

Min-cost Selfish Multicast with Network Coding

Sandeep Bhadra *Student Member, IEEE*, Sanjay Shakkottai *Member, IEEE*, and Piyush Gupta *Member, IEEE*

Abstract—The single-source min-cost multicast problem, which can be framed as a convex optimization problem with the use of network codes and convex increasing edge costs is considered. A decentralized approach to this problem is presented by Lun, Ratnakar et. al. for the case where all users cooperate to reach the global minimum. Further, the cost for the scenario where each of the multicast receivers greedily routes its flows is analyzed and the existence of a Nash equilibrium is proved.

An allocation rule by which edge cost at each edge is allocated to flows through that edge is presented. We prove that under our pricing rule, the flow cost at user equilibrium is the same as the min-cost. This leads to the construction of a selfish flow-steering algorithm for each receiver, which is also globally optimal. Further, the algorithm is extended for completely distributed flow adaptation at nodes in the network to achieve globally minimal cost in steady state. Analogous results are also presented for the case of multiple multicast sessions.

Index Terms—network coding, minimum cost multicast, game theory, Nash equilibrium, convex optimization

I. INTRODUCTION

The single-source multicast problem for network coding has received much attention in recent years due to the tractability of designing optimal linear network codes for this case. Ahlswede, et. al. in [2] prove that for networks where the min-cut max-flow rate cannot be achieved by simple forwarding of packets, coding incoming packets at intermediate routers (network-coding) can help achieve the max-flow min-cut rate for such networks. Further, Ho et al. [3], [4] suggest the use of Random Linear Codes (RLCs) that can achieve the above linear network code rate asymptotically in the size of the symbol alphabet used for encoding/decoding. Since the intermediate routers can code randomly independent of other routers in the network, RLCs offer the means for decentralized design of network codes and form the basis for practical network coding schemes [5].

The problem of finding the minimum-cost multicast tree for networks has been studied extensively. For a general directed graph with a cost function at each edge, a specified root (source) and a subset of the nodes (receivers), the problem of finding a minimum-cost arborescence rooted at the source and spanning all the receivers is called the Directed Steiner

Tree (DST) problem. Approximation algorithms for the DST, which is known to be NP-hard, has received considerable attention in recent years. Charikar et al. [6] present a non-trivial $O(i(i-1)k^{1/i})$ algorithm in $O(n^i k^2 i)$ time where k is the number of receivers. An LP-relaxation of the problem leads Zosin and Khuller [7] to a $O(D+1)$ -approximation for the special case when the subgraph induced by the non-terminal nodes is a tree of depth D .

A decentralized but cooperative scheme has been suggested by Lun et al. in [1] where the authors solve the network-coding min-cost optimization from [8] using primal-dual methods by message passing between intermediate routers. However, this scheme requires a separate (differential-equation based) controller at each intermediate router for every flow passing through it. Further, many current models of heterogeneous network service provisioning assume that selfish routing decisions are made by end-users based on the price of the links [9], [10]. Such scenarios are likely to emerge with ad hoc or sensor networking where each end node is attached to a single multicast sink and therefore seeks to minimize its own cost. The dual problem of maximizing utility in a congestion game over a packet-forwarding network is considered in [9], [11]. Recently, the authors in [12] have framed this congestion control problem for network coding for single- and multiple-source multicasts as a generalization of the Eisenberg-Gale convex program to compute market equilibrium in the presence of economies of scale. Further, the primal-dual algorithm in [1] requires computationally intensive steps to be performed at each intermediate router.

In this paper, we seek to design a min-cost flow-allocation algorithm when users are non-cooperative and minimize computation performed at each intermediate router. The users are assumed to be selfish agents that play a non-cooperative game to minimize personal costs selfishly without regard to the global or social optimal, and the expectation is that these users reach a Nash equilibrium if one exists. It is well-known that Nash equilibria do not optimize social welfare in general - a classical example of such an inefficient equilibrium is the ‘Prisoner’s Dilemma’ [13]. Thus it immediately becomes important to quantify the inefficiency inherent in a selfish solution - dubbed the ‘price of anarchy’ [14], [15].

