# Minimum-Cost Multicast over Coded Packet Networks

Desmond S. Lun, Niranjan Ratnakar, Muriel Médard, Ralf Koetter,
David R. Karger, Tracey Ho, Ebad Ahmed, and Fang Zhao

**Abstract**

We consider the problem of establishing minimum-cost multicast connections over coded packet networks, i.e. packet networks where the contents of outgoing packets are arbitrary, causal functions of the contents of received packets. We consider both wireline and wireless packet networks as well as both static multicast (where membership of the multicast group remains constant for the duration of the connection) and dynamic multicast (where membership of the multicast group changes in time, with nodes joining and leaving the group).

For static multicast, we reduce the problem to a polynomial-time solvable optimization problem, and we present decentralized algorithms for solving it. These algorithms, when coupled with existing decentralized schemes for constructing network codes, yield a fully decentralized approach for achieving minimum-cost multicast. By contrast, establishing minimum-cost static multicast connections over routed packet networks is a very difficult problem even using centralized computation, except in the special cases of unicast and broadcast connections.

This work was supported by the National Science Foundation under grant nos. CCR-0093349, CCR-0325496, and CCR-0325673, by Hewlett-Packard under grant no. 008542-008, by the Air Force Office of Scientific Research under grant no. F49620-01-1-0365, and by the Vodafone Foundation.

This paper was presented in part at the International Symposium on Information Theory and its Applications, Parma, Italy, October 2004; in part at IEEE Infocom, Miami, FL, March 2005; and in part at the First Workshop on Network Coding, Theory, and Applications, Riva del Garda, Italy, April 2005.

D. S. Lun, M. Médard, E. Ahmed, and F. Zhao are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: dslun@mit.edu, medard@mit.edu, ebad@mit.edu, zhaof@mit.edu).

N. Ratnakar and R. Koetter are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA (e-mail: ratnakar@uiuc.edu, koetter@uiuc.edu).

D. R. Karger is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: karger@mit.edu).

T. Ho is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA (e-mail: tho@caltech.edu)

For dynamic multicast, we reduce the problem to a dynamic programming problem and apply the theory of dynamic programming to suggest how it may be solved.

# Minimum-Cost Multicast over Coded Packet Networks

## I. Introduction

A typical node in today's packet networks is capable of two functions: forwarding (i.e. copying an incoming packet onto an outgoing link) and replicating (i.e. copying an incoming packet onto several outgoing links). But there is no intrinsic reason why we must assume these are the only functions ever permitted to nodes and, in application-level overlay networks and multi-hop wireless networks, for example, allowing nodes to have a wider variety of functions makes sense. We therefore consider packet networks where the contents of outgoing packets are arbitrary, causal functions of the contents of received packets, and we call such networks coded packet networks.

Coded packet networks were put forward by Ahlswede et al. [1], and numerous subsequent papers, e.g., [2], [3], [4], [5], [6], have built upon their work. These papers, however, all assume the availability of dedicated network resources, and scant attention is paid to the problem of determining the allocation of network resources to dedicate to a particular connection or set of connections. This is the problem we tackle. More precisely, we aim to find minimum-cost subgraphs that allow given multicast connections to be established (with appropriate coding) over coded packet networks.

The analogous problem for routed packet networks is old and difficult. It dates to the 1980s and, in the simplest case—that of static multicast in wireline networks with linear cost—it equates to the Steiner tree problem, which is well-known to be NP-complete [7], [8]. The emphasis, therefore, has been on heuristic methods. These methods include heuristics for the Steiner tree problem on undirected (e.g., [7], [9], [8]) and directed (e.g., [10], [11], [12]) graphs, for multicast tree generation in wireless networks (e.g. [13]), and for the dynamic or on-line Steiner tree problem (e.g., [8], [14], [15]). Finding minimum-cost subgraphs in coded packet networks, however, is much easier and as we shall see, in many cases, we are able to find optimal subgraphs in polynomial time using decentralized computation. Moreover, since coded packet networks are less constrained than routed ones, the minimum cost for a given connection

is generally less.

In our problem, we take given multicast connections and thus include unicast and broadcast connections as special cases. But we do not consider optimizing the subgraph for multiple connections taking place simultaneously. One reason for this is that coding for multiple connections is a very difficult problem—one that, in fact, remains currently open with only cumbersome bounds on the asymptotic capability of coding [16] and examples that demonstrate the insufficiency of various classes of linear codes [17], [18], [19], [20]. An obvious, but sub-optimal, approach to coding is to code for each connection separately, which is referred to as superposition coding [21]. When using superposition coding, finding minimum-cost allocations for multiple connections means extending the approach for single connections (namely, the approach taken in this paper) in a straightforward way that is completely analogous to the extension that needs to be done for traditional routed packet networks, and this problem of minimum-cost allocations for multiple connections using superposition coding is addressed in [22]. An alternative approach to coding that outperforms superposition coding, but that remains sub-optimal, is discussed in [23].

We choose here to restrict our attention to single connections because the subgraph selection problem is simpler and because minimum-cost single connections are interesting in their own right: Whenever each multicast group has a selfish cost objective, or when the network sets link weights to meet its objective or enforce certain policies and each multicast group is subject to a minimum-weight objective, we wish to set up single multicast connections at minimum cost.

Finally, we mention that a related problem to subgraph selection, that of throughput maximization, is studied for coded networks in [24], [25] and that an alternative formulation of the subgraph selection problem for coded wireless packet networks is given in [26].

The body of this paper is composed of four sections: Sections II and III deal with static multicast (where membership of the multicast group remains constant for the duration of the connection) for wireline and wireless packet networks, respectively; Section IV gives a comparison of the proposed techniques for static multicast with techniques in routed packet networks; and Section V deals with dynamic multicast (where membership of the multicast group changes in time, with nodes joining and leaving the group). We conclude in Section VI and, in doing so, we give a sampling of the avenues for future investigation that our work opens up.

## II. WIRELINE PACKET NETWORKS

We represent the network with a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. Each arc $(i, j)$ represents a lossless point-to-point link from node $i$ to node $j$. We denote by $z_{ij}$ the rate at which coded packets are injected into arc $(i, j)$. The rate vector $z$, consisting of $z_{ij}$, $(i, j) \in \mathcal{A}$, is called a subgraph, and we assume that it must lie within a constraint set $Z$ for, if not, the packet queues associated with one or more arcs becomes unstable. We reasonably assume that $Z$ is a convex subset of the positive orthant containing the origin. We associate with the network a cost function $f$ (reflecting, for example, the average latency or energy consumption) that maps valid rate vectors to real numbers and that we seek to minimize.

Suppose we have a source node $s$ wishing to transmit packets at a positive, real rate $R$ to a non-empty set of sink nodes $T$. Consider the following optimization problem:

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

where

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

*Theorem 1:* The vector $z$ is part of a feasible solution for the optimization problem (1) if and only if there exists a network code that sets up a multicast connection in the wireline network represented by graph $\mathcal{G}$ at rate arbitrarily close to $R$ from source $s$ to sinks in the set $T$ and that injects packets at rate arbitrarily close to $z_{ij}$ on each arc $(i, j)$.

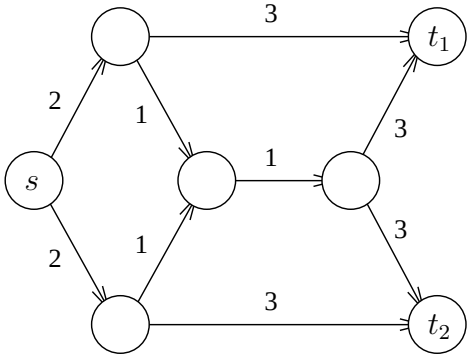
*Proof:* First suppose that $z$ is part of a feasible solution for the problem. Then, for any $t$ in $T$, we see that the maximum flow from $s$ to $t$ in the network where each arc $(i, j)$ has maximum input rate $z_{ij}$ is at least $R$. So, by Theorem 1 of [1], a coding solution that injects packets at rate arbitrarily close to $z_{ij}$ on each arc $(i, j)$ exists. Conversely, suppose that we have a coding solution that injects packets at rate arbitrarily close to $z_{ij}$ on each arc $(i, j)$. Then the

maximum input rate of each arc must be at least $z_{ij}$ and moreover, again by Theorem 1 of [1], flows of size $R$ exist from $s$ to $t$ for each $t$ in $T$. Therefore the vector $z$ is part of a feasible solution for the optimization problem. ■

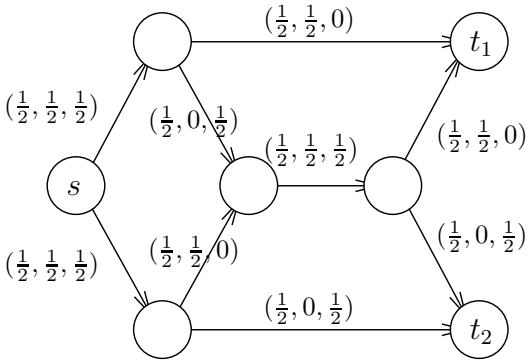
From Theorem 1, it follows immediately that optimization problem (1) finds the optimal cost for an asymptotically-achievable, rate-$R$ multicast connection from $s$ to $T$.

