Achieving Minimum-Cost Multicast: A Decentralized Approach Based on Network Coding

Desmond S. Lun*, Niranjan Ratnakar†, Ralf Koetter†,
Muriel Médard*, Ebad Ahmed*, and Hyunjoo Lee*
*Laboratory for Information and Decision Systems
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
E-mail: {dslun, medard, ebad, jennylee}@mit.edu
†Coordinated Science Laboratory
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
E-mail: {ratnakar, koetter}@uiuc.edu

Abstract— **We present decentralized algorithms that compute minimum-cost subgraphs for establishing multicast connections in networks that use coding. These algorithms, coupled with existing decentralized schemes for constructing network codes, constitute a fully decentralized approach for achieving minimum-cost multicast. Our approach is in sharp contrast to the prevailing approach based on approximation algorithms for the directed Steiner tree problem, which is suboptimal and generally assumes centralized computation with full network knowledge. We also give extensions beyond the basic problem of fixed-rate multicast in networks with directed point-to-point links, and consider the problem of minimum-energy multicast in wireless networks as well as the case of a concave utility function at the sender.**

I. Introduction

In networks where each node's functions are limited to the relaying or replication of received data, establishing minimum-cost multicast connections generally means finding the shortest tree connecting a set of points in a directed graph; in other words, solving the Steiner tree problem on directed graphs. But this is by no means a simple proposition, as the Steiner tree problem on directed graphs is NP-complete [1]. In light of the intractability of the problem, a number of approximation algorithms have been proposed, for example, [2], [3], [1]. There are, however, two main drawbacks to using these approximation algorithms: first, they are of course suboptimal; and, secondly, they assume centralized computation with full network knowledge.

The recent proposal of network coding [4], [5] has suggested relaxing the restrictions on the functions permitted for a node and to allow nodes to perform algebraic operations on received data, which completely subsumes relaying and replicating. When network coding is used, the problem of establishing minimum-cost multicast connections equates to two essentially decoupled problems [6]: one of determining the subgraph to code over (how much flow to put on each link), and the other of determining the code to use over that subgraph (the content of the flows).

For the latter problem, a variety of methods have been proposed, which include employing simple random linear coding at every node [7], [8], [9], [10]. Such random linear coding schemes are completely decentralized, requiring no co-ordination between nodes, and can operate under dynamic conditions [11].

In the present paper, we present optimal, decentralized methods for solving the former problem, which, when married with the random linear coding schemes, results in a fully decentralized approach for achieving minimum-cost multicast. More specifically, we describe two decentralized algorithms for computing minimum-cost subgraphs: one that applies for linear cost functions (which arise naturally when, for example, a monetary or energy cost must be paid for each link usage) and the other that applies for strictly convex cost functions (which arise naturally when, for example, the cost is latency or congestion). Our algorithms, being decentralized, can be operated in much the same manner as popular decentralized methods for point-to-point routing, such as distributed Bellman-Ford [12, Section 5.2].

We emphasize that our decentralized approach actually achieves minimum-cost multicast with network coding, which, because coding subsumes relaying and replicating, is guaranteed to be at least as good as minimum-cost multicast without network coding. Since the latter must generally be approximated, we expect the cost

Approach	Network	Average multicast cost			
		2 terminals	4 terminals	8 terminals	16 terminals
DST approximation	Telstra (au)	17.0	28.9	41.7	62.8
Network coding	Telstra (au)	13.5	21.5	32.8	48.0
DST approximation	Ebone (eu)	28.2	43.0	69.7	115.3
Network coding	Ebone (eu)	20.7	32.4	50.4	77.8
DST approximation	Exodus (us)	43.8	62.7	91.2	116.0
Network coding	Exodus (us)	33.4	49.1	68.0	92.9

TABLE I

AVERAGE COST OF RANDOM MULTICAST CONNECTIONS OF UNIT RATE FOR VARIOUS APPROACHES IN GRAPHS REPRESENTING VARIOUS ISP NETWORKS. THE COST PER UNIT FLOW ON EACH LINK IS THE LINK WEIGHT AS ASSESSED BY THE ROCKETFUEL PROJECT OF THE UNIVERSITY OF WASHINGTON [13].

improvement due to our decentralized approach to be quite significant. Indeed, we witnessed reductions in the average total weight of random multicast connections in various Internet Service Provider (ISP) networks ranging from 20% to 33% as a result of using our network-coding based approach rather than an approach based on the DST (Directed Steiner Tree) approximation algorithm described in [3] (see Table I).

In the following section, we review the optimization problem for minimum-cost subgraphs presented in [6]. We present decentralized algorithms that solve the optimization problem for linear cost functions and strictly convex cost functions in Sections III and IV, respectively. In Section V, we extend our algorithms to apply to the problem of minimum-energy multicast in wireless networks and the case of a concave utility function at the sender.

II. MINIMUM-COST MULTICAST WITH NETWORK CODING

Suppose that the directed graph $G = (N, A)$ models the network. Let $f_{ij}(\cdot)$ and c_{ij} be the cost function and the capacity, respectively, of link (i, j). We assume that $f_{ij}(\cdot)$ is convex and monotonically increasing and that c_{ij} is non-negative. Then the optimal cost for a rate R multicast connection from source node s to terminal nodes T that is asymptotically-achievable with network coding is given by the following optimization problem [6]:

$$\begin{aligned}
&\text{minimize} \sum_{(i,j)\in A} f_{ij}(z_{ij}) \\
&\text{subject to } z_{ij} \geq x_{ij}^{(t)}, \qquad \forall\ (i,j) \in A,\ t \in T, \\
&\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \\
&\qquad\qquad\qquad\qquad \forall\ i \in N,\ t \in T, \\
&c_{ij} \geq x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in A,\ t \in T,
\end{aligned} \tag{1}$$

where

$$\sigma_i^{(t)} = \begin{cases} R & \text{if } i = s, \\ -R & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Essentially, the constraints of problem (1) restrict z to being the link-capacities of a subgraph that admits minimum-flows of R from s to each t in T. So the fact that problem (1) finds the optimal cost for a rate R multicast follows almost directly from Theorem 1 of [4]. Note, however, that the solution is in general only asymptotically achievable using arbitrarily long codes.