The unicast selfish-agent min-cost routing problem is a classical problem in transportation literature and has been discussed in [16], [17], which corresponds to the uncoded packet forwarding scenario. Recently, [14], [18], [19] calculated the price of anarchy for this problem for a variety of convex cost functions for the capacitated and uncapacitated links. However, the optimization problem for the multicast min-cost flow with network coding as shown in the following section departs significantly from the min-cost unicast flow problem for uncoded packets and thus motivates independent

S. Bhadra and S. Shakkottai are with the Wireless Networking and Communications Group, Department of Electrical and Computer Engineering, The University of Texas at Austin, {bhadra, shakkottai}@ece.utexas.edu. P. Gupta is with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, pgupta@research.bell-labs.com. The research of S. Bhadra and S. Shakkottai was supported by NSF Grants ACI-0305644, CNS-0325788 and CNS-0347400, and that of P. Gupta was supported in part by NSF Grants CCR-0325673 and CNS-0519535. Early results leading to this paper were presented as an invited talk at the Workshop on Mathematical Modeling and Analysis of Computer Networks, Waterloo, Canada, May 2005 and in the Second Workshop on Network Coding, Theory and Applications (NETCOD), Boston, MA, April 2006.

analysis.

A. Main Contributions

In this paper, we consider the min-cost flow routing problem with network coding for the selfish-agent case. We first consider the case with a single source and T multicast sinks (receivers), with each sink requiring a total rate R . We study the case where the network supports multi-path routing. A flow (along a particular path) from the source to a sink accumulates a cost that depends on the flow rate as well as the congestion on each of the links the flow traverses. Each sink t “steers” the flow rate allocation among its paths (i.e., among all paths from the source to the selected sink t such that the sum rate across paths is R) such that its *total cost* is minimized (in other words, a “greedy” setup for each sink). We then generalize this framework to consider a multiple-sources scenario. The main contributions are as follows.

- (i) We present the min-cost optimization problem for the single-source multicast with network coding and derive an asymptotically accurate approximation to that problem in Section II. The selfish routing scenario is presented in Section III where a market is defined for bandwidth, being sold by links (sellers), that is utilized by flows to individual sinks (buyers).

We develop a mechanism for links (sellers) to allocate the link-costs among users of the link and demonstrate that for *monomial edge cost functions* (see section III), a Nash equilibrium exists, and that the flow allocation at Nash equilibrium corresponds to the min-cost flow. Further, we show that capacitated links (i.e, links with capacity constraints) can be approximated arbitrarily closely using edge cost functions in the monomial class described in Section III.

In other words, we show that the mechanism that we develop for link pricing leads to a rate allocation among users such that “greedy” flow rate allocation by each sink leads to a globally optimal flow rate allocation that minimizes the *total cost* in the network. In terms of algorithmic game-theoretic literature, this means that the ‘price of anarchy’ [14], [15] for the considered “greedy” system is 1.

- (ii) In Section IV, we consider the multiple-source multicast problem and demonstrate a sub-optimal greedy scheme to achieve min-cost by selectively network coding within individual multicast sessions and not across sessions. (We note in passing that the general multi-source multi-destination network coding problem is intractable (NP hard) [20].)
- (iii) Next, in Section V we present *UESSM*, User Equilibrium with Single Source Multicast, a non-cooperative decentralized flow-steering algorithm that provably converges arbitrarily close to a min-cost flow allocation for the class of convex, monomial edge cost functions defined in Section III. At each receiver, UESSM “steers” flows across the paths leading to it in order to greedily minimize its own cost. This allows us to achieve the min-cost flow with network coding, without having to maintain state or

perform per-flow primal-dual type calculations at every intermediate router. All that links have to do in UESSM is to allocate link costs according to the rule developed in subsection III-A.

- (iv) We next develop the Local Distributed Selfish Routing Algorithm (*LDSRA*) for min-cost routing. This algorithm is a local distributed algorithm where nodes in the network adjust flow fractions based on the local flow and cost information at each node and its neighbors. This is an analog of the Bellman-Ford algorithm, however, in the context of network coding. By using the end-to-end delay experienced by a probe packet as the marginal cost, LDSRA minimizes the total network latency (sum cost) by reallocating flows from the more expensive (greater delay) neighbor toward a cheaper (lower delay) one. We finally present simulation results for both UESSM and LDSRA to illustrate convergence properties.