As an example, consider the network depicted in Figure 1(a). We wish to achieve multicast of unit rate to two sinks, $t_1$ and $t_2$. We have $Z = [0,1]^{|\mathcal{A}|}$ and $f(z) = \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij}$, where $a_{ij}$ is the cost per unit rate shown beside each link. An optimal solution to problem (1) for this network is shown in Figure 1(b). We have flows, $x^{(1)}$ and $x^{(2)}$, of unit size from $s$ to $t_1$ and $t_2$, respectively and, for each arc $(i,j)$, $z_{ij} = \max(x_{ij}^{(1)}, x_{ij}^{(2)})$, as we expect from the optimization. To achieve the optimal cost, we code over the subgraph $z$. A code of length 2 for the subgraph is given in [1, Figure 7], which we reproduce in Figure 1(c). In the figure, $X_1$ and $X_2$ refer to the two packets in a coding block. The coding that is performed is that one of the interior nodes receives both $X_1$ and $X_2$ and forms the binary sum of the two, outputting the packet $X_1 + X_2$. The code allows both $t_1$ and $t_2$ to recover both $X_1$ and $X_2$ and it achieves a cost of $19/2$.

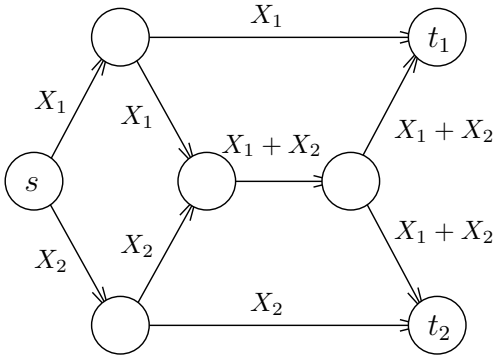
Given a solution of problem (1), there are various coding schemes that can be used to realize the connection. The schemes described in [27], [6] operate continuously, with each node continually sending out packets as causal functions of received packets. The schemes described in [1], [2], [3], [4], [5], on the other hand, operate in a block-by-block manner, with each node sending out a block of packets as a function of its received block. In the latter case, the delay incurred by each arc's block is upper bounded by $\delta/R$ for some non-negative integer $\delta$ provided that $z_{ij}/R \in \mathbb{Z}/\delta$ for all $(i,j) \in \mathcal{A}$. We unfortunately cannot place such constraints into problem (1) since they would make it prohibitively difficult. An alternative is, given $z$, to take $\lceil \delta z/R \rceil R/\delta$ as the subgraph instead. Since $\lceil \delta z/R \rceil R/\delta < (\delta z/R + 1)R/\delta = z + R/\delta$, we can guarantee that $\lceil \delta z/R \rceil R/\delta$ lies in the constraint set $Z$ by looking at $z + R/\delta$ instead of $z$, resulting in the




(a) Each arc is marked with its cost per unit rate.




(b) Each arc is marked with the triple $(z_{ij}, x_{ij}^{(1)}, x_{ij}^{(2)})$.




(c) Each arc is marked with its code.

Fig. 1. A network with multicast from $s$ to $T = \{t_1, t_2\}$.

optimization problem

$$\begin{aligned} &\text{minimize } f(z + R/\delta) \\ &\text{subject to } z + R/\delta \in Z, \\ &\qquad z_{ij} \geq x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in \mathcal{A},\ t \in T, \\ &\qquad \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N},\ t \in T. \end{aligned} \tag{2}$$

We see that, by suitable redefinition of $f$ and $Z$, problem (2) can be reduced to problem (1). Hence, in the remainder of the paper, we focus only on problem (1).

### *A. Linear, separable cost and separable constraints*

The case of linear, separable cost and separable constraints addresses scenarios where a fixed cost (e.g., monetary cost, energy cost, or imaginary weight cost) is paid per unit rate placed on an arc and each arc is subject to a separate constraint (the closed interval from 0 to some non-negative capacity). This is the case in the network depicted in Figure 1(a). So, with each arc $(i,j)$, we associate non-negative numbers $a_{ij}$ and $c_{ij}$, which are the cost per unit rate and the capacity of the arc, respectively. Hence, the optimization problem (1) becomes the following linear optimization problem.

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

Unfortunately, the linear optimization problem (3) as it stands requires centralized computation with full knowledge of the network. Motivated by successful network algorithms such as distributed Bellman-Ford [29, Section 5.2], we seek a decentralized method for solving problem (3), which, when married with decentralized schemes for constructing network codes [5], [6], [27], results in a fully decentralized approach for achieving minimum-cost multicast in the case of linear, separable cost and separable constraints.

Toward the end of developing such an algorithm, we consider the Lagrangian dual problem

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

where

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

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

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

and capacity constraints

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

Subproblem (5) is a standard linear minimum-cost flow problem, which can be solved using a multitude of different methods (see, for example, [30, Chapters 4–7] or [31, Chapters 9–11]); in particular, it can be solved in an asynchronous, distributed manner using the $\varepsilon$-relaxation method [32, Sections 5.3 and 6.5]. In addition, if the connection rate is small compared to the arc capacities (more precisely, if $R \leq c_{ij}$ for all $(i,j)\in\mathcal{A}$), then subproblem (5) reduces to a shortest path problem, which admits a simple, asynchronous, distributed solution [29, Section 5.2].

Now, to solve the dual problem (4), we employ subgradient optimization (see, for example, [33, Section 6.3.1] or [34, 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

$$p_{ij}[n+1] := \arg\min_{v\in P_{ij}} \sum_{t\in T} (v^{(t)} - (p_{ij}^{(t)}[n] + \theta[n] x_{ij}^{(t)}[n]))^2 \tag{6}$$

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

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

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

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

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

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

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 due to Sherali and Choi [35].

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

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

and

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

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 (3) from any accumulation point of the sequence of primal iterates $\{\tilde{x}[n]\}$ given by

$$\tilde{x}[n] := \sum_{l=1}^{n} \mu_l[n]x[l], \quad n = 0, 1, \ldots. \tag{7}$$

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 [35, 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

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

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

We now have a relatively simple algorithm for computing optimal feasible solutions to problem (3) 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 arcs. For example, for all arcs $(i, j)$ in $\mathcal{A}$, 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 arcs for subproblem (5) for each $t \in T$, we can apply the $\varepsilon$-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 (6) and (7), respectively. We then re-apply the $\varepsilon$-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 $\varepsilon$-relaxation method for convex cost network flow problems [36], [37]), a significantly more direct and natural method is possible, which we proceed to present.

*B. Convex, separable cost and separable constraints*

Let us now consider the case where, rather than a cost per unit rate for each arc, we have a convex, monotonically increasing cost function $f_{ij}$ for arc $(i,j)$. Such cost functions arise naturally when the cost is, e.g., latency or congestion. The optimization problem (1) becomes the following convex optimization problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in\mathcal{A}} f_{ij}(z_{ij}) \\ \text{subject to} \quad & z_{ij} \geq x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T, \\ & \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i\in\mathcal{N},\ t\in T. \end{aligned} \tag{8}$$

Note that the capacity constraints have been removed, since they can be enforced by making arcs arbitrarily costly as their flows approach their respective capacities. We again seek a decentralized method for solving the subgraph selection problem.

We note that $z_{ij} = \max_{t\in T} x_{ij}^{(t)}$ at an optimal solution of problem (8) and that $f_{ij}(\max_{t\in T} x_{ij}^{(t)})$ is a convex function of $x_{ij}$ since a monotonically increasing, convex function of a convex function is convex. Hence it follows that problem (8) can be restated as the following convex optimization problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in\mathcal{A}} f_{ij}(z_{ij}) \\ \text{subject to} \quad & z_{ij} = \max_{t\in T} x_{ij}^{(t)}, \qquad \forall\ (i,j)\in\mathcal{A}, \\ & \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i\in\mathcal{N},\ t\in T, \\ & x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T. \end{aligned} \tag{9}$$

Unfortunately, the max function is not everywhere differentiable, and this can pose problems for algorithm design. We therefore solve the following modification of problem (9) where the

max norm is replaced by an $l^n$-norm. This replacement was originally proposed in [38].

$$\begin{aligned}
&\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} f_{ij}(z'_{ij}) \\
&\text{subject to } z'_{ij} = \left(\sum_{t\in T}(x_{ij}^{(t)})^n\right)^{1/n}, \qquad \forall\ (i,j)\in\mathcal{A}, \\
&\qquad\qquad \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i\in\mathcal{N},\ t\in T, \\
&\qquad\qquad x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T.
\end{aligned} \tag{10}$$

We have that $z'_{ij} \geq z_{ij}$ for all $n > 0$ and that $z'_{ij}$ approaches $z_{ij}$ as $n$ approaches infinity. Thus, we shall assume that $n$ is large and attempt to develop a decentralized algorithm to solve problem (10). Note that, since $z'_{ij} \geq z_{ij}$, a code with rate $z'_{ij}$ on each arc $(i,j)$ exists for any feasible solution.

Problem (10) is a convex multicommodity flow problem. There are many algorithms for convex multicommodity flow problems (see [39] for a survey), some of which (e.g. the algorithms in [40], [41]) are well-suited for decentralized implementation. These algorithms can certainly be used, but, in this paper, we propose solving problem (10) using a primal-dual algorithm derived from the primal-dual approach to internet congestion control (see [42, Section 3.4]).