If we wish to restrict the maximum link delay to be δ symbol periods and we can tolerate a rate that falls short of R by no more than $1/\delta$, then we actually need $z_{ij} \in \mathbb{Z}/\delta$ for all $(i, j) \in A$ (so that we can code over δ symbol periods). But, of course, placing such constraints into problem (1) would make it prohibitively difficult. The alternative is, given z, to take $\lceil \delta z \rceil/\delta$. We can guarantee that $\lceil \delta z \rceil/\delta$ satisfies the capacity constraints by substituting $z + 1/\delta$ in place of z into problem (1) since $\lceil \delta z \rceil/\delta < (\delta z + 1)/\delta = z + 1/\delta$, which results in the optimization problem

$$\begin{aligned}
&\text{minimize} \sum_{(i,j)\in A} f_{ij}(z_{ij} + 1/\delta) \\
&\text{subject to } z_{ij} \geq x_{ij}^{(t)}, \qquad \forall\ (i,j) \in A,\ t \in T, \\
&\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \\
&\qquad\qquad\qquad\qquad \forall\ i \in N,\ t \in T, \\
&c_{ij} - 1/\delta \geq x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in A,\ t \in T.
\end{aligned} \tag{3}$$

We notice that problem (3) has the same form as problem (1) and, therefore, any algorithm that solves problem (1) can be applied to problem (3). Hence, in the remainder of the paper, we focus only on problem (1).

We proceed to develop a decentralized algorithm to solve problem (1), commencing with the case of linear cost functions.

III. LINEAR COST FUNCTIONS

Let the cost function for link (i, j) be $f_{ij}(z_{ij}) = a_{ij}z_{ij}$, where a_{ij} is a non-negative real number. Towards the end of developing a decentralized algorithm to solve problem (1), we consider the Lagrangian dual

$$\begin{aligned} &\text{maximize } \sum_{t\in T} q^{(t)}(p^{(t)}) \\ &\text{subject to } \sum_{t\in T} p_{ij}^{(t)} = a_{ij}, \qquad \forall\ (i,j) \in A, \\ &\qquad\qquad p_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in A,\ t \in T, \end{aligned} \tag{4}$$

where

$$q^{(t)}(p^{(t)}) = \min_{x^{(t)}\in F^{(t)}} \sum_{(i,j)\in A} p_{ij}^{(t)} x_{ij}^{(t)}, \tag{5}$$

and $F^{(t)}$ is the bounded polyhedron of points $x^{(t)}$ satisfying the conservation of flow constraints

$$\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \quad \forall\ i \in N, \tag{6}$$

and capacity constraints

$$0 \leq x_{ij}^{(t)} \leq c_{ij}, \quad \forall\ (i,j) \in A. \tag{7}$$

Subproblem (5) is a standard linear minimum-cost flow problem, which can be solved using a multitude of different methods (see, for example, [14, Chapters 4–7] or [15, Chapters 9–11]); in particular, it can be solved in an asynchronous, distributed manner using the ε-relaxation method [16, Sections 5.3 and 6.5]. In addition, if the connection rate is negligible compared to the link capacities (that is, if $R \leq c_{ij}$ for all $(i, j) \in A$), then subproblem (5) reduces to a shortest path problem, which admits a simple, asynchronous, distributed solution [12, Section 5.2].

Now, to solve the dual problem (4), we employ subgradient optimization (see, for example, [17, Section I.2.4]). We start with an iterate $p[0]$ in the feasible set of (4) and, given an iterate $p[n]$ for some non-negative integer n, we solve subproblem (5) for each t in T to obtain $x[n]$. We then assign

$$p_{ij}[n+1] := \arg\min_{v\in P_{ij}} \sum_{t\in T} (v^{(t)} - (p_{ij}^{(t)}[n] + \theta[n]x_{ij}^{(t)}[n]))^2 \tag{8}$$

for each $(i, j) \in A$, where P_{ij} is the $|T|$-dimensional simplex

$$P_{ij} = \left\{ v \,\middle|\, \sum_{t\in T} v^{(t)} = a_{ij},\ v \geq 0 \right\} \tag{9}$$

and $\theta[n] > 0$ is an appropriate step size. Thus, $p_{ij}[n+1]$ is set to be the Euclidean projection of $p_{ij}[n]+\theta[n]x_{ij}[n]$ onto P_{ij}.

To perform the projection, we use the following algorithm, the justification of which we defer to Appendix I. Let $u := p_{ij}[n] + \theta[n]x_{ij}[n]$ and suppose we index the elements of T such that $u^{(t_1)} \geq u^{(t_2)} \geq \ldots \geq u^{(t_{|T|})}$. Take $\hat{k}$ to be the smallest k such that

$$\frac{1}{k}\left(a_{ij} - \sum_{r=1}^{t_k} u^{(r)}\right) \leq -u^{(t_{k+1})} \tag{10}$$

or set $\hat{k} = |T|$ if no such k exists. Then the projection is achieved by

$$p_{ij}^{(t)}[n+1] = \begin{cases} u^{(t)} + \frac{a_{ij} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}}{\hat{k}} & \text{if } t \in \{t_1, \ldots, t_{\hat{k}}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

The disadvantage of subgradient optimization is that, whilst it yields good approximations of the optimal value of the Lagrangian dual problem (4) after sufficient iteration, it does not necessarily yield a primal optimal solution. There are, however, methods for recovering primal solutions in subgradient optimization. We employ the following method, which is essentially due to Sherali and Choi [18].