II. GLOBAL EQUILIBRIUM

Consider a directed graph $G = (N, A)$ that models the network with the set of nodes N and the set of directed edges between them A . We consider a multicast session of rate R from source $s \in N$ to each of the sinks $t \in T, T \subseteq N$ implemented via multipath routing along the directed graph (network model). Flows along the set of paths \mathcal{P}_t from s to t are indexed as $f_P \in \mathbb{R}$ for all $P \in \mathcal{P}_t$; $\mathcal{P} = \cup_{t \in T} \mathcal{P}_t$ is the set of all possible paths. Note that an edge may carry two or more flows to the same sink due to the presence of multipath routing.

We will associate a cost with the flow through each link on the network and formulate a global min-cost problem as one that minimizes the sum cost over the network. Accordingly, let $c_e(\cdot)$ be the cost function corresponding to edge $e \in A$ taking as argument a variable x dependent on the flows through edge e . We assume that the function $c_e(x)$ is strictly convex, positive, differentiable and monotonically increasing in variable x , with $c_e(0) = 0$. Further, we define the edge marginal cost $m_e(x) = c_e(x)/x$. We assume that the marginal cost is continuous and strictly increasing, with $m_e(0) = 0$.

Under traditional packet forwarding where packets are treated as objects, the cost of operating an edge in the graph is a function of the load of all packets that traverse that edge to all sinks. Most of the classical work on network optimization therefore deals with cost functions that take as argument the total fluid flow of packets to all sinks passing through that edge. That is, $x = \sum_{t \in T} \sum_{P \in \mathcal{P}_t} f_P$ and the corresponding edge cost incurred is $c_e(\sum_{t \in T} \sum_{P \in \mathcal{P}_t} f_P)$.

However, since we consider the case where intermediate routers perform random linear coding across packets to different sinks, it can be shown that the cost function $c_e(\cdot)$ takes as argument $x = \max_{t \in T} \sum_{P \in \mathcal{P}_t} f_P$ [8]. To see this intuitively, observe that under Random Linear Coding (RLC), packets to different sinks are linearly combined by the router to form a coded packet. In the fluid sense therefore, RLC allows for flows to different sinks to ‘merge’ to form the coded flow. This implies that on any edge, the effective size of the coded packet stream is dominated by the largest among net flows to each sink that traverse the edge.

Formally, the optimal cost for a rate R multicast connection from a single source $s \in N$ to sink nodes $T \subset N$ is given by the solution to the following optimization problem similar to [1], [8],

$$\begin{aligned} \text{GLOBAL}(G, c, R) \\ \text{minimize } C(f) &= \sum_{e \in A} c_e(z_e) \\ \text{subject to } z_e &= \max_t \left\{ \sum_{P \in \mathcal{P}_t: e \in P} f_P \right\} \quad \forall e \in A \\ f_P &\geq 0 \quad \forall P \in \mathcal{P} \\ \sum_{P \in \mathcal{P}_t} f_P &= R \quad \forall t \in T. \end{aligned}$$

However, since $\max\{\dots\}$ is not differentiable everywhere, motivated by the approach in [21], [1], we use the \mathcal{L}_n -approximation

$$\max\{x_1, x_2, \dots, x_k\} = \lim_{n \rightarrow \infty} \left(\sum_{i=1}^k x_i^n \right)^{1/n}$$

for analysis, thereby avoiding sub-gradient methods. Following the approximation of the $\max()$ above the \mathcal{L}_n -relaxed cost function of $\text{GLOBAL}(G, c, R)$,

$$C_n(f) = \sum_{e \in A} c_e \left(\left[\sum_{t \in T} \left(\sum_{P \in \mathcal{P}_t: e \in P} f_P \right)^n \right]^{1/n} \right)$$

is differentiable everywhere. Formally, let C_n^* be the optimal solution to \mathcal{L}_n - $\text{GLOBAL}(G, c, R)$ and C^* be the optimal solution to $\text{GLOBAL}(G, c, R)$.

We note that this \mathcal{L}_n -approximation is motivated by the fact that as $n \rightarrow \infty$, $|C^* - C_n^*| \rightarrow 0$. Later (in Section III) we will derive bounds on the approximation error for finite n for the class of functions considered in Section III (cf. Remark 3.1).