We restrict ourselves to the case where $\{f_{ij}\}$ are strictly convex. Since the variable $z'_{ij}$ is a strictly convex function of $x_{ij}$, it follows that the objective function for problem (10) is strictly convex, so the problem admits a unique solution for any integer $n > 0$. Let $U(x) := -\sum_{(i,j)\in\mathcal{A}} f_{ij}((\sum_{t\in T}(x_{ij}^{(t)})^n)^{1/n})$, and let $(y)_x^+$ for $x \geq 0$ denote the following function of $y$:

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

Consider the following continuous-time primal-dual algorithm:

$$\dot{x}_{ij}^{(t)} = k_{ij}^{(t)}(x_{ij}^{(t)}) \left(\frac{\partial U(x)}{\partial x_{ij}^{(t)}} - q_{ij}^{(t)} + \lambda_{ij}^{(t)}\right), \tag{11}$$

$$\dot{p}_i^{(t)} = h_i^{(t)}(p_i^{(t)})(y_i^{(t)} - \sigma_i^{(t)}), \tag{12}$$

$$\dot{\lambda}_{ij}^{(t)} = m_{ij}^{(t)}(\lambda_{ij}^{(t)}) \left(-x_{ij}^{(t)}\right)_{\lambda_{ij}^{(t)}}^+, \tag{13}$$

where

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

$$y_i^{(t)} := \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)},$$

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.

*Proposition 1:* The algorithm specified by Equations (11)–(13) 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 (10). We have to choose $\lambda$, however, with non-negative entries as the initial choice.

We associate a processor with each arc $(i, j)$ 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 arcs.

We assume that the processor for node $i$ keeps track of the variables $\{p_i^{(t)}\}_{t\in T}$, while the processor for arc $(i, j)$ keeps track of the variables $\{\lambda_{ij}^{(t)}\}_{t\in T}$ and $\{x_{ij}^{(t)}\}_{t\in T}$. With this assumption, the algorithm is decentralized in the following sense:

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

This fact is evident from equations (11)–(13) by noting that

$$\frac{\partial U(x)}{\partial x_{ij}^{(t)}} = -f_{ij}(z'_{ij})\left(x_{ij}^{(t)}/z'_{ij}\right)^{n-1}.$$

In implementing the primal-dual algorithm, we must bear the following points in mind.

- The primal-dual algorithm in (11)–(13) is a continuous time algorithm. To discretize the

algorithm, we consider time steps $m = 1, 2, \ldots$ and replace the derivatives by differences:

$$x_{ij}^{(t)}[m+1] = x_{ij}^{(t)}[m] + \alpha_{ij}^{(t)}[m]\left(\frac{\partial U(x[m])}{\partial x_{ij}^{(t)}[m]} - q_{ij}^{(t)}[m] + \lambda_{ij}^{(t)}[m]\right),$$

$$p_i^{(t)}[m+1] = p_i^{(t)}[m] + \beta_i^{(t)}[m](y_i^{(t)}[m] - \sigma_i^{(t)}),$$

$$\lambda_{ij}^{(t)}[m+1] = \lambda_{ij}^{(t)}[m] + \gamma_{ij}^{(t)}[m]\left(-x_{ij}^{(t)}[m]\right)^+_{\lambda_{ij}^{(t)}[m]},$$

where

$$q_{ij}^{(t)}[m] := p_i^{(t)}[m] - p_j^{(t)}[m],$$

$$y_i^{(t)}[m] := \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)}[m] - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)}[m],$$

and $\alpha_{ij}^{(t)}[m] > 0$, $\beta_i^{(t)}[m] > 0$, and $\gamma_{ij}^{(t)}[m] > 0$ can be thought of as step sizes.

- While the algorithm is guaranteed to converge to the optimum solution, the value of the variables at any time instant $m$ is not necessarily a feasible solution. A start-up time is required before a feasible solution is computed.
- Unfortunately, the above algorithm is a synchronous algorithm where the various processors need to exchange information at regular intervals. It is an interesting problem to investigate an asynchronous implementation of the primal-dual algorithm.

### *C. Elastic rate demand*

We have thus far focused on the case of an inelastic rate demand, which is presumably provided by a separate flow control algorithm. But this flow control does not necessarily need to be done separately. Thus, we now suppose that the rate demand is elastic and that it is represented by a utility function that has the same units as the cost function, and we seek to maximize utility minus cost. We continue to assume strictly convex, separable cost and separable constraints.

We associate with the source a utility function $U_r$ such that $U_r(R)$ is the utility derived by the source when $R$ is the data rate. The function $U_r$ is assumed to be a strictly concave and

increasing. Hence, in this setup, the problem we address is as follows:

$$\begin{aligned} &\text{maximize } U(x,R) \\ &\text{subject to } \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N}\setminus\{t\},\ t\in T, \\ &R \geq 0, \\ &x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T, \end{aligned} \tag{14}$$

where $U(x,R) := U_r(R) - \sum_{(i,j)\in\mathcal{A}} f_{ij}(\sum_{t\in T}(x_{ij}^{(t)})^n)^{1/n})$. In problem (14), some of the flow constraints have been dropped by making the observation that the equality constraints at a sink $t$, namely

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

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.

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

$$\begin{aligned} \dot{x}_{ij}^{(t)} &= k_{ij}^{(t)}(x_{ij}^{(t)})\left(\frac{\partial U(x,R)}{\partial x_{ij}^{(t)}} - q_{ij}^{(t)} + \lambda_{ij}^{(t)}\right), \\ \dot{R} &= k_R(R)\left(\frac{\partial U(x,R)}{\partial R} - q_R + \lambda_R\right), \\ \dot{p}_i^{(t)} &= h_i^{(t)}(p_i^{(t)})y_i^{(t)}, \\ \dot{\lambda}_{ij}^{(t)} &= m_{ij}^{(t)}(\lambda_{ij}^{(t)})\left(-x_{ij}^{(t)}\right)_{\lambda_{ij}^{(t)}}^{+}, \\ \dot{\lambda}_R &= m_R(\lambda_R)\left(-R\right)_{\lambda_R}^{+}, \end{aligned}$$

where

$$\begin{aligned} q_{ij}^{(t)} &:= p_i^{(t)} - p_j^{(t)}, \\ q_R &:= -\sum_{t\in T} p_s^{(t)}, \\ y_i^{(t)} &:= \sum_{\{j|(i,j)\in\mathcal{A}\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} \hat{x}_{ji}^{(t)} - \sigma_i^{(t)}. \end{aligned}$$

It can be shown using similar arguments as those for Proposition 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.

## III. Wireless packet networks

To model wireless packet networks, we take the model for wireline packet networks and include the effect of two new factors: link lossiness and link broadcast. Link lossiness refers to the dropping or loss of packets as they are transmitted over a link; and link broadcast refers to how links, rather than necessarily being point-to-point, may originate from a single node and reach more than one other node. Our model includes networks consisting of lossy point-to-point links and networks consisting of lossless broadcast links as special cases.

We represent the network with a directed hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of hyperarcs. A hypergraph is a generalization of a graph, where, rather than arcs, we have hyperarcs. A hyperarc is a pair $(i, J)$, where $i$, the start node, is an element of $\mathcal{N}$ and $J$, the set of end nodes, is a non-empty subset of $\mathcal{N}$. Each hyperarc $(i, J)$ represents a lossy broadcast link from node $i$ to nodes in the non-empty set $J$. We denote by $z_{iJ}$ the rate at which coded packets are injected into hyperarc $(i, J)$, and we denote by $z_{iJK}$ the rate at which packets, injected into hyperarc $(i, J)$, are received by exactly the set of nodes $K \subset J$. Hence $z_{iJ} := \sum_{K \subset J} z_{iJK}$. Let

$$b_{iJK} := \frac{\sum_{\{L \subset J | L \cap K \neq \emptyset\}} z_{iJL}}{z_{iJ}}.$$

The rate vector $z$, consisting of $z_{iJ}$, $(i, J) \in \mathcal{A}$, is called a subgraph, and we assume that it must lie within a constraint set $Z$ for, if not, the packet queues associated with one or more hyperarcs becomes unstable (for examples of constraint sets $Z$ that pertain specifically to multi-hop wireless networks, see [43], [44], [45], [46], [47], [48]). We reasonably assume that $Z$ is a convex subset of the positive orthant containing the origin. We associate with the network a cost function $f$ (reflecting, for example, the average latency or energy consumption) that maps valid rate vectors to real numbers and that we seek to minimize.

Suppose we have a source node $s$ wishing to transmit packets at a positive, real rate $R$ to a

non-empty set of sink nodes $T$. Consider the following optimization problem:

$$\begin{aligned}
&\text{minimize } f(z)\\
&\text{subject to } z \in Z,\\
&\qquad z_{iJ} b_{iJK} \geq \sum_{j\in K} x_{iJj}^{(t)}, \qquad \forall\ (i,J) \in \mathcal{A},\ K \subset J,\ t \in T,\\
&\qquad \sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj}^{(t)} - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N},\ t \in T,\\
&\qquad x_{iJj}^{(t)} \geq 0, \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J,\ t \in T.
\end{aligned} \tag{15}$$

*Theorem 2:* The vector $z$ is part of a feasible solution for the optimization problem (15) if and only if there exists a network code that sets up a multicast connection in the wireless network represented by hypergraph $\mathcal{H}$ at rate arbitrarily close to $R$ from source $s$ to sinks in the set $T$ and that injects packets at rate arbitrarily close to $z_{iJ}$ on each hyperarc $(i,J)$.