Let $\{\mu_l[n]\}_{l=1,\ldots,n}$ be a sequence of convex combination weights for each non-negative integer n, i.e. $\sum_{l=1}^{n} \mu_l[n] = 1$ and $\mu_l[n] \geq 0$ for all $l = 1, \ldots, n$. Further, let us define

$$\gamma_{ln} := \frac{\mu_l[n]}{\theta[n]}, \quad l = 1, \ldots, n,\ n = 0, 1, \ldots, \tag{12}$$

and

$$\Delta\gamma_n^{\max} := \max_{l=2,\ldots,n} \{\gamma_{ln} - \gamma_{(l-1)n}\}. \tag{13}$$

If the step sizes $\{\theta[n]\}$ and convex combination weights $\{\mu_l[n]\}$ are chosen such that

1) $\gamma_{ln} \geq \gamma_{(l-1)n}$ for all $l = 2, \ldots, n$ and $n = 0, 1, \ldots$,
2) $\Delta\gamma_n^{\max} \to 0$ as $n \to \infty$, and
3) $\gamma_{1n} \to 0$ as $n \to \infty$ and $\gamma_{nn} \leq \delta$ for all $n = 0, 1, \ldots$ for some $\delta > 0$,

then we obtain an optimal solution to the primal problem (1) from any accumulation point of the sequence of primal iterates $\{\tilde{x}[n]\}$ given by

$$\tilde{x}[n] := \sum_{l=1}^{n} \mu_l[n]x[l], \quad n = 0, 1, \ldots. \tag{14}$$

We justify this primal recovery method in Appendix I.

The required conditions on the step sizes and convex combination weights are satisfied by the following choices [18, Corollaries 2–4]:

1) step sizes $\{\theta[n]\}$ such that $\theta[n] > 0$, $\lim_{n\to 0} \theta[n] = 0$, $\sum_{n=1}^{\infty} \theta_n = \infty$, and convex combination weights $\{\mu_l[n]\}$ given by $\mu_l[n] = \theta[l]/\sum_{k=1}^{n} \theta[k]$ for all $l = 1, \ldots, n$, $n = 0, 1, \ldots$;
2) step sizes $\{\theta[n]\}$ given by $\theta[n] = a/(b + cn)$ for all $n = 0, 1, \ldots,$, where $a > 0$, $b \geq 0$ and $c > 0$, and convex combination weights $\{\mu_l[n]\}$ given by $\mu_l[n] = 1/n$ for all $l = 1, \ldots, n$, $n = 0, 1, \ldots$; and
3) step sizes $\{\theta[n]\}$ given by $\theta[n] = n^{-\alpha}$ for all $n = 0, 1, \ldots,$, where $0 < \alpha < 1$, and convex combination weights $\{\mu_l[n]\}$ given by $\mu_l[n] = 1/n$ for all $l = 1, \ldots, n$, $n = 0, 1, \ldots$.

Moreover, for all three choices, we have $\mu_l[n+1]/\mu_l[n]$ independent of l for all n, so primal iterates can be computed iteratively using

$$\begin{aligned}\tilde{x}[n] &= \sum_{l=1}^{n} \mu_l[n]x[l] \\ &= \sum_{l=1}^{n-1} \mu_l[n]x[l] + \mu_n[n]x[n] \\ &= \phi[n-1]\tilde{x}[n-1] + \mu_n[n]x[n],\end{aligned} \tag{15}$$

where $\phi[n] := \mu_l[n+1]/\mu_l[n]$.

We now have a relatively simple algorithm for computing optimal feasible solutions to problem (1) in a decentralized manner, with computation taking place at each node, which needs only to be aware of the capacities and costs of its incoming and outgoing links. For example, for all links (i, j) in A, we can set $p_{ij}^{(t)}[0] = a_{ij}/|T|$ at both nodes i and j. Since each node has the capacities and costs of its incoming and outgoing links for subproblem (5) for each $t \in T$, we can apply the ε-relaxation method to obtain flows $x^{(t)}[0]$ for each $t \in T$, which we use to compute $p_{ij}[1]$ and $\tilde{x}_{ij}[0]$ at both nodes i and j using equations (8) and (14), respectively. We then re-apply the ε-relaxation method and so on.

Although the decentralized algorithm that we have just discussed could perhaps be extended to convex cost functions (by modifying the dual problem and employing the ε-relaxation method for convex cost network flow problems [19], [20]), a significantly more direct and natural method is possible, which we proceed to present.

IV. Strictly convex cost functions

In this section, we assume that $f_{ij}(z_{ij})$ is a strictly convex, monotone increasing function of z_{ij}. We can still consider linear cost, if we wish, by choosing $f_{ij}(z_{ij}) = (a_{ij}z_{ij})^{1+\alpha}$, which is a strictly convex function for $\alpha > 0$, and which approaches the linear cost function $a_{ij}z_{ij}$ as α approaches 0. Note, however, that the convergence time for the algorithm proposed in the sequel approaches ∞ as α approaches 0.

Note that $f_{ij}(z_{ij}) = f_{ij}(\max_{t\in T} x_{ij}^{(t)})$ is a convex function of $x_{ij}^{(t)}$ since a monotone increasing, convex function of a convex function is convex. Thus, it follows that problem (1) can be restated as the following convex problem.

$$\begin{aligned}&\text{minimize} \sum_{(i,j)\in A} f_{ij}(z_{ij}) \\ &\text{subject to } z_{ij} = \max_{t\in T} x_{ij}^{(t)}, \quad \forall\ (i,j) \in A, \\ &\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \\ &\qquad\qquad \forall\ i \in N,\ t \in T, \\ &x_{ij}^{(t)} \geq 0, \quad \forall\ (i,j) \in A,\ t \in T.\end{aligned} \tag{16}$$

Note that the capacity constraints have been removed removed, since they can be enforced by making edges arbitrarily costly as their flows approach their respective capacities.

Since the function $z_{ij} = \max_{t\in T} x_{ij}^{(t)}$ is not a strictly convex function, this convex problem might admit multiple solutions with the same minimum cost. Further, z_{ij} as a function of $\max_{t\in T} x_{ij}^{(t)}$ is not differentiable everywhere. These factors pose some problems for algorithm design. We address this problem by solving the following modification of problem (16).

$$\begin{aligned}&\text{minimize} \sum_{(i,j)\in A} f_{ij}(z'_{ij}) \\ &\text{subject to } z'_{ij} = \left(\sum_{t\in T}\left(x_{ij}^{(t)}\right)^n\right)^{\frac{1}{n}}, \quad \forall\ (i,j) \in A, \\ &\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \\ &\qquad\qquad \forall\ i \in N,\ t \in T, \\ &x_{ij}^{(t)} \geq 0, \quad \forall\ (i,j) \in A,\ t \in T.\end{aligned} \tag{17}$$

Since the function $z'_{ij} = (\sum_{t \in T} (x_{ij}^{(t)})^n)^{\frac{1}{n}}$ is a strictly convex function, it follows that the above problem is a strictly convex function and admits a unique solution for any integer $n > 0$. For $n > 0$, $z'_{ij} \geq z_{ij}$ and, as n approaches ∞, z'_{ij} approaches z_{ij}. In the sequel, we assume that n is large and we solve the problem in (17) using a decentralized algorithm. (Since $z'_{ij} \geq z_{ij}$, a network code with a usage of z'_{ij} on each edge (i,j) exists for every n. This implies that we do not need to consider n as ∞ for the results to hold. However, since we desire z'_{ij} to be close to z_{ij}, we need to pick a large value for n.)