Since the cost functions are convex and the constraints form a convex set, the first-order Karush-Kuhn-Tucker conditions [22] are necessary and sufficient to solve \mathcal{L}_n - $\text{GLOBAL}(G, c, R)$. We summarize the results in the following lemma.

Let $z_e^{(n)}$ be the corresponding \mathcal{L}_n relaxation of z_e defined as

$$z_e^{(n)} \triangleq \left(\sum_{t \in T} x_{e,t}^n \right)^{1/n}$$

where

$$x_{e,j} \triangleq \sum_{P \in \mathcal{P}_j: e \in P} f_P$$

is the total flow of type j through the edge e .

Lemma 2.1: A network coded multicast flow f^* is optimal for \mathcal{L}_n - $\text{GLOBAL}(G, c, R)$ if and only if for all $t \in T$, and any paths $P_1, P_2 \in \mathcal{P}_t$ with strictly positive flows $f_{P_1}^*, f_{P_2}^* > 0$

$$\sum_{e \in P_1} c'_e(z_e^{(n)*}) \alpha_{e,j}^{(n)*} = \sum_{e \in P_2} c'_e(z_e^{(n)*}) \alpha_{e,j}^{(n)*}, \quad (1)$$

for,

$$\alpha_{e,j}^{(n)} \triangleq \frac{z_e^{(n)}}{x_{e,j}} \cdot \frac{1}{\sum_{t \in T} \left(\frac{x_{e,t}}{x_{e,j}} \right)^n}. \quad (2)$$

Proof: We append the cost function with the linear constraints via the Lagrangian multipliers λ_t, μ_P to form the Lagrangian

$$L(f, \lambda, \mu) = C_n(f) + \sum_{t \in T} \lambda_t \left(\sum_{P \in \mathcal{P}_t} f_P - R \right) - \sum_{P \in \mathcal{P}} \mu_P f_P.$$

We differentiate the Lagrangian with respect to each flow f_P , and the Lagrangian multipliers and equate each partial differential to zero to form a set of simultaneous equations in f, λ and μ . Solving these equations yields a minimizing solution f^*, λ^*, μ^* . Note that for all $P \in \mathcal{P}$, f_P^* and μ_P^* are complementary slack, i.e. $f_P^* \mu_P^* = 0$ with $\mu_P^* \geq 0$. Hence for paths with strictly positive flow, differentiating $L(f, \lambda, \mu)$ with respect to a particular flow f_{P_1} , for $P_1 \in \mathcal{P}_j$, gives

$$\sum_{e \in P_1} c'_e(z_e^{(n)}) \left(\frac{\sum_{P \in \mathcal{P}_j: e \in P} f_P}{z_e^{(n)}} \right)^{n-1} + \lambda_j = 0,$$

where $c'_e(x) = \frac{\partial c_e(x)}{\partial x}$. This implies that $\forall P_1, P_2 \in \mathcal{P}_j$ with $f_{P_1}^* > 0$,

$$\begin{aligned} \sum_{e \in P_1} c'_e(z_e^{(n)*}) \left(\frac{\sum_{P \in \mathcal{P}_j: e \in P} f_P^*}{z_e^{(n)*}} \right)^{n-1} \\ \leq \sum_{e \in P_2} c'_e(z_e^{(n)*}) \left(\frac{\sum_{P \in \mathcal{P}_j: e \in P} f_P^*}{z_e^{(n)*}} \right)^{n-1}. \end{aligned} \quad (3)$$

We are now done. ■

We note in passing that the behavior of

$$\alpha_{e,j} \triangleq \lim_{n \rightarrow \infty} \alpha_{e,j}^{(n)} \quad (4)$$

is not immediately clear – we cannot immediately state if the limit even exists. However, for any $n \in \mathbb{N}$ and any edge e with positive flows, we have from (2) that

$$\begin{aligned} \sum_{t \in T} \alpha_{e,t}^{(n)} &= \left[\frac{z_e^{(n-1)}}{z_e^{(n)}} \right]^{n-1} \\ &\geq 1 \end{aligned} \quad (5)$$

where the last inequality follows since the continuous function $L(p) = (\sum_i x_i^p)^{1/p}$ can be seen to be monotone decreasing in p for all $p \geq 1$ when all $x_i \geq 0$. Further, from Hölder's inequality,

$$\begin{aligned} \sum_{t \in T} x_{e,t}^{n-1} \cdot 1 &\leq \left(\sum_{t \in T} x_{e,t}^n \right)^{\frac{n-1}{n}} \left(\sum_{t \in T} 1 \right)^{\frac{1}{n}} \\ &= (z_e^{(n)})^{n-1} |T|^{1/n}. \end{aligned}$$