*Proof:* The proof is much the same as that for Theorem 1. But, instead of Theorem 1 of [1], we use Theorem 2 of [6]. ■

In the lossless case, we have $b_{iJK} = 1$ for all non-empty $K \subset J$ and $b_{iJ\emptyset} = 0$. Hence, problem (15) simplifies to the following optimization problem.

$$\begin{aligned}
&\text{minimize } f(z)\\
&\text{subject to } z \in Z,\\
&\qquad z_{iJ} \geq \sum_{j\in J} x_{iJj}^{(t)}, \qquad \forall\ (i,J) \in \mathcal{A},\ t \in T,\\
&\qquad \sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj}^{(t)} - \sum_{\{j|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N},\ t \in T,\\
&\qquad x_{iJj}^{(t)} \geq 0, \qquad \forall\ (i,J) \in \mathcal{A},\ j \in J,\ t \in T.
\end{aligned} \tag{16}$$

A simplification of problem (16) can be made if we assume that, when nodes transmit in a lossless network, they reach all nodes in a certain area, with cost increasing as this area is increased. More precisely, suppose that we have separable cost, so $f(z) = \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ})$. Suppose further that each node $i$ has $M_i$ outgoing hyperarcs $(i, J_1^{(i)}), (i, J_2^{(i)}), \ldots, (i, J_{M_i}^{(i)})$ with $J_1^{(i)} \subsetneq J_2^{(i)} \subsetneq \cdots \subsetneq J_{M_i}^{(i)}$. (We assume that there are no identical links, as duplicate links can effectively be treated as a single link.) Then, we assume that $f_{iJ_1^{(i)}}(\zeta) < f_{iJ_2^{(i)}}(\zeta) < \cdots < f_{iJ_{M_i}^{(i)}}(\zeta)$

for all $\zeta \geq 0$ and nodes $i$. For $(i,j) \in \mathcal{A}' := \{(i,j) | (i,J) \in A, J \ni j\}$, we introduce the variables

$$\hat{x}_{ij}^{(t)} := \sum_{m=m(i,j)}^{M_i} x_{iJ_m^{(i)}j}^{(t)},$$

where $m(i,j)$ is the unique $m$ such that $j \in J_m^{(i)} \setminus J_{m-1}^{(i)}$ (we define $J_0^{(i)} := \emptyset$ for all $i \in \mathcal{N}$ for convenience). Now, problem (16) can be reformulated as the following optimization problem, which has substantially fewer variables.

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{iJ}) \\
\text{subject to} \quad & z \in Z, \\
& \sum_{n=m}^{M_i} z_{iJ_n^{(i)}} \geq \sum_{k \in J_{M_i}^{(i)} \setminus J_{m-1}^{(i)}} \hat{x}_{ik}^{(t)}, \qquad \forall\ i \in \mathcal{N},\ m = 1,\ldots,M_i,\ t \in T, \\
& \sum_{\{j|(i,j)\in\mathcal{A}'\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}'\}} \hat{x}_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N},\ t \in T, \\
& \hat{x}_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in \mathcal{A}',\ t \in T.
\end{aligned} \tag{17}$$

*Proposition 2:* Suppose that $f(z) = \sum_{(i,J)\in\mathcal{A}} f_{iJ}(z_{ij})$ and that $f_{iJ_1^{(i)}}(\zeta) < f_{iJ_2^{(i)}}(\zeta) < \cdots < f_{iJ_{M_i}^{(i)}}(\zeta)$ for all $\zeta \geq 0$ and nodes $i$. Then problem (16) and problem (17) are equivalent in the sense that they have the same optimal cost and $z$ is part of an optimal solution for (16) if and only if it is part of an optimal solution for (17).

*Proof:* See Appendix III. ■

We see that, provided that $\{b_{iJK}\}$ are constant, problems (15) and (16) are of essentially the same form as problem (1), albeit with possibly more linear constraints relating $z$ and $x$, and, if we drop the constraint set $Z$ and consider linear, separable cost or convex, separable cost, then the decentralized algorithms discussed in Sections II-A and II-B can be applied with little modification. In the case of problem (17), the subgradient method of Section II-A can be applied once we note that its Lagrangian dual,

$$\begin{aligned}
\text{maximize} \quad & \sum_{t\in T} \hat{q}^{(t)}(p^{(t)}) \\
\text{subject to} \quad & \sum_{t\in T} p_{iJ_m^{(i)}}^{(t)} = s_{iJ_m^{(i)}}, \qquad \forall\ i \in \mathcal{N},\ m = 1,\ldots,M_i, \\
& p_{iJ}^{(t)} \geq 0, \qquad \forall\ (i,J) \in \mathcal{A},\ t \in T,
\end{aligned}$$

where

$$s_{iJ_m^{(i)}} := a_{iJ_m^{(i)}} - a_{iJ_{m-1}^{(i)}},$$

and

$$\hat{q}^{(t)}(p^{(t)}) := \min_{\hat{x}^{(t)} \in \hat{F}^{(t)}} \sum_{(i,j) \in \mathcal{A}'} \left( \sum_{m=1}^{m(i,j)} p^{(t)}_{iJ_m^{(i)}} \right) \hat{x}^{(t)}_{ij},$$

is of the same form as (4).

## IV. Comparison with techniques in routed packet networks

In this section, we report on the results of several simulations that we conducted to assess the performance of the proposed techniques. We begin with wireline networks.

In routed wireline networks, the standard approach to establishing minimum-cost multicast connections is to find the shortest tree rooted at the source that reaches all the sinks, which equates to solving the Steiner tree problem on directed graphs [10]. For coded networks, the analogous problem to finding the shortest tree is solving the linear optimization problem (3) in the case where $c_{ij} = +\infty$, which, being a linear optimization problem, admits a polynomial-time solution. By contrast, the Steiner tree problem on directed graphs is well-known to be NP-complete. Although tractable approximation algorithms exist for the Steiner tree problem on directed graphs (for example, [10], [11], [12]), the solutions thus obtained are suboptimal relative to minimum-cost multicast without coding, which in turn is suboptimal relative to when coding is used, since coding subsumes forwarding and replicating (for example, the optimal cost for a Steiner tree in the network in Figure 1(a) is 10, as opposed to $19/2$). Thus, coding promises potentially significant cost improvements.

We conducted simulations where we took graphs representing various Internet Service Provider (ISP) networks and assessed the average total weight of random multicast connections using, first, our proposed network-coding based solution and, second, routing over the tree given by the Directed Steiner Tree (DST) approximation algorithm described in [11]. The graphs, and their associated link weights, were obtained from the Rocketfuel project of the University of Washington [28]. The approximation algorithm in [11] was chosen for comparison as it achieves a poly-logarithmic approximation ratio (it achieves an approximation ratio of $O(\log^2 |T|)$, where $|T|$ is the number of sink nodes), which is roughly as good as can be expected from any practical algorithm, since it has been shown that it is highly unlikely that there exists a polynomial-time

algorithm that can achieve an approximation factor smaller than logarithmic [10]. The results of the simulations are tabulated in Table I. We see that, depending on the network and the size of the multicast group, the average cost reduction ranges from 10% to 33%. Though these reductions are modest, it is important to keep in mind that our proposed solution easily accommodates decentralized operation.

For wireless networks, one specific problem of interest is that of minimum-energy multicast (see, for example, [13], [49]). In this problem, we wish to achieve minimum-energy multicast in a lossless wireless network without explicit regard for throughput or bandwidth, so the constraint set $Z$ can be dropped altogether. The cost function is linear and separable, namely, it is $f(z) = \sum_{(i,J)\in\mathcal{A}} a_{iJ} z_{iJ}$, where $a_{iJ}$ represents the energy required to transmit a packet to nodes in $J$ from node $i$. Hence problem (17) becomes a linear optimization problem with a polynomial number of constraints, which can therefore be solved in polynomial time. By contrast, the same problem using traditional routing-based approaches is NP-complete—in fact, the special case of broadcast in itself is NP-complete, a result shown in [49], [50]. The problem must therefore be addressed using polynomial-time heuristics such as the MIP algorithm proposed in [13].

We conducted simulations where we placed nodes randomly in a $10 \times 10$ square with a radius of connectivity of 3 and assessed the average total energy of random multicast connections using first, our proposed network-coding based solution and, second, the routing solution given by the MIP algorithm. The energy required to transmit at rate $z$ to a distance $d$ was taken to be $d^2 z$. The results of the simulations are tabulated in Table II. We see that, depending on the size of the network and the size of the multicast group, the average energy reduction ranges from 13% to 49%. These reductions are more substantial than those for the wireline simulations, but are still modest. Again, it is important to keep in mind that the proposed solution easily accommodates decentralized operation.