Let $U(x) := -\sum_{(i,j)\in A} f_{ij}((\sum_{t\in T}(x_{ij}^{(t)})^n)^{\frac{1}{n}})$. Then problem (17) can be restated as follows:

$$
\begin{aligned}
&\text{maximize } U(x) \\
&\text{subject to} \\
&\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \quad \forall\, i \in N,\, t \in T, \qquad (18)\\
&x_{ij}^{(t)} \geq 0, \qquad \forall\, (i,j) \in A,\, t \in T.
\end{aligned}
$$

Lemma 1: The function $U(x)$ is a strictly concave function of x.

Proof: The lemma follows because $f_{ij}(\cdot)$ is a monotone increasing, strictly convex function and z'_{ij} is a strictly convex function of x_{ij}. ■

The Lagrangian for problem (18) is as follows:

$$
\begin{aligned}
L(x,p,\lambda) = U(x) & - \sum_{t\in T}\left\{ \sum_{i\in N} p_i^{(t)} \left(\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} \right.\right. \\
& \left.\left. - \sigma_i^{(t)} \right) + \sum_{(i,j)\in A} \lambda_{ij}^{(t)} x_{ij}^{(t)} \right\}. \qquad (19)
\end{aligned}
$$

Since $U(x)$ is a strictly concave function of x (by Lemma 1), there exist a unique minimizing solution for problem (18), say $\hat{x}$, and Lagrange multipliers, say $\hat{p}$ and $\hat{\lambda}$, which satisfy the following Karush-Kuhn-Tucker conditions.

$$
\frac{\partial L(\hat{x},\hat{p},\hat{\lambda})}{\partial x_{ij}^{(t)}} = \left(\frac{\partial U(\hat{x})}{\partial x_{ij}^{(t)}} - \left(\hat{p}_i^{(t)} - \hat{p}_j^{(t)} \right) + \hat{\lambda}_{ij}^{(t)} \right) = 0, \quad \forall\, (i,j) \in A,\, t \in T, \qquad (20)
$$

$$
\sum_{\{j|(i,j)\in A\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} \hat{x}_{ji}^{(t)} = \sigma_i^{(t)}, \quad \forall\, i \in N,\, t \in T, \qquad (21)
$$

$$
\hat{x}_{ij}^{(t)} \geq 0 \qquad \forall\, (i,j) \in A,\, t \in T, \qquad (22)
$$

$$
\hat{\lambda}_{ij}^{(t)} \geq 0 \qquad \forall\, (i,j) \in A,\, t \in T, \qquad (23)
$$

$$
\hat{\lambda}_{ij}^{(t)} \hat{x}_{ij} = 0 \qquad \forall\, (i,j) \in A,\, t \in T. \qquad (24)
$$

Let $(y)_x^+$ for $x \geq 0$ denote the following function of y:

$$
(y)_x^+ = \begin{cases} y & \text{if } x > 0, \\ \max\{y, 0\} & \text{if } x \leq 0. \end{cases} \qquad (25)
$$

To solve problem (18) in a decentralized fashion, consider the following primal-dual algorithm [25]:[1]

$$
\dot{x}_{ij}^{(t)} = k_{ij}^{(t)}(x_{ij}^{t}) \left(\frac{\partial U(x)}{\partial x_{ij}^{(t)}} - q_{ij}^{(t)} + \lambda_{ij}^{(t)} \right), \qquad (26)
$$

$$
\dot{p}_i^{(t)} = h_i^{(t)}(p_i^{(t)})(y_i^{(t)} - \sigma_i^{(t)}), \qquad (27)
$$

$$
\dot{\lambda}_{ij}^{(t)} = m_{ij}^{(t)}(\lambda_{ij}^{(t)}) \left(-x_{ij}^{(t)} \right)^+_{\lambda_{ij}^{(t)}}, \qquad (28)
$$

where

$$
q_{ij}^{(t)} := p_i^{(t)} - p_j^{(t)}, \qquad (29)
$$

$$
y_i^{(t)} := \sum_{\{j|(i,j)\in A\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} \hat{x}_{ji}^{(t)}, \qquad (30)
$$

and $k_{ij}^{(t)}(x_{ij}^{(t)}) > 0$, $h_i^{(t)}(p_i^{(t)}) > 0$, and $m_{ij}^{(t)}(\lambda_{ij}^{(t)}) > 0$ are non-decreasing continuous functions of $x_{ij}^{(t)}$, $p_i^{(t)}$, and $\lambda_{ij}^{(t)}$ respectively.

Theorem 1: The algorithm specified by Equations (26)–(28) is globally, asymptotically stable.

Proof: See Appendix II. ■

The global, asymptotic stability of the algorithm implies that no matter what the initial choice of (x,p) is, the primal-dual algorithm will converge to the unique solution of problem (18). We have to choose λ, however, with non-negative entries as the initial choice.

We associate a processor with each edge (i,j) and node i. In a typical setting where there is one processor at every node, we could assign the processor at a node to be its own processor as well as the processor for all its outgoing edges.

[1]A primal-dual algorithm is a method for solving convex optimization problems by computing the primal and the dual variables simultaneously and moving towards the saddle point at each step, as opposed to a pure dual approach, where the optimal dual variables are computed and the primal is later recovered.