Using the definition of $\alpha_{e,t}^{(n)}$ from (2) and the above inequality,

$$\begin{aligned} \sum_{t \in T} \alpha_{e,t}^{(n)} &= \frac{\sum_{t \in T} x_{e,t}^{n-1}}{(z_e^{(n)})^{n-1}} \\ &\leq |T|^{1/n} \end{aligned} \quad (6)$$

for every value of n . Then, from (6) and (5), it follows that

$$\lim_{n \rightarrow \infty} \sum_{t \in T} \alpha_{e,t}^{(n)} = 1.$$

III. SELFISH ROUTING AND EQUILIBRIUM

The solution to GLOBAL finds the optimum flow that minimizes routing cost in the overall network cost. This section deals with the system under the condition that each receiver minimizes its own cost to achieve user equilibrium under a defined bandwidth market to model selfish behavior as shown below. The ultimate goal of this section (and the next, respectively), is to show that under certain conditions on c_e , the user equilibrium corresponds to the global equilibrium (is comparable to the global equilibrium of a related optimization, respectively). These results will motivate a user-equilibrium based distributed optimization algorithm, discussed in Section V.

A. The bandwidth market and link price-allocation

Each edge $e \in A$ sells bandwidth to the receivers (sinks) which are the users. Note that in the solution to the global problem we were merely concerned with the effective cost of the edge $c_e(z_e^{(n)})$ and did not need to consider how the cost of an edge in the network is divided among the flows through that network, while this sharing of costs (price allocation) needs to be defined for the user costs.

Hence, we propose a price allocation rule at each link and subsequently show that under this protocol, the sum cost under user equilibrium is equal to the min-cost for a wide range of cost functions c_e . Our price allocation rule is as follows. For each edge e the total cost of the flows $c_e(z_e^{(n)})$ is divided among flows of all type $t \in T$ so that $\frac{x_{e,j}^n}{\sum_{t \in T} x_{e,t}^n}$ fraction of the edge cost is borne by the flows in $f_P, P \in \mathcal{P}_j$ of type j . In turn $x_{e,j}$ is divided among all flows of type j through edge e in the ratio $f_P/x_{e,j}$ for all $P \in \mathcal{P}_j$. Thus the marginal cost of a flow f_P through a path $P \in \mathcal{P}_j, j \in T$

$$d_P^{(n)}(f) \triangleq \sum_{e \in P} c_e(z_e^{(n)}) \frac{1}{x_{e,j}} \frac{x_{e,j}^n}{\sum_{t \in T} x_{e,t}^n}. \quad (7)$$

Observe that by simply multiplying and dividing by $z_e^{(n)}$, (7) can be written as

$$d_P^{(n)}(f) = \sum_{e \in P} \frac{c_e(z_e^{(n)})}{z_e^{(n)}} \alpha_{e,j}^{(n)},$$

where $\alpha_{e,j}^{(n)}$ is as defined in (2).

B. User costs and equilibrium

Under the selfish condition, each flow from source s to destination j tries to minimize its marginal cost. This corresponds to each receiver minimizing its own total cost selfishly.

Since the cost functions are continuous and differentiable everywhere, we define *user equilibrium* as follows,

Definition 3.1: A user equilibrium is a flow allocation f feasible in \mathcal{L}_n -GLOBAL(G, c, R) such that for any $P_1, P_2 \in \mathcal{P}_t$ where $f_{P_1} > 0$,

$$d_{P_1}^{(n)}(f) \leq d_{P_2}^{(n)}(f). \quad (8)$$

Note that this version of user equilibrium is also referred to as a *local Nash equilibrium* or *Wardrop equilibrium* in existing literature [16], [19].