We conducted simulations on our decentralized algorithms under the same set up, for a network of 30 nodes and a multicast group of 4 terminals. In Figure 2, we show the average behavior of the subgradient method of Section II-A applied to problem (17). The algorithm was run under two choices of step sizes and convex combination weights. The curve labeled "original primal recovery" refers to the case where the step sizes are given by $\theta[n] = n^{-0.8}$ and the convex combination weights by $\mu_l[n] = 1/n$. The curve labeled "modified primal recovery" refers to the case where the step sizes are given by $\theta[n] = n^{-0.8}$ and the convex combination weights by

| Network | Approach | Average multicast cost | | | |
|---|---|---|---|---|---|
| | | 2 sinks | 4 sinks | 8 sinks | 16 sinks |
| Telstra (au) | DST approximation | 17.0 | 28.9 | 41.7 | 62.8 |
| | Network coding | 13.5 | 21.5 | 32.8 | 48.0 |
| Sprint (us) | DST approximation | 30.2 | 46.5 | 71.6 | 127.4 |
| | Network coding | 22.3 | 35.5 | 56.4 | 103.6 |
| Ebone (eu) | DST approximation | 28.2 | 43.0 | 69.7 | 115.3 |
| | Network coding | 20.7 | 32.4 | 50.4 | 77.8 |
| Tiscali (eu) | DST approximation | 32.6 | 49.9 | 78.4 | 121.7 |
| | Network coding | 24.5 | 37.7 | 57.7 | 81.7 |
| Exodus (us) | DST approximation | 43.8 | 62.7 | 91.2 | 116.0 |
| | Network coding | 33.4 | 49.1 | 68.0 | 92.9 |
| Abovenet (us) | DST approximation | 27.2 | 42.8 | 67.3 | 75.0 |
| | Network coding | 21.8 | 33.8 | 60.0 | 67.3 |

TABLE I

AVERAGE COST OF RANDOM MULTICAST CONNECTIONS OF UNIT RATE FOR VARIOUS APPROACHES IN GRAPHS REPRESENTING VARIOUS ISP NETWORKS. THE COST PER UNIT RATE ON EACH ARC IS THE LINK WEIGHT AS ASSESSED BY THE ROCKETFUEL PROJECT OF THE UNIVERSITY OF WASHINGTON [28]. SOURCE AND SINK NODES WERE SELECTED ACCORDING TO A UNIFORM DISTRIBUTION OVER ALL POSSIBLE SELECTIONS.

| Network size | Approach | Average multicast energy | | | |
|---|---|---|---|---|---|
| | | 2 sinks | 4 sinks | 8 sinks | 16 sinks |
| 20 nodes | MIP algorithm | 30.6 | 33.8 | 41.6 | 47.4 |
| | Network coding | 15.5 | 23.3 | 29.9 | 38.1 |
| 30 nodes | MIP algorithm | 26.8 | 31.9 | 37.7 | 43.3 |
| | Network coding | 15.4 | 21.7 | 28.3 | 37.8 |
| 40 nodes | MIP algorithm | 24.4 | 29.3 | 35.1 | 42.3 |
| | Network coding | 14.5 | 20.6 | 25.6 | 30.5 |
| 50 nodes | MIP algorithm | 22.6 | 27.3 | 32.8 | 37.3 |
| | Network coding | 12.8 | 17.7 | 25.3 | 30.3 |

TABLE II

AVERAGE ENERGY OF RANDOM MULTICAST CONNECTIONS OF UNIT RATE FOR VARIOUS APPROACHES IN RANDOM WIRELESS NETWORKS OF VARYING SIZE. NODES WERE PLACED RANDOMLY WITHIN A $10 \times 10$ SQUARE WITH A RADIUS OF CONNECTIVITY OF 3. THE ENERGY REQUIRED TO TRANSMIT AT RATE $z$ TO A DISTANCE $d$ WAS TAKEN TO BE $d^2 z$. SOURCE AND SINK NODES WERE SELECTED ACCORDING TO A UNIFORM DISTRIBUTION OVER ALL POSSIBLE SELECTIONS.

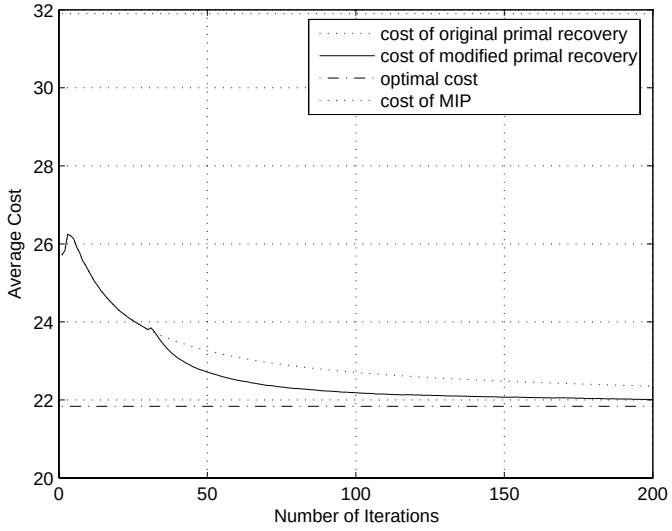



Fig. 2. Average energy of a random 4-terminal multicast of unit rate in a 30-node wireless network using the subgradient method of Section II-A. Nodes were placed randomly within a $10 \times 10$ square with a radius of connectivity of 3. The energy required to transmit at rate $z$ to a distance $d$ was taken to be $d^2 z$. Source and sink nodes were selected according to a uniform distribution over all possible selections.

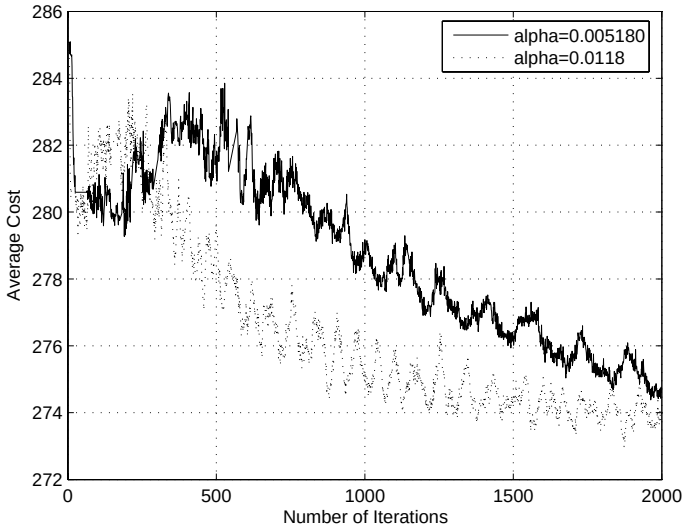



Fig. 3. Average energy of a random 4-terminal multicast of unit rate in a 30-node wireless network using the primal-dual method of Section II-B. Nodes were placed randomly within a $10 \times 10$ square with a radius of connectivity of 3. The energy required to transmit at rate $z$ to a distance $d$ was taken to be $d^2 e^z$. Source and sink nodes were selected according to a uniform distribution over all possible selections.

$\mu_l[n] = 1/n$, if $n < 30$, and $\mu_l[n] = 1/30$, if $n \geq 30$. The modified primal recovery rule was chosen as a heuristic to lessen the effect of poor primal solutions obtained in early iterations. For reference, the optimal cost of problem (17) is shown, as is the cost obtained by the MIP algorithm. We see that, for both choices of step sizes and convex combination weights, the cost after the first iteration is already lower than that from the MIP algorithm. Moreover, in fewer than 50 iterations, the cost using modified primal recovery is within $5\%$ of the optimal value. Thus, in a small number of iterations, the subgradient method yields significantly lower energy consumption than that obtained by the MIP algorithm, which is centralized.

In Figure 3, we show the average behavior of the primal-dual method of Section II-B applied to problem (16). To make the cost strictly convex, the energy required to transmit at rate $z$ to a distance $d$ was taken to be $d^2e^z$. Recall that we do not necessarily have a feasible solution at each iteration. Thus, to compare the cost at the end of each iteration, we recover a feasible solution from the vector $z'[m]$ as follows: We take the subgraph defined by $z'[m]$ and compute the maximum flow from source $s$ to sinks in the set $\mathcal{T}$. We then find *any* subgraph of $z'[m]$ that provides this maximum flow and scale the subgraph so obtained to provide the desired flow. The cost of the scaled subgraph is assumed to be the cost of the solution at the end of each iteration. We chose the step sizes as follows: $\alpha_{ij}^{(t)}[m] = \alpha$, $\beta_i^{(t)}[m] = 20\alpha$, and $\gamma_{ij}^{(t)}[m]$ was chosen to be large. The algorithm was run under two choices of $\alpha$. We see, from our results, that the value of $\alpha$ has to be carefully chosen. Larger values of $\alpha$ generally lead to more oscillatory behavior but faster convergence.

## V. Dynamic multicast

In many applications, membership of the multicast group changes in time, with nodes joining and leaving the group, rather than remaining constant for the duration of the connection, as we have thus far assumed. Under these dynamic conditions, we often cannot simply re-establish the connection with every membership change because doing so would cause an unacceptable disruption in the service being delivered to those nodes remaining in the group. A good example of an application where such issues arise is real-time media distribution. Thus, we desire to find minimum-cost time-varying subgraphs that can deliver continuous service to dynamic multicast groups.