We assume that the processor for node i keeps track of the variables $\{p_i^{(t)}\}_{t\in T}$, while the processor for edge (i,j) keeps track of the variables $\{\lambda_{ij}^{(t)}\}_{t\in T}$ and $\{x_{ij}^{(t)}\}_{t\in T}$. Note that for the actual construction of a network code, we need z'. With this assumption, the algorithm is decentralized in the following sense:

- a node processor needs only to exchange information with the processors for edges coming in or out of the node; and
- an edge processor needs only to exchange information with the processors for nodes that it is connected to.

By noting that $\partial U(x)/\partial x_{ij}^{(t)} = -f_{ij}(z'_{ij}) \left(x_{ij}^{(t)}/z'_{ij}\right)^{n-1}$, we see from equations (26)–(28) that this is indeed the case.

V. Extensions

We now consider some extensions of the model described in Section II.

A. Minimum-energy multicast in wireless networks

In wireless networks, the cost that we are often interested in minimizing is the energy usage, since we expect nodes to operate on limited power supplies. But computing the energy cost in wireless networks is complicated by the omnidirectionality of the antennas; when transmitting from node i to node j, we get transmission to all nodes whose distance from i is less than that from i to j "for free" — a phenomenon referred to as the "wireless multicast advantage" in [21]. Under this phenomenon, even the problem of minimum-energy broadcast in wireless networks without network coding is NP-complete [22], [23].

Suppose that a_{ij} is the energy required to send a unit flow from node i to node j. Then, for the problem of minimum-energy multicast in wireless networks, the relevant optimization problem that needs to be solved is the following [6]:

$$\begin{aligned}
&\text{minimize} \quad \sum_{(i,j)\in A} a_{ij} z_{ij} \\
&\text{subject to} \quad \sum_{\{k|(i,k)\in A,(i,k)\succeq(i,j)\}} (z_{ik} - x_{ik}^{(t)}) \geq 0, \\
&\qquad\qquad\qquad\qquad \forall\ (i,j) \in A',\ t \in T, \\
&\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \\
&\qquad\qquad\qquad\qquad \forall\ i \in N,\ t \in T, \\
&x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in A,\ t \in T,
\end{aligned} \tag{31}$$

where A' is a subset of A with the property that the constraint

$$\sum_{\{k|(i,k)\in A,(i,k)\succeq(i,j)\}} (z_{ik} - x_{ik}^{(t)}) \geq 0 \tag{32}$$

is unique for all $(i,j) \in A'$ (for example, if (i,j_1) and (i,j_2) are unique members of A such that $a_{ij_1} = a_{ij_2}$, then only one of the two is in A').

The linear optimization problem (31) can also be solved in a distributed manner much the same as that for problem (1). The relevant Lagrangian dual problem is

$$\begin{aligned}
&\text{maximize} \quad \sum_{t\in T} q^{(t)}(p^{(t)}) \\
&\text{subject to} \quad \sum_{t\in T} \sum_{\{k|(i,k)\in A',(i,k)\preceq(i,j)\}} p_{ik}^{(t)} = a_{ij} \\
&\qquad\qquad\qquad\qquad \forall\ (i,j) \in A, \\
&p_{ij}^{(t)} \geq 0 \qquad \forall\ (i,j) \in A,\ t \in T,
\end{aligned} \tag{33}$$

where

$$q^{(t)}(p^{(t)}) = \min_{x^{(t)}\in F^{(t)}} \sum_{(i,j)\in A} \left(\sum_{\{k|(i,k)\in A',(i,k)\preceq(i,j)\}} p_{ik}^{(t)} \right) x_{ij}^{(t)}. \tag{34}$$

Note that problem (33) can be rewritten as

$$\begin{aligned}
&\text{maximize} \quad \sum_{t\in T} q^{(t)}(p^{(t)}) \\
&\text{subject to} \quad \sum_{t\in T} p_{ij}^{(t)} = a_{ij} - \sum_{\{k|(i,k)\in A',(i,k)\preceq(i,j)\}\setminus\{j\}} a_{ij} \\
&\qquad\qquad\qquad\qquad \forall\ (i,j) \in A', \\
&p_{ij}^{(t)} \geq 0 \qquad \forall\ (i,j) \in A,\ t \in T,
\end{aligned} \tag{35}$$

which has the same form as problem (4) and to which the same distributed solution method can be applied. One slight modification, however, is necessary: Rather than calculating the calculating the dual variable $p_{ij}[n]$ for each iteration at both nodes i and j, it is computed at node i only and the shortest path link cost $\sum_{\{k|(i,k)\in A',(i,k)\preceq(i,j)\}} p_{ik}[n]$ is sent to node j.

B. Concave utility functions

Frequently, rather than demanding fixed rate connections, users are willing to forego some rate in return for lower cost. Such a scenario is typically modeled by associating a utility function with each connection that has the same units as the cost function. The objective, then, is to maximize utility minus cost.

We modify the original problem by associating a utility function $U_r(R)$ with the source such that $U_r(R)$ is the utility derived by the source when R is the data rate. The function $U_r(R)$ is assumed to be a strictly concave, increasing function of R. Further, we drop some constraints in the original problem by making the observation that the equality constraints at a terminal t, namely

$$\sum_{\{j|(t,j)\in A\}} x_{tj}^{(t)} - \sum_{\{j|(j,t)\in A\}} x_{jt}^{(t)} = \sigma_t^{(t)} = -R, \quad (36)$$

follow from the constraints at the source and at the other nodes. The dropping of these constraints is crucial to the proof that the algorithm presented in the sequel is decentralized. Thus, in this setup, the problem we address is as follows:

$$\begin{aligned}
&\text{maximize } U(x,R)\\
&\text{subject to}\\
&\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)},\\
&\qquad\qquad\qquad\qquad \forall\ i \in N \setminus \{t\},\ t \in T,\\
&R \geq 0,\\
&x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in A,\ t \in T,
\end{aligned} \quad (37)$$

where $U(x,R) := U_r(R) - \sum_{(i,j)\in A} f_{ij}(z'_{ij})$.

