run time for the heuristics is still one to two orders of magnitude less than that for the optimum, but they might fail to find a solution even if one exists, when routing multiple-multicast sessions. In a forthcoming paper, we will present a complete evaluation of the optimum multicast routing algorithm under realistic conditions, and compare its performance to the heuristic algorithms.

For large networks, the optimum solution presented here is not practical; its main use is as benchmark for other multicast routing algorithms.

Appendix

In this appendix we show an extension of the decomposition problem shown in [9], used in the solution of the linear relaxation of the optimum multicast routing problem. Consider the following optimization problem:

GIVEN: $\boldsymbol{A}_i, \boldsymbol{b}_i, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{V}, \boldsymbol{U}_i$
MINIMIZE:

$$\sum_{i=1}^{T} \boldsymbol{C}_i \boldsymbol{x}_i + \boldsymbol{E}\boldsymbol{y} \tag{19}$$

WITH RESPECT TO: $\boldsymbol{x}_i, \boldsymbol{y}, \boldsymbol{s}$
SUBJECT TO:

$$\boldsymbol{A}_i \boldsymbol{x}_i = \boldsymbol{b}_i, \quad i = 1, \ldots, T \tag{20}$$

$$\sum_{i=1}^{T} \boldsymbol{D}_i \boldsymbol{x}_i + \boldsymbol{F}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{V} \tag{21}$$

$$0 \leq \boldsymbol{x}_i \leq \boldsymbol{U}_i$$

$$\boldsymbol{y} \geq 0$$

$$\boldsymbol{s} \geq 0$$

Where $\boldsymbol{A}_i$ is $N \times K$, $\boldsymbol{b}_i$ is $N \times 1$, $\boldsymbol{C}_i$ is $1 \times K$, $\boldsymbol{D}_i$ is $M \times K$, $\boldsymbol{E}$ is $1 \times L$, $\boldsymbol{F}$ is $N \times L$, $\boldsymbol{y}$ is $L \times 1$, $\boldsymbol{x}_i$ is $K \times 1$, and $\boldsymbol{s}, \boldsymbol{V}$ are $M \times 1$. Additionally, we assume that all elements in vector $\boldsymbol{V}$ are non-negative.

Define $\boldsymbol{X}_i = \{\boldsymbol{x}_i : \boldsymbol{A}_i \boldsymbol{x}_i = \boldsymbol{b}_i, \quad 0 \leq \boldsymbol{x}_i \leq \boldsymbol{U}_i\}$. Since $\boldsymbol{x}_i$ is limited, the set $\boldsymbol{X}_i$ will have a finite number k_i of extreme points. Therefore, any $\boldsymbol{x}_i \in \boldsymbol{X}_i$ can be written as a convex combination of these extreme points, as follows:

$$\boldsymbol{x}_i = \sum_{j=1}^{k_i} \lambda_{ij} \boldsymbol{x}_{ij} \tag{22}$$

$$\lambda_{ij} \geq 0 \tag{23}$$

$$\sum_{j=1}^{k_i} \lambda_{ij} = 1 \tag{24}$$

The $\boldsymbol{x}_{ij}$ in equation (22) are the extreme points of $\boldsymbol{X}_i$.

Introducing equations (22), (23) and (24) into equations (19) and (21), the optimization problem becomes:
GIVEN: $\boldsymbol{x}_{ij}, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{V}$
MINIMIZE:

$$\sum_{i=1}^{T} \sum_{j=1}^{k_i} \boldsymbol{C}_i (\lambda_{ij} \boldsymbol{x}_{ij}) + \boldsymbol{E}\boldsymbol{y} \tag{25}$$

WITH RESPECT TO: λ_{ij}, $\boldsymbol{y}, \boldsymbol{s}$
SUBJECT TO:

$$\sum_{i=1}^{T} \sum_{j=1}^{k_i} \boldsymbol{D}_i (\lambda_{ij} \boldsymbol{x}_{ij}) + \boldsymbol{F}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{V} \tag{26}$$

$$\sum_{j=1}^{k_i} \lambda_{ij} = 1, \quad i = 1, \ldots, T \tag{27}$$

$$\lambda_{ij} \geq 0 \qquad \boldsymbol{y} \geq 0 \qquad \boldsymbol{s} \geq 0$$

Equations (25), (26) and (27) can be written in matrix format as:

MINIMIZE:

$$\begin{bmatrix} \boldsymbol{C}_i \boldsymbol{x}_{ij} & \boldsymbol{E} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{y} \\ \boldsymbol{s} \end{bmatrix} \tag{28}$$