Corresponding to this equilibrium, the total system cost for the flow f at Nash equilibrium is then

$$C_n(f) \triangleq \sum_{P \in \mathcal{P}} d_P^{(n)}(f) f_P.$$

In other words, any small $\epsilon \rightarrow 0$ change to the flow allocations from path P_1 to P_2 will only increase the sum cost along the paths in \mathcal{P}_t for sink t . The notion of a local Nash equilibrium can be practically justified in scenarios where end users are in a distributed setting, with no or partial knowledge of the system, and try to reach their own local selfish optima by making small modifications to the flow allocations across paths in \mathcal{P}_t , where the flow steering proceeds only if that provides the selfish agent with immediate cost reduction.

C. User equilibrium vs. Global optimum

The similarity between the conditions in Lemma 2.1 and Definition 3.1 have been noticed for the case of costs depending on sum flows through an edge by Dafermos and Sparrow [16] and Beckman [17] and is cited by [14]. An important difference in our case is that while the edge cost in [14], [16], [17] is proportionally divided among all the flows through it, here, the cost is mainly borne by the sink with the maximum flow through the edge. The following lemma (adapted from [16], [17]) allows us to formulate the Nash equilibrium condition for a particular set of edge cost functions in terms of a global optimum for the same graph over a *different set of edge-cost functions*.

Lemma 3.1 ([16], [17]): A single-source multicast flow f solves \mathcal{L}_n -GLOBAL(G, c, R) if and only if it is in local Nash equilibrium for \mathcal{L}_n -GLOBAL($G, c'(x)x, R$). Further, a local Nash equilibrium flow f exists for \mathcal{L}_n -GLOBAL(G, c, R). Moreover, if f and \tilde{f} are feasible flows at Nash equilibrium, then $C_n(f) = C_n(\tilde{f})$.

Proof: Comparing the KKT conditions from Lemma 2.1 with the user equilibrium conditions from Definition 3.1 leads us directly to the first statement of the Lemma. For the second statement, note that solving for a flow in local Nash equilibrium for \mathcal{L}_n -GLOBAL(G, c, R) corresponds to finding a (local) optimum flow for \mathcal{L}_n -GLOBAL(G, h, R), where $h_e(x) = \int_0^x c_e(t)/t dt$. Since c_e is continuous and monotonically increasing, h_e is strictly convex. Consequently, \mathcal{L}_n -GLOBAL(G, h, R) is a convex optimization over a convex set which implies that the optimum cost is unique, even though the solution points (the local minima) are not necessarily unique. ■

This ensures that there exists a flow allocation that satisfies the user equilibrium (8).

We can now present the analog of the main result in Roughgarden and Tardos [14] for the min-cost multicast problem with network coding in the following theorem.

Theorem 3.1: If for an instance $\mathcal{L}_n - (G, c, R)$ the cost function at each edge e is of the monomial form $c_e(z_e^{(n)}) = a_e(z_e^{(n)})^{k+1}$ for any fixed $k \in \mathbb{R}$, $k > 0$, then for all $n \in \mathbb{N}$, the cost of flow f at local Nash equilibrium $C_n(f)$ equals the cost $C_n(f^*)$ of the global min-cost flow f^* .

Proof: Since $c_e(z_e^{(n)})/z_e^{(n)} = a_e(z_e^{(n)})^k$ is monotonic increasing in $z_e^{(n)}$ for $k > 0$, we know from Lemma 3.1 that a Nash equilibrium exists. Further, note that for the given class of cost functions, the Nash condition (8) is the same as the KKT condition in (1). Hence, a Nash flow is also an optimum flow for the instance $\mathcal{L}_n - (G, c, R)$ and thus the cost functions are the same. ■

We note that notwithstanding the simplicity of the proof, the above result is significant due to its application in Section V. The result above implies that for a large class of edge cost functions, a global min-cost multicast with network coding can be achieved by merely steering flows across edges to achieve user equilibrium corresponding to each sink t . In other words, the *price of anarchy* is 1.

Note that in general, the global min-cost flow can be achieved if each link charges the ‘‘Lagrangian cost’’ $h_e(x) = \int_0^x c_e(t)/t dt$ instead of the true cost $c_e(x)$. However, this would imply that the seller (link) earns an amount disproportionate to the true value of the goods or services (bandwidth) sold. The link-price allocation scheme detailed in subsection III-A ensures that the seller receives the ‘fair’ cost $c_e(x)$ but charges the selfish users differently so as to ensure that user equilibrium coincides with the socially-optimal flow allocation.