This problem can be solved by the following primal-dual algorithm.

$$\dot{x}_{ij}^{(t)} = k_{ij}^{(t)}(x_{ij}^{(t)}) \left(\frac{\partial U(x,R)}{\partial x_{ij}^{(t)}} - q_{ij}^{(t)} + \lambda_{ij}^{(t)} \right), \quad (38)$$

$$\dot{R} = k_R(R) \left(\frac{\partial U(x,R)}{\partial R} - q_R + \lambda_R \right), \quad (39)$$

$$\dot{p}_i^{(t)} = h_i^{(t)}(p_i^{(t)}) y_i^{(t)}, \quad (40)$$

$$\dot{\lambda}_{ij}^{(t)} = m_{ij}^{(t)}(\lambda_{ij}^{(t)}) \left(-x_{ij}^{(t)} \right)^+_{\lambda_{ij}^{(t)}}, \quad (41)$$

$$\dot{\lambda}_R = m_R(\lambda_R) \left(-R \right)^+_{\lambda_R}, \quad (42)$$

where

$$q_{ij}^{(t)} := p_i^{(t)} - p_j^{(t)}, \quad (43)$$

$$q_R := -\sum_{t\in T} p_s^{(t)}, \quad (44)$$

$$y_i^{(t)} := \sum_{\{j|(i,j)\in A\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} \hat{x}_{ji}^{(t)} - \sigma_i^{(t)}. \quad (45)$$

It can be shown using similar arguments as those for Theorem 1 that this algorithm is globally, asymptotically stable.

In addition, by letting the source s keep track of the rate R, it can be seen that the algorithm is decentralized.

VI. Conclusion

We have described decentralized algorithms that compute minimum-cost subgraphs for establishing multicast connections in networks that use coding. These algorithms complement existing decentralized schemes for constructing network codes [7], [8], [9], [10] well, and they together form a fully decentralized approach for achieving minimum-cost multicast.

The prevailing approach based on approximation algorithms for the directed Steiner tree problem, on the other hand, is suboptimal and generally assumes centralized computation with full network knowledge. Thus, the approach we propose is attractive for any network that services multicast connections, particularly networks such as overlay networks and multihop wireless networks, where legacy issues are not a significant obstacle.

We have also given extensions beyond the basic problem of fixed-rate multicast in networks with directed point-to-point links, and considered the problem of minimum-energy multicast in wireless networks as well as the case of a concave utility function at the sender. In both instances, we obtain a decentralized algorithm for computing an optimal subgraph by making slight modifications to the decentralized algorithms for the basic problem.

The directions for further work that we are currently considering include examining the stability of the algorithms under changing conditions, such as changing link costs, as well as examining their speeds of convergence, both by simulation and analysis. In addition, the primal-dual algorithm presented in Section IV is a continuous time algorithm, whilst a discrete time implementation would be required in practice. This issue needs further investigation.

Acknowledgments

N. Ratnakar and R. Koetter would like to thank R. Srikant for helpful discussions and suggestions.

Appendix I

We wish to solve the following problem.

$$\begin{aligned}
&\text{minimize } \sum_{t\in T} (v^{(t)} - u^{(t)})^2\\
&\text{subject to } v \in P_{ij},
\end{aligned} \quad (46)$$

where P_{ij} is the $|T|$-dimensional simplex

$$P_{ij} = \left\{ v \,\middle|\, \sum_{t \in T} v^{(t)} = a_{ij},\, v \geq 0 \right\}. \qquad (47)$$

First, since the objective function and the constraint set P_{ij} are both convex, it is straightforward to establish that a necessary and sufficient condition for global optimality of $\hat{v}^{(t)}$ in P_{ij} is

$$\hat{v}^{(t)} > 0 \Rightarrow (u^{(t)} - \hat{v}^{(t)}) \geq (u^{(r)} - \hat{v}^{(t)}), \qquad \forall\, r \in T \qquad (48)$$

(see, for example, [24, Section 2.1]). Suppose we index the elements of T such that $u^{(t_1)} \geq u^{(t_2)} \geq \ldots \geq u^{(t_{|T|})}$. We then note that there must be an index k in the set $\{1, \ldots, |T|\}$ such that $v^{(t_l)} > 0$ for $l = 1, \ldots, k$ and $v^{(t_l)} = 0$ for $l > k + 1$, for, if not, then a feasible solution with lower cost can be obtained by swapping around components of the vector. Therefore, condition (48) implies that there must exist some d such that $\hat{v}^{(t)} = u^{(t)} + d$ for all $t \in \{t_1, \ldots, t_k\}$ and that $d \leq -u^{(t)}$ for all $t \in \{t_{k+1}, \ldots, t_{|T|}\}$, which is equivalent to $d \leq -u^{(t_{k+1})}$. Since $\hat{v}^{(t)}$ is in the simplex P_{ij}, it follows that

$$kd + \sum_{t=1}^{t_k} u^{(t)} = a_{ij}, \qquad (49)$$

which gives

$$d = \frac{1}{k} \left(a_{ij} - \sum_{t=1}^{t_k} u^{(t)} \right). \qquad (50)$$

By taking $k = \hat{k}$, where $\hat{k}$ is the smallest k such that

$$\frac{1}{k} \left(a_{ij} - \sum_{r=1}^{t_k} u^{(r)} \right) \leq -u^{(t_{k+1})}, \qquad (51)$$

(or, if no such k exists, then $\hat{k} = |T|$), we see that we have

$$\frac{1}{\hat{k} - 1} \left(a_{ij} - \sum_{t=1}^{t_{k-1}} u^{(t)} \right) > -u^{(t_k)}, \qquad (52)$$

which can be rearranged to give

$$d = \frac{1}{\hat{k}} \left(a_{ij} - \sum_{t=1}^{t_k} u^{(t)} \right) > -u^{(t_k)}. \qquad (53)$$

Hence, if $v^{(t)}$ is given by

$$v^{(t)} = \begin{cases} u^{(t)} + \frac{a_{ij} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}}{\hat{k}} & \text{if } t \in \{t_1, \ldots, t_{\hat{k}}\}, \\ 0 & \text{otherwise,} \end{cases} \qquad (54)$$

then $v^{(t)}$ is feasible and we see that the optimality condition (48) is satisfied. Note that, since $d \leq -u^{(t_{k+1})}$, equation (54) can also be written as

$$v^{(t)} = \max \left(0, u^{(t)} + \frac{1}{\hat{k}} \left(a_{ij} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)} \right) \right). \qquad (55)$$