Although our objective is clear, our description of the problem is currently vague. Indeed,

one of the principal hurdles to tackling the problem of dynamic multicast lies in formulating the problem in such a way that it is suitable for analysis and addresses our objective. For routed networks, the problem is generally formulated as the dynamic Steiner tree problem, which was first proposed in [14]. Under this formulation, the focus is on worst-case behavior and modifications of the multicast tree are allowed only when nodes join or leave the multicast group. The formulation is adequate, but not compelling; indeed, there is no compelling reason for the restriction on when the multicast tree can be modified.

In our formulation for coded networks, we draw some inspiration from [14], but we focus on expected behavior rather than worst-case behavior, and we do not restrict modifications of the multicast subgraph to when nodes join or leave the multicast tree. We focus on wireline networks for simplicity, though our considerations apply equally to wireless networks. We formulate the problem as follows.

We employ a basic unit of time that is related to the time that it takes for changes in the multicast subgraph to settle. In particular, suppose that at a given time the multicast subgraph is $z$ and that it is capable of supporting a multicast connection to sink nodes $T$. Then, in one unit time, we can change the multicast subgraph to $z'$, which is capable of supporting a multicast connection to sink nodes $T'$, without disrupting the service being delivered to $T \cap T'$ provided that (componentwise) $z \geq z'$ or $z \leq z'$. The interpretation of this assumption is that we allow, in one time unit, only for the subgraph to increase, meaning that any sink node receiving a particular stream will continue to receive it (albeit with possible changes in the code, depending on how the coding is implemented) and therefore facing no significant disruption to service; or for the subgraph to decrease, meaning that any sink node receiving a particular stream will be forced to reduce to a subset of that stream, but one that is sufficient to recover the source's transmission provided that the sink node is in $T'$, and therefore again facing no significant disruption to service. We do not allow for both operations to take place in a single unit of time (which would allow for arbitrary changes) because, in that case, sink nodes may face temporary disruptions to service when decreases to the multicast subgraph follow too closely to increases.

As an example, consider the four node network shown in Figure 4. Suppose that $s = 1$ and that, at a given time, we have $T = \{2, 4\}$. We support a multicast of unit rate with the subgraph

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (1, 0, 1, 0).$$

Now suppose that the group membership changes, and node 2 leaves while node 3 joins, so $T' = \{3, 4\}$. As a result, we decide that we wish to change to the subgraph

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (0, 1, 0, 1).$$

If we simply make the change naïvely in a single time unit, then node 4 may face a temporary disruption to its service as packets on $(2, 4)$ stop arriving and before packets on $(3, 4)$ start arriving. The assumption that we have made on allowed operations ensures that we must first increase the subgraph to

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (1, 1, 1, 1),$$

allow for the change to settle by waiting for one time unit, then decrease the subgraph to

$$(z_{12}, z_{13}, z_{24}, z_{34}) = (0, 1, 0, 1).$$

With this series of operations, node 4 maintains continuous service throughout the subgraph change.

We discretize the time axis into time intervals of a single time unit. We suppose that at the beginning of each time interval, we receive zero or more requests from sink nodes that are not currently part of the multicast group to join and zero or more requests from sink nodes that are currently part of the multicast group to leave. We model these join and leave requests as a discrete stochastic process and make the assumption that, once all the members of the multicast group leave, the connection is over and remains in that state forever. Let $T_m$ denote the sink nodes in the multicast group at the end of time interval $m$. Then, we assume that

$$\lim_{m \to \infty} \Pr(T_m \neq \emptyset | T_0 = T) = 0 \tag{18}$$

for any initial multicast group $T$. A possible, simple model of join and leave requests is to model $|T_m|$ as a birth-death process with a single absorbing state at state 0, and to choose a node uniformly from $\mathcal{N}' \setminus T_m$, where $\mathcal{N}' := \mathcal{N} \setminus \{s\}$, at each birth and from $T_m$ at each death.

Let $z^{(m)}$ be the multicast subgraph at the beginning of time interval $m$, which, by the assumptions made thus far, means that it supports a multicast connection to sink nodes $T_{m-1}$. Let $V_{m-1}$ and $W_{m-1}$ be the join and leave requests that arrive at the end of time interval $m-1$, respectively. Hence, $V_{m-1} \subset \mathcal{N}' \setminus T_{m-1}$, $W_{m-1} \subset T_{m-1}$, and $T_m = (T_{m-1} \setminus W_{m-1}) \cup V_{m-1}$. We choose $z^{(m+1)}$ from $z^{(m)}$ and $T_m$ using the function $\mu_m$, so $z^{(m+1)} = \mu_m(z^{(m)}, T_m)$, where $z^{(m+1)}$ must lie in a particular constraint set $U(z^{(m)}, T_m)$.

To characterize the constraint set $U(z,T)$, recall the optimization problem for minimum-cost multicast in wireline packet networks developed in Section II:

$$\begin{aligned} &\text{minimize } f(z) \\ &\text{subject to } z \in Z, \\ &\qquad z_{ij} \geq x_{ij}^{(t)} \geq 0, \qquad \forall\ (i,j) \in \mathcal{A},\ t \in T, \\ &\qquad \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i \in \mathcal{N},\ t \in T, \end{aligned} \tag{19}$$

Therefore, it follows that we can write $U(z,T) = U_+(z,T) \cup U_-(z,T)$, where

$$U_+(z,T) = \{z' \in Z(T) | z' \geq z\},$$
$$U_-(z,T) = \{z' \in Z(T) | z' \leq z\},$$

and $Z(T)$ is the feasible set of problem (19) for a given $T$; i.e. if we have the subgraph $z$ at the beginning of a time interval, and we must go to a subgraph that supports multicast to $T$, then the allowable subgraphs are those that support multicast to $T$ and either increase $z$ (those in $U_+(z,T)$) or decrease $z$ (those in $U_-(z,T)$).

Note that, if we have separable constraints, then $U(z^{(m)}, T_m) \neq \emptyset$ for all $z^{(m)} \in Z$ provided that $Z(T_m) \neq \emptyset$; that is, from any feasible subgraph at stage $m$, it is possible to go to a feasible subgraph at stage $m+1$ provided that one exists for the multicast group $T_m$. But while this is the case for coded networks, it is not always the case for routed networks. Indeed, if multiple multicast trees are being used (as discussed in [51], for example), then it is definitely possible to find ourselves in a state where we cannot achieve multicast at stage $m+1$ even though static multicast to $T_m$ is possible using multiple multicast trees.

As an example of this phenomenon, consider the network depicted in Figure 5. Suppose that each arc is of unit capacity, that $s = 1$, and that, at a given time, we have $T = \{6, 8\}$. We support a multicast of rate 2 with the two trees $\{(1,3),(3,4),(4,5),(5,6),(5,7),(7,8)\}$ and $\{(1,2),(2,6),(6,8)\}$, each carrying unit rate. Now suppose that the group membership changes, and node 6 leaves while node 7 joins, so $T' = \{7, 8\}$. It is clear that static multicast to $T'$ is possible using multiple multicast trees (we simply reflect the solution for $T$), but we cannot achieve multicast to $T'$ by only adding edges to the two existing trees. Our only recourse at this stage is to abandon the existing trees and establish new ones, which causes a disruption to the

service of node 8, or to slowly reconfigure the existing trees, which causes a delay before node 7 is actually joined to the group.

Returning to the problem at hand, we see that our objective is to find a policy $\pi = \{\mu_0, \mu_1, \dots, \}$ that minimizes the cost function

$$J_\pi(z^{(0)}, T_0) = \lim_{M\to\infty} \mathbb{E}\left[\sum_{m=0}^{M-1} f(z^{(m+1)})\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}(T_m)\right],$$

where $\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}$ is the characteristic function for $2^{\mathcal{N}'} \setminus \{\emptyset\}$ (i.e. $\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}(T) = 1$ if $T \neq \emptyset$, and $\chi_{2^{\mathcal{N}'}\setminus\{\emptyset\}}(T) = 0$ if $T = \emptyset$).

We impose the assumption that we have separable constraints and that $Z(\mathcal{N}') \neq \emptyset$; that is, we assume that there exists a subgraph that supports broadcast. This assumption ensures that the constraint set $U(z, T)$ is non-empty for all $z \in Z$ and $T \subset \mathcal{N}'$. Thus, from condition (18), it follows that there exists at least one policy $\pi$ (namely, one that uses some fixed $z \in Z(\mathcal{N}')$ until the multicast group is empty) such that $J_\pi(z^{(0)}, T_0) < \infty$.

It is now not difficult to see that we are dealing with an undiscounted, infinite-horizon dynamic programming problem (see, for example, [52, Chapter 3]), and we can apply the theory developed for such problems to our problem. So doing, we first note that the optimal cost function $J^* := \min_\pi J_\pi$ satisfies Bellman's equation; namely, we have

$$J^*(z, T) = \min_{u\in U(z,T)} \{f(u) + \mathbb{E}[J^*(u, (T \setminus V) \cup W)]\}$$

if $T \neq \emptyset$, and $J^*(z, T) = 0$ if $T = \emptyset$. Moreover, the optimal cost is achieved by the stationary policy $\pi = \{\mu, \mu, \dots\}$, where $\mu$ is given by

$$\mu(z, T) = \arg\min_{u\in U(z,T)} \{f(u) + \mathbb{E}[J^*(u, (T \setminus V) \cup W)]\} \tag{20}$$

if $T \neq \emptyset$, and $\mu(z, T) = 0$ if $T = \emptyset$.