Observe that at $n = 1$, the \mathcal{L}_1 -GLOBAL(G, c, R) problem is the same as the classical min-cost flow-allocation problem. Also, correspondingly, our price allocation reduces to the allocation of link cost to a sink in linear proportion to the magnitude of flow to that sink through the particular link – thereby making the marginal cost of every flow through a link the same. This is exactly the same as the anarchic scenario in [14] where each flow through a particular edge e has the same marginal cost (edge delay) l_e and the net cost of that edge $c_e = l_e \sum_{e \in P, P \in \mathcal{P}} f_P$.

In general, the results herein define a differentiated pricing scheme for a shared service whose cost depends not on the sum of the demands but on the max demand. At the limit $n \rightarrow \infty$, we observe that only the set of users $T' = \arg \max_{t \in T} \sum_{P \in \mathcal{P}_t: e \in P} f_P$ pay for the cost of the link. Our price allocation rule automatically induces separate selfish agents to collaborate to benefit from this economy of scale.

Remark 3.1: We now revisit the issue of the approximation error resulting from the \mathcal{L}_n -relaxation of GLOBAL(G, c, R). Recall that in Section II, the approximation was motivated by the fact that the error in the optimal cost (for any convex, increasing, differentiable link cost function) approaches 0 as $n \rightarrow \infty$. In this remark, we strengthen this statement by deriving bounds on the approximation error for finite values of n for

the class of functions of the form $c_e(x) = a_e x^{k+1}$ considered in Theorem 3.1. For any given $\delta > 0$, we compute an $n(\delta)$ such that for any $n > n(\delta)$, the fractional approximation error (i.e., percentage error) $|C^* - C_n^*|/C^* \leq \delta$ is satisfied.

Let f^* be a solution to GLOBAL(G, c, R) and f_n^* be a solution to \mathcal{L}_n -GLOBAL(G, c, R). Further, let an optimal sum flow through edge e in the unrelaxed GLOBAL problem be denoted by $x_{e,t}^* \triangleq \sum_{P \in \mathcal{P}_t: e \in P} f_P^*$.

Observe that for any vector $(x_{e,t})_{t \in T}$ of size $|T|$, if $z_e \triangleq \max_{t \in T} x_{e,t}$ and $z_e^{(n)} \triangleq (\sum_{t \in T} x_{e,t}^n)^{1/n}$, then we can bound the difference

$$z_e^{(n)} - z_e \leq (|T|^{1/n} - 1)z_e.$$

Since f^* is not necessarily an optimal flow allocation for \mathcal{L}_n -GLOBAL(G, c, R),

$$\begin{aligned} C_n^* &\leq \sum_{e \in A} a_e |T|^{(k+1)/n} (z_e^*)^{k+1} \\ &\leq |T|^{(k+1)/n} \sum_{e \in A} a_e (z_e^*)^{k+1} \\ &= |T|^{(k+1)/n} C^*. \end{aligned}$$

Thus, to ensure that $|C^* - C_n^*|/C^* \leq \delta$, we can solve for n to arrive at

$$n(\delta) > \frac{(k+1) \log |T|}{\log(1 + \delta)}.$$

We note in passing that this bound is independent of the graph topology. We also refer the reader to Figures 1 and 2 in Section V-E which show that the \mathcal{L}_n -relaxation closely approximates the $\max(\cdot)$ function for even small values of n . ■

D. Multicast over Capacitated Links

We next construct a feasible multicast over a set of capacitated links based on the above analysis. Let k_e be the capacity of edge $e \in A$. Suppose it is further desirable that most links in the network are loaded below $(1 - \delta)$ factor of their capacities. This may be necessary to satisfy quality of service requirements, such as those on the average delay (note that in the presence of bursty traffic, the queueing delay becomes unbounded as the load approaches unity¹).