We now turn to showing that any accumulation point of the sequence of primal iterates $\{x[n]\}$ given by (14) is an optimal solution the primal problem (1). Suppose that the dual feasible solution that the subgradient method converges to is $\bar{p}$. Then there exists some m such that for $n \geq m$

$$p_{ij}^{(t)}[n+1] = p_{ij}^{(t)}[n] + \theta[n] x_{ij}^{(t)}[n] + d_{ij}[n] \qquad (56)$$

for all $(i, j) \in A$ and $t \in T$ such that $\bar{p}_{ij}^{(t)} > 0$. Therefore, if $\bar{p}_{ij}^{(t)} > 0$, then for $n > m$ we have

$$\begin{aligned} \tilde{x}_{ij}^{(t)}[n] &= \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] + \sum_{l=m+1}^{n} \mu_l[n] x_{ij}^{(t)}[l] \\ &= \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] \\ &\quad + \sum_{l=m+1}^{n} \frac{\mu_l[n]}{\theta[n]} (p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n] - d_{ij}[n]) \\ &= \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] \\ &\quad + \sum_{l=m+1}^{n} \gamma_{ln} (p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n]) \\ &\quad - \sum_{l=m+1}^{n} \gamma_{ln} d_{ij}[n]. \end{aligned} \qquad (57)$$

Otherwise, if $\bar{p}_{ij}^{(t)} = 0$, then from equation (55), we have

$$p_{ij}^{(t)}[n+1] \geq p_{ij}^{(t)}[n] + \theta[n] x_{ij}^{(t)}[n] + d_{ij}[n], \qquad (58)$$

so

$$\begin{aligned} \tilde{x}_{ij}^{(t)}[n] \leq \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] \\ + \sum_{l=m+1}^{n} \gamma_{ln} (p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n]) \\ - \sum_{l=m+1}^{n} \gamma_{ln} d_{ij}[n]. \end{aligned} \qquad (59)$$

It is straightforward to see that the sequence of iterates $\{\tilde{x}[n]\}$ is primal feasible, and that we obtain a primal feasible sequence $\{z[n]\}$ by setting $z_{ij}[n] := \max_{t\in T} \tilde{x}_{ij}^{(t)}[n]$. Sherali and Choi [18] showed that, if the required conditions on the step sizes $\{\theta[n]\}$ and convex combination weights $\{\mu_l[n]\}$ are satisfied, then

$$\sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] + \sum_{l=m+1}^{n} \gamma_{ln}(p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n]) \to 0 \tag{60}$$

as $k \to \infty$; hence we see from equations (57) and (59) that, for k sufficiently large,

$$z_{ij}[n] = -\sum_{l=m+1}^{n} \gamma_{ln} d_{ij}[n] \tag{61}$$

and, therefore, that complementary slackness with $\bar{p}$ holds in the limit of any convergent subsequence of $\{\tilde{x}[n]\}$.

Appendix II
Proof of Theorem 1

We prove the stability of the primal-dual algorithm by using the theory of Lyapunov stability. For more details on this, we refer the reader to [25, Section 3.10].

From Equation (19), it can be verified that $(\hat{x}, \hat{p}, \hat{\lambda})$ is an equilibrium point of the primal-dual algorithm. We now prove that this point is globally, asymptotically stable.

Consider the following function as a candidate for the Lyapunov function:

$$\begin{aligned} &V(x, p, \lambda) \\ &\quad= \sum_{t\in T} \Bigg\{ \sum_{(i,j)\in A} \Bigg(\int_{\hat{x}_{ij}^{(t)}}^{x_{ij}^{(t)}} \frac{1}{k_{ij}^{(t)}(\sigma)} (\sigma - \hat{x}_{ij}^{(t)}) d\sigma \\ &\qquad + \int_{\hat{\lambda}_{ij}^{(t)}}^{\lambda_{ij}^{(t)}} \frac{1}{m_{ij}^{(t)}(\gamma)} (\gamma - \hat{\lambda}_{ij}^{(t)}) d\gamma \Bigg) \\ &\qquad + \sum_{i\in N} \int_{\hat{p}_i^{(t)}}^{p_i^{(t)}} \frac{1}{h_i^{(t)}(\beta)} (\beta - \hat{p}_i^{(t)}) d\beta \Bigg\}. \end{aligned} \tag{62}$$

Note that $V(\hat{x}, \hat{p}, \hat{\lambda}) = 0$. Since, $k_{ij}^{(t)}(\sigma) > 0$, if $x_{ij}^{(t)} \neq \hat{x}_{ij}^{(t)}$, we have $\int_{\hat{x}_{ij}^{(t)}}^{x_{ij}^{(t)}} \frac{1}{k_{ij}^{(t)}(\sigma)} (\sigma - \hat{x}_{ij}^{(t)}) d\sigma > 0$. This argument can be extended to the other terms as well. Thus, whenever $(x, p, \lambda) \neq (\hat{x}, \hat{p}, \hat{\lambda})$, we have $V(x, p, \lambda) > 0$.

Now,

$$\begin{aligned} \dot{V} = \sum_{t\in T} \Bigg\{ \sum_{(i,j)\in A} \Bigg[\left(-x_{ij}^{(t)}\right)^{+}_{\lambda_{ij}^{(t)}} (\lambda_{ij}^{(t)} - \hat{\lambda}_{ij}^{(t)}) \\ \left(\frac{\partial U(x)}{\partial x_{ij}^{(t)}} - q_{ij}^{(t)} + \lambda_{ij}^{(t)} \right) \cdot (x_{ij}^{(t)} - \hat{x}_{ij}^{(t)}) \Bigg] \\ + \sum_{i\in N} (y_i^{(t)} - \sigma_i^{(t)})(p_i^{(t)} - \hat{p}_i^{(t)}) \Bigg\}. \end{aligned} \tag{63}$$