The fact that the optimal cost can be achieved by a stationary policy limits the space in which we need to search for optimal policies significantly, but we are still left with the difficulty that the state space is uncountably large; it is the space of all possible pairs $(z, T)$, which is $Z \times 2^{\mathcal{N}'}$. The size of the state space more or less eliminates the possibility of using techniques such as value iteration to obtain $J^*$.

On the other hand, given $J^*$, it does not seem at all implausible that we can compute the optimal decision at the beginning of each time interval using (20). Indeed, the constraint set

is the union of two polyhedra, which can be handled by optimizing over each separately, and, although the objective function may not necessarily be convex even if $f$ is convex owing to the term $\mathbb{E}[J^*(u, (T \setminus V) \cup W)]$, we are, at any rate, unable to obtain $J^*$ precisely on account of the large state space, and can restrict our attention to approximations that make problem (20) tractable.

For dynamic programming problems, there are many approximations that have been developed to cope with large state spaces (see, for example, [52, Section 2.3.3]). In particular, we can approximate $J^*(z, T)$ by $\tilde{J}(z, T, r)$, where $\tilde{J}(z, T, r)$ is of some fixed form, and $r$ is a parameter vector that is determined by some form of optimization, which can be performed offline if the graph $\mathcal{G}$ is static. Depending upon the approximation that is used, we may even be able to solve problem (20) using the decentralized algorithms described in Section II (or simple modifications thereof). The specific approximations $\tilde{J}(z, T, r)$ that we can use and their performance are beyond the scope of this paper.

## VI. Conclusion

Routing is certainly a satisfactory way to operate packet networks. It clearly works, but it is not clear that it should be used for all types of networks. As we have mentioned, application-layer overlay networks and multi-hop wireless networks are two types of networks where coding is a definite alternative.

To actually use coding, however, we must apply to coding the same considerations that we normally apply to routing. This paper did exactly that: We took the cost consideration from routed packet networks and applied it to coded packet networks. More specifically, we considered the problem of finding minimum-cost subgraphs to support multicast connections over coded packet networks—both wireline and wireless. As we saw, this problem is effectively decoupled from the coding problem: To establish minimum-cost multicast connections, we can first determine the rate to inject coded packets on each arc, then determine the contents of those packets.

Our work therefore brings coded packet networks one step closer to realization. But, to actually see that happen, much work remains to be done. For example, designing protocols around our algorithms is a clear task, as is designing protocols to implement coding schemes. In addition, there are some important issues coming directly from this paper that require further exploration. Some of these relate to the decentralized algorithms, e.g., their stability under changing conditions

(e.g., changing arc costs, changing graph topology), their speeds of convergence, their demands on computation and message-exchange, and their behavior under asynchronism. Another topic to explore is specific approximation methods for use in our formulation of dynamic multicast.

On a broader level, we could design other algorithms for the general problem formulation given in this paper (steps in this direction have already been taken in [53]). And we could give more thought to the cost functions themselves. Where do they come from? Do cost functions for routed packet networks make sense for coded ones? If a coded packet network is priced, how should the pricing be done? And how should the resultant cost be shared among the members of the multicast group?

In short, we believe that realizing coded packet networks is a worthwhile goal, and we see our work as an integral step toward this goal. Much promising work, requiring various expertise, remains.

## ACKNOWLEDGMENTS

The authors would like to thank R. Srikant for helpful discussions and suggestions and Hyunjoo Lee for her work on the simulation software.

## APPENDIX I

We wish to solve the following problem.

$$\text{minimize } \sum_{t\in T}(v^{(t)}-u^{(t)})^2$$

$$\text{subject to } v \in P_{ij},$$

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

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

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

$$\hat{v}^{(t)}>0 \Rightarrow (u^{(t)}-\hat{v}^{(t)}) \geq (u^{(r)}-\hat{v}^{(r)}), \qquad \forall\ r\in T \tag{21}$$

(see, for example, [33, 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 (21) 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

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

which gives

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

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

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

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

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

which can be rearranged to give

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

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

$$v^{(t)} = \begin{cases} u^{(t)} + \frac{a_{ij} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}}{\hat{k}} & \text{if } t \in \{t_1, \ldots, t_{\hat{k}}\}, \\ 0 & \text{otherwise}, \end{cases} \tag{22}$$

then $v^{(t)}$ is feasible and we see that the optimality condition (21) is satisfied. Note that, since $d \leq -u^{(t_{k+1})}$, equation (22) can also be written as

$$v^{(t)} = \max\left(0, u^{(t)} + \frac{1}{\hat{k}}\left(a_{ij} - \sum_{r=1}^{t_{\hat{k}}} u^{(r)}\right)\right). \tag{23}$$

We now turn to showing that any accumulation point of the sequence of primal iterates $\{x[n]\}$ given by (7) is an optimal solution the primal problem (3). Suppose that the dual feasible solution that the subgradient method converges to is $\bar{p}$. Then there exists some $m$ such that for $n \geq m$

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

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

$$\begin{aligned}\tilde{x}_{ij}^{(t)}[n] &= \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] + \sum_{l=m+1}^{n} \mu_l[n] x_{ij}^{(t)}[l] \\ &= \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] + \sum_{l=m+1}^{n} \frac{\mu_l[n]}{\theta[n]} (p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n] - c_{ij}[n]) \\ &= \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] + \sum_{l=m+1}^{n} \gamma_{ln} (p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n]) - \sum_{l=m+1}^{n} \gamma_{ln} c_{ij}[n].\end{aligned} \tag{24}$$

Otherwise, if $\bar{p}_{ij}^{(t)} = 0$, then from equation (23), we have

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

so

$$\begin{aligned}\tilde{x}_{ij}^{(t)}[n] \leq \sum_{l=1}^{m} \mu_l[n] x_{ij}^{(t)}[l] + \sum_{l=m+1}^{n} \gamma_{ln} (p_{ij}^{(t)}[n+1] - p_{ij}^{(t)}[n]) \\ - \sum_{l=m+1}^{n} \gamma_{ln} c_{ij}[n].\end{aligned} \tag{25}$$

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 [35] showed that, if the required conditions on the step sizes $\{\theta[n]\}$ and convex combination weights $\{\mu_l[n]\}$ are satisfied, then

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

as $k \to \infty$; hence we see from equations (24) and (25) that, for $k$ sufficiently large,

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

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

## APPENDIX II
## PROOF OF PROPOSITION 1

We prove the stability of the primal-dual algorithm by using the theory of Lyapunov stability (see, for example, [42, Section 3.10]). This proof is based on the proof of Theorem 3.7 of [42].

The Lagrangian for problem (10) is as follows:

$$L(x,p,\lambda) = U(x) - \sum_{t\in T}\left\{\sum_{i\in\mathcal{N}} p_i^{(t)}\left(\sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} - \sigma_i^{(t)}\right) - \sum_{(i,j)\in\mathcal{A}} \lambda_{ij}^{(t)} x_{ij}^{(t)}\right\}. \tag{26}$$

The function $U$ is strictly concave since $f_{ij}$ is a monotonically increasing, strictly convex function and $z'_{ij}$ is a strictly convex function of $x_{ij}$, so there exists a unique minimizing solution for problem (10), say $\hat{x}$, and Lagrange multipliers, say $\hat{p}$ and $\hat{\lambda}$, which satisfy the following Karush-Kuhn-Tucker conditions.

$$\frac{\partial L(\hat{x},\hat{p},\hat{\lambda})}{\partial x_{ij}^{(t)}} = \left(\frac{\partial U(\hat{x})}{\partial x_{ij}^{(t)}} - \left(\hat{p}_i^{(t)} - \hat{p}_j^{(t)}\right) + \hat{\lambda}_{ij}^{(t)}\right) = 0, \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T, \tag{27}$$

$$\sum_{\{j|(i,j)\in\mathcal{A}\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} \hat{x}_{ji}^{(t)} = \sigma_i^{(t)}, \qquad \forall\ i\in\mathcal{N},\ t\in T, \tag{28}$$

$$\hat{x}_{ij}^{(t)} \geq 0 \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T, \tag{29}$$

$$\hat{\lambda}_{ij}^{(t)} \geq 0 \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T, \tag{30}$$

$$\hat{\lambda}_{ij}^{(t)}\hat{x}_{ij} = 0 \qquad \forall\ (i,j)\in\mathcal{A},\ t\in T. \tag{31}$$

From Equation (26), 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:

$$V(x,p,\lambda) = \sum_{t\in T}\left\{\sum_{(i,j)\in\mathcal{A}}\left(\int_{\hat{x}_{ij}^{(t)}}^{x_{ij}^{(t)}} \frac{1}{k_{ij}^{(t)}(\sigma)}(\sigma - \hat{x}_{ij}^{(t)})d\sigma + \int_{\hat{\lambda}_{ij}^{(t)}}^{\lambda_{ij}^{(t)}} \frac{1}{m_{ij}^{(t)}(\gamma)}(\gamma - \hat{\lambda}_{ij}^{(t)})d\gamma\right)\right.$$
$$\left. + \sum_{i\in\mathcal{N}}\int_{\hat{p}_i^{(t)}}^{p_i^{(t)}} \frac{1}{h_i^{(t)}(\beta)}(\beta - \hat{p}_i^{(t)})d\beta\right\}.$$