SUBJECT TO:

$$\begin{bmatrix} \overline{\boldsymbol{A}} & \boldsymbol{F} & \boldsymbol{I}_M \\ \overline{\boldsymbol{D}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{y} \\ \boldsymbol{s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{1} \end{bmatrix} \tag{29}$$

where:

$\overline{\boldsymbol{A}} = [\boldsymbol{D}_i \boldsymbol{x}_{ij}]$

$$\overline{\boldsymbol{D}} = \begin{bmatrix} 1 \cdots 1 & 0 & \cdots & 0 \\ 0 & 1 \cdots 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \underbrace{0}_{k_1} & \underbrace{0}_{k_2} & \cdots & \underbrace{1 \cdots 1}_{k_T} \end{bmatrix}$$

Assume that a basic feasible solution in terms of the λ_{ij}'s and $\boldsymbol{y}, \boldsymbol{s}$ is available, and let $[\ \boldsymbol{\omega} \quad \boldsymbol{\alpha}\ ]$ be the vector of dual variables corresponding to this basic solution. To determine the entering variable, one has to compute the following vector:

$$\begin{bmatrix} \boldsymbol{\omega} & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{A}} & \boldsymbol{F} & \boldsymbol{I}_M \\ \overline{\boldsymbol{D}} & 0 & 0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{C}_i \boldsymbol{x}_{ij} & \boldsymbol{E} & 0 \end{bmatrix} \tag{30}$$

The positive elements in the vector defined by equation (30) correspond to variables that can enter the basis and improve the objective function. Substituting the values for $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{D}}$ in equation (30), we find:

$$\begin{bmatrix} (\boldsymbol{\omega D}_i - \boldsymbol{C}_i)\boldsymbol{x}_{ij} + \alpha_i & \boldsymbol{\omega F} - \boldsymbol{E} & \boldsymbol{\omega} \end{bmatrix} \tag{31}$$

From equation (31) we can see that:

1. If $(\boldsymbol{\omega D}_i - \boldsymbol{C}_i)\boldsymbol{x}_{ij} + \alpha_i > 0$, then λ_{ij} can enter the basis;
2. If $(\boldsymbol{\omega F} - \boldsymbol{E})_k > 0$, then y_k can enter the basis;
3. If $\omega_l > 0$, then s_l can enter the basis.

Given $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$, conditions 2 and 3 above are immediate to compute. Since $\boldsymbol{x}_{ij}$ is an extreme point of $\boldsymbol{X}_i$, condition 1 above can be rewritten as:

a. Solve the following subproblem:
Maximize $(\boldsymbol{\omega D}_i - \boldsymbol{C}_i)\boldsymbol{x}_i + \alpha_i$
Subject to $\boldsymbol{A}_i \boldsymbol{x}_i = \boldsymbol{b}_i, \quad 0 \le \boldsymbol{x}_i \le \boldsymbol{U}_i$

b. If the objective value of the problem solved in (a.) is positive, then the λ_{ij} corresponding to the optimum $\boldsymbol{x}_i$ in that problem can enter the basis. Otherwise, no λ_{ij} from subproblem i can enter the basis.

Therefore, the optimization problem defined by equations (19), (20) and (21) can be solved iteratively solving the T subproblems in (a.), with coefficients ω given by the master problem. The final solution will be a linear combination of the subproblem solutions whose λ_{ij} remain in the basis at optimality.

References

[1] Bertsekas, D., and Gallager, R., *Data Networks*, Prentice-Hall, New Jersey, 1987.

[2] Deering, S.E., and Cheriton, D.R., "Multicast Routing in datagram internetworks and extended LANs," *ACM Trans. on Computer Systems*, May 1990, vol.8, no.2, p. 85-110.

[3] Moy, J., "Multicast Extensions to OSPF," *IETF Internet Draft*, July 1993.

[4] Karp, R.M., "Reducibility among combinatorial problems", in *Complexity of Computer Communications*, R.E. Miller and J.W. Thatcher (editors), pp.85-103, Plenum Press, New York 1972.

[5] Hwang, F.K., and Richards, D.S., "Steiner Tree Problems," *Networks*, Jan. 1992, vol.22, no.1, p. 55-89.

[6] L. Kou, G. Markowsky and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica* 15, pp 141-5, 1981.

[7] V.P. Kompella et al, "Multicast Routing for Multimedia Communication," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, June 1993, pp 286-92.

[8] F.S. Hillier and G.J. Lieberman, *Introduction to Operations Research*, 5th edition, New York, McGraw-Hill, 1990.

[9] M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, *Linear Programming and Network Flows, 2nd ed.*, John Wiley & Sons, New York, 1990.