Note that

$$\left(-x_{ij}^{(t)}\right)^{+}_{\lambda_{ij}^{(t)}} (\lambda_{ij}^{(t)} - \hat{\lambda}_{ij}^{(t)}) \leq -x_{ij}^{(t)} (\lambda_{ij}^{(t)} - \hat{\lambda}_{ij}^{(t)}), \tag{64}$$

since the inequality is an equality if either $x_{ij}^{(t)} \leq 0$ or $\lambda_{ij}^{(t)} \geq 0$; and, in the case when $x_{ij}^{(t)} > 0$ and $\lambda_{ij}^{(t)} < 0$, we have $(-x_{ij}^{(t)})^{+}_{\lambda_{ij}^{(t)}} = 0$ and, since $\hat{\lambda}_{ij}^{(t)} \geq 0$, $-x_{ij}^{(t)}(\lambda_{ij}^{(t)} - \hat{\lambda}_{ij}^{(t)}) \geq 0$. Therefore,

$$\begin{aligned} \dot{V} &\leq \sum_{t\in T} \Bigg\{ \sum_{(i,j)\in A} \Bigg[-x_{ij}^{(t)} (\lambda_{ij}^{(t)} - \hat{\lambda}_{ij}^{(t)}) \\ &\qquad \left(\frac{\partial U(x)}{\partial x_{ij}^{(t)}} - q_{ij}^{(t)} + \lambda_{ij}^{(t)} \right) \cdot (x_{ij}^{(t)} - \hat{x}_{ij}^{(t)}) \Bigg] \\ &\qquad + \sum_{i\in N} (y_i^{(t)} - \sigma_i^{(t)})(p_i^{(t)} - \hat{p}_i^{(t)}) \Bigg\} \\ &= (\hat{q} - q)'(x - \hat{x}) + (\hat{p} - p)'(y - \hat{y}) \\ &\quad + \sum_{t\in T} \Bigg\{ \sum_{(i,j)\in A} \Bigg[-\hat{x}_{ij}^{(t)} (\lambda_{ij}^{(t)} - \hat{\lambda}_{ij}^{(t)}) \\ &\qquad \left(\frac{\partial U(x)}{\partial x_{ij}^{(t)}} - \hat{q}_{ij}^{(t)} + \hat{\lambda}_{ij}^{(t)} \right) \cdot (x_{ij}^{(t)} - \hat{x}_{ij}^{(t)}) \Bigg] \\ &\qquad + \sum_{i\in N} (\hat{y}_i^{(t)} - \sigma_i^{(t)})(p_i^{(t)} - \hat{p}_i^{(t)}) \Bigg\} \\ &= (\bigtriangledown U(x) - \bigtriangledown U(\hat{x}))'(x - \hat{x}) - \lambda' \hat{x}, \end{aligned} \tag{65}$$

where the last line follows from Karush-Kuhn-Tucker conditions (20)–(24) and the fact that

$$\begin{aligned} p'y &= \sum_{t\in T} \sum_{i\in N} p_i^{(t)} \left(\sum_{\{j|(i,j)\in A\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} \hat{x}_{ji}^{(t)} \right) \\ &= \sum_{t\in T} \sum_{(i,j)\in A} x_{ij}^{(t)} (p_i^{(t)} - p_j^{(t)}) = q'x. \end{aligned} \tag{66}$$

Thus, owing to the strict concavity of $U(x)$, we have $\dot{V} \leq -\lambda' \hat{x}$, with equality if and only if $x = \hat{x}$. So it follows that $\dot{V} \leq 0$ for all $\lambda \geq 0$, since $\hat{x} \geq 0$.

If the initial choice of λ is such that $\lambda(0) \geq 0$, we see from the primal-dual algorithm that $\lambda(\tau) \geq 0$. This is true since $\dot{\lambda} \geq 0$ whenever $\lambda \leq 0$. Thus, it follows by the theory of Lyapunov stability that the algorithm is indeed globally, asymptotically stable.

REFERENCES

[1] S. Ramanathan, "Multicast tree generation in networks with asymmetric links," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, pp. 558–568, Aug. 1996.

[2] L. Zosin and S. Khuller, "On directed Steiner trees," in *Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, 2002, pp. 59–63.

[3] M. Charikar, C. Chekuri, T.-y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," in *Proc. Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1998)*, 1998, pp. 192–200.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[5] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[6] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," accepted to *2004 International Symposium on Information Theory and its Applications (ISITA 2004)*.

[7] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "Toward a random operation of networks," submitted to *IEEE Trans. Inform. Theory*.

[8] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. 41st Annual Allerton Conference on Communication, Control and Computing*, Oct. 2003.

[9] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. 2003 IEEE International Symposium on Information Theory (ISIT 2003)*, 2003.

[10] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conference on Communication, Control and Computing*, Oct. 2003.

[11] T. Ho, B. Leong, M. Médard, R. Koetter, Y.-H. Chang, and M. Effros, "On the utility of network coding in dynamic environments," in *Proc. 2004 International Workshop on Wireless Ad-hoc Networks (IWWAN 2004)*, 2004.

[12] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1992.

[13] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proc. Second Internet Measurement Workshop (IMW 2002)*, 2002.

[14] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, MA: Athena Scientific, 1998.

[15] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice Hall, 1993.

[16] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[17] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY: John Wiley & Sons, Inc., 1999.

[18] H. D. Sherali and G. Choi, "Recovery of primal solutions when using subgradient optimization methods to solve langrangian duals of linear programs," *Oper. Res. Lett.*, vol. 19, pp. 105–113, 1996.

[19] D. P. Bertsekas, L. C. Polymenakos, and P. Tseng, "An ϵ-relaxation method for separable convex cost network flow problems," *SIAM J. Optim.*, vol. 7, no. 3, pp. 853–870, Aug. 1997.

[20] R. de Leone, R. R. Meyer, and A. Zakarian, "A partitioned ϵ-relaxation algorithm for separable convex network flow problems," *Computational Optimization and Applications*, vol. 12, no. 1–3, pp. 107–126, Jan. 1999.

[21] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, pp. 481–492, 2002.

[22] A. Ahluwalia, E. Modiano, and L. Shu, "On the complexity and distributed construction of energy-efficient broadcast trees in static ad hoc wireless networks," in *Proc. 2002 Conference on Information Sciences and Systems (CISS 2002)*, Mar. 2002.

[23] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks," in *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC '02)*, 2002, pp. 112–122.

[24] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[25] R. Srikant, *The Mathematics of Internet Congestion Control*. Boston, MA: Brikhäuser, 2004.