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,

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

Note that

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

since the inequality is an equality if either $x_{ij}^{(t)} \le 0$ or $\lambda_{ij}^{(t)} \ge 0$; 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 $\hat{\lambda}_{ij}^{(t)} \ge 0$, $-x_{ij}^{(t)}(\lambda_{ij}^{(t)}-\hat{\lambda}_{ij}^{(t)}) \ge 0$. Therefore,

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

where the last line follows from Karush-Kuhn-Tucker conditions (27)–(31) and the fact that

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

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

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

## APPENDIX III
## PROOF OF PROPOSITION 2

Suppose $(x, z)$ is a feasible solution to problem (16). Then, for all $(i, j) \in \mathcal{A}'$ and $t \in T$,

$$\begin{aligned}
\sum_{m=m(i,j)}^{M_i} z_{iJ_m^{(i)}} &\geq \sum_{m=m(i,j)}^{M_i} \sum_{k \in J_m^{(i)}} x^{(t)}_{iJ_m^{(i)}k} \\
&= \sum_{k \in J_{M_i}^{(i)}} \sum_{m=\max(m(i,j),m(i,k))}^{M_i} x^{(t)}_{iJ_m^{(i)}k} \\
&\geq \sum_{k \in J_{M_i}^{(i)} \setminus J_{m(i,j)-1}^{(i)}} \sum_{m=\max(m(i,j),m(i,k))}^{M_i} x^{(t)}_{iJ_m^{(i)}k} \\
&= \sum_{k \in J_{M_i}^{(i)} \setminus J_{m(i,j)-1}^{(i)}} \sum_{m=m(i,k)}^{M_i} x^{(t)}_{iJ_m^{(i)}k} \\
&= \sum_{k \in J_{M_i}^{(i)} \setminus J_{m(i,j)-1}^{(i)}} \hat{x}^{(t)}_{ik}.
\end{aligned}$$

Hence $(\hat{x}, z)$ is a feasible solution of problem (17) with the same cost.

Now suppose $(\hat{x}, z)$ is an optimal solution of problem (17). Since $f_{iJ_1^{(i)}}(\zeta) < f_{iJ_2^{(i)}}(\zeta) < \cdots < f_{iJ_{M_i}^{(i)}}(\zeta)$ for all $\zeta \geq 0$ and $i \in \mathcal{N}$ by assumption, it follows that, for all $i \in \mathcal{N}$, the sequence $z_{iJ_1^{(i)}}, z_{iJ_2^{(i)}}, \ldots, z_{iJ_{M_i}^{(i)}}$ is given recursively, starting from $m = M_i$, by

$$z_{iJ_m^{(i)}} = \max_{t \in T} \left\{ \sum_{k \in J_{M_i}^{(i)} \setminus J_{m-1}^{(i)}} \hat{x}^{(t)}_{ik} - \sum_{l=m+1}^{M_i} z_{iJ_l^{(i)}} \right\}.$$

Hence $z_{iJ_m^{(i)}} \geq 0$ for all $i \in \mathcal{N}$ and $m = 1, 2, \ldots, M_i$. We then set, starting from $m = M_i$ and $j \in J_{M_i}^{(i)}$,

$$x^{(t)}_{iJ_m^{(i)}j} := \min \left( \hat{x}^{(t)}_{ij} - \sum_{l=m+1}^{M_i} x_{iJ_l^{(i)}j}, z_{iJ_m^{(i)}} - \sum_{k \in J_{M_i}^{(i)} \setminus J_{m(i,j)}^{(i)}} x^{(t)}_{iJ_m^{(i)}k} \right).$$

It is now difficult to see that $(x, z)$ is a feasible solution of problem (16) with the same cost.

Therefore, the optimal costs of problems (16) and (17) are the same and, since the objective functions for the two problems are the same, $z$ is part of an optimal solution for problem (16) if and only if it is part of an optimal solution for problem (17).

# REFERENCES


[1] 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.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

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

[4] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[5] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://web.mit.edu/trace/www/itrandom-revision.pdf

[6] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://arxiv.org/abs/cs.IT/0510070

[7] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. 31, no. 3, pp. 343–351, Mar. 1983.

[8] B. M. Waxman, "Routing of multicast connections," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[9] P. Winter, "Steiner problem in networks: A survey," *Networks*, vol. 17, pp. 129–167, 1987.

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

[11] M. Charikar, C. Chekuri, T.-y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," *J. Algorithms*, vol. 33, no. 1, pp. 73–91, Oct. 1999.

[12] 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.

[13] 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.

[14] M. Imase and B. M. Waxman, "Dynamic Steiner tree problem," *SIAM J. Disc. Math.*, vol. 4, no. 3, pp. 369–384, Aug. 1991.

[15] J. Westbrook and D. C. K. Yan, "Greedy algorithms for the on-line Steiner tree and generalized Steiner problems," in *Proc. Third Workshop on Algorithms and Data Structures (WADS '93)*, 1993, pp. 622–633.

[16] L. Song, R. W. Yeung, and N. Cai, "Zero-error network coding for acyclic networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3129–3139, Dec. 2003.

[17] M. Médard, M. Effros, D. Karger, and T. Ho, "On coding for non-multicast networks," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[18] A. Rasala Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[19] S. Riis, "Linear versus non-linear Boolean functions in network flow," in *Proc. 2004 Conference on Information Sciences and Systems (CISS 2004)*, Mar. 2004.

[20] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[21] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 412–422, Mar. 1995.

[22] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal distributed multicast routing using network coding: Theory and applications," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 2, pp. 47–49, 2004.

[23] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proc. 2004 International Symposium on Information Theory and its Applications (ISITA 2004)*, Oct. 2004, pp. 1232–1237.

[24] Z. Li, B. Li, D. Jiang, and L. C. Lau, "On achieving optimal throughput with network coding," in *Proc. IEEE Infocom 2005*, Miami, FL, Mar. 2005.

[25] Z. Li and B. Li, "Efficient and distributed computation of maximum multicast rates," in *Proc. IEEE Infocom 2005*, Miami, FL, Mar. 2005.

[26] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1906–1918, Nov. 2005.

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

[28] 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.

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

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

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

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

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

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

[35] 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.

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

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

[38] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 274–285, Apr. 2004.

[39] A. Ouorou, P. Mahey, and J.-P. Vial, "A survey of algorithms for convex multicommodity flow problems," *Manage. Sci.*, vol. 46, no. 1, pp. 126–147, Jan. 2000.

[40] D. P. Bertsekas, "A class of optimal routing algorithms for communication networks," in *Proc. 5th International Conference on Computers and Communication (ICCC '80)*, Oct. 1980, pp. 71–76.

[41] D. P. Bertsekas, E. M. Gafni, and R. G. Gallager, "Second derivative algorithms for minimum delay distributed routing in networks," *IEEE Trans. Commun.*, vol. 32, no. 8, pp. 911–919, Aug. 1984.

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

[43] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proc. IEEE Infocom 2003*, vol. 1, Mar.–Apr. 2003, pp. 702–711.

[44] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MobiCom '03: Proc. 9th Annual International Conference on Mobile Computing and Networking*, 2003, pp. 66–80.

[45] M. Johansson, L. Xiao, and S. Boyd, "Simultaneous routing and power allocation in CDMA wireless data networks," in *Proc. 2003 IEEE International Conference on Communications (ICC 2003)*, vol. 1, May 2003, pp. 51–55.

[46] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136–1144, July 2004.

[47] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels," *IEEE/ACM Trans. Networking*, vol. 13, no. 4, pp. 868–880, Aug. 2005.

[48] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, "Network planning in wireless ad hoc networks: A cross-layer approach," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 136–150, Jan. 2005.

[49] 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.

[50] 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.

[51] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *Proc. 2004 IEEE International Symposium on Information Theory (ISIT 2004)*, Chicago, IL, June–July 2004.

[52] D. P. Berksekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2001, vol. 2.

[53] Y. Xi and E. M. Yeh, "Distributed algorithms for minimum cost multicast with network coding," in *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005.

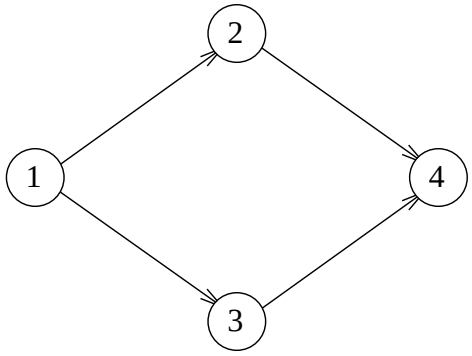



Fig. 4. A four node network.

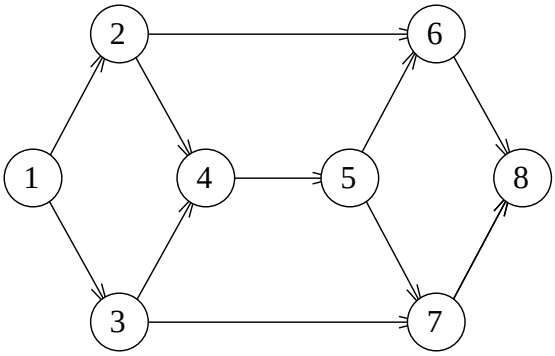



Fig. 5. A network used for dynamic multicast.