¹Recall that in the presence of stochastically time-varying flows (for instance, bursty packet rates from applications such as video/multimedia, flow connection initiations and terminations, etc), the average delay of a flow in a queueing system becomes large as the flow size gets close to the link capacity. This can be readily seen in an M/M/1 queue where the average delay increases as $1/\delta$. In general, for a GI/GI/1 queue, with inter-arrival times T of variance σ_T^2 and inter-service times X of variance σ_X^2 , from Kingman’s upper and lower bounds [23],

$$\frac{E[T]\sigma_X^2 - E[X](2 - \rho)}{2(1 - \rho)} \leq E[W] \leq \frac{\sigma_X^2 + \sigma_T^2}{2(1 - \rho)}$$

we know that as the offered load ρ approaches 1 – that is as the mean arrival rate of data on a link is close to the link capacity – the mean waiting time for a packet entering the queue $E[W]$ scales in proportion to $(1 - \rho)^{-1}$. The result can be extended from one queue to a Generalized Jackson Network of queues, where Gamarnik and Zeevi [24] demonstrate that in the heavy traffic limit as $\rho_n = 1 - k/\sqrt{n}$ for large n , the mean delay scales as $O(\sqrt{n})$.

To solve the above constrained multicast problem, we define the cost function of each edge, $e \in A$, as

$$c_e(x) \triangleq \left(\frac{x}{k_e(1-\delta)} \right)^m.$$

Observe that as $m \rightarrow \infty$, edge costs will tend to zero for edges that satisfy the above constraints and become large for edges that do not. Hence, if there is a feasible flow allocation f for (G, c, R) over the constrained links, then the cost $C(f)$ at Nash equilibrium will be small, tending to zero as m becomes large. In fact, there exists m_0 such that for all $m > m_0$ the flow at Nash equilibrium satisfy the capacity constraints of all edges. More specifically, let $|A|$ denote the size of A (i.e., it is the total number of edges in the network), then an upper bound on m_0 is

$$m_0 = \frac{\log |A|}{-\log(1-\delta)}. \quad (9)$$

The reason for this is as follows. Since there exists a feasible flow f satisfying the above constraints, the global min-cost with the edge cost functions as above is at most $|A|$. Now, from Theorem 3.1, the cost of flow \hat{f} at Nash equilibrium is the same as the global min-cost. Hence, it is too at most $|A|$. However, if \hat{f} were to assign flow to an edge greater than its capacity, the cost on that edge alone will be at least $1/(1-\delta)^m$, which would be greater than $|A|$ for any $m > m_0$, where m_0 is as in (9).

IV. MULTIPLE MULTICASTS

In this section, we generalize the single-source multicast problem to the multiple-multicast sessions problem where each session corresponds to a source node taken from the set $S \in N$. Within each multicast session $s \in S$, network coding is performed across packets destined for sinks in set $T_s \subseteq N$. However, packets are not encoded across sessions to ensure computationally tractable decoding at each sink. So, each sink $t \in T_s$, can steer its flows across the set of paths \mathcal{P}_t^s from source s to sink t so as to deliver a total rate of R_t^s . As before, the total set of paths $\mathcal{P} \triangleq \bigcup_{s \in S} \bigcup_{t \in T_s} \mathcal{P}_t^s$.

We can then formulate the min-cost problem M-GLOBAL(G, c, R) for multiple multicasts in the same way as [12]:

$$\begin{aligned} \text{M-GLOBAL}(G, c, R) \\ \text{minimize } C(f) &= \sum_{e \in A} c_e(z_e) \\ \text{subject to } z_e &= \sum_{s \in S} z_{e,s} \quad \forall e \in A \\ z_{e,s} &\triangleq \max_{t \in T_s} \left\{ \sum_{P \in \mathcal{P}_t^s: e \in P} f_P \right\}, \quad f_P \geq 0 \quad \forall P \in \mathcal{P} \\ \sum_{P \in \mathcal{P}_t^s} f_P &= R_t^s \quad \forall s \in S, \forall t \in T_s. \end{aligned}$$

The corresponding \mathcal{L}_n -relaxed cost function for M-GLOBAL(G, c, R) is

$$C_n(f) = \sum_{e \in A} c_e \left(\sum_{s \in S} \left[\sum_{t \in T_s} \left(\sum_{P \in \mathcal{P}_t^s: e \in P} f_P \right)^n \right]^{1/n} \right).$$

Differentiating the equivalent Lagrangian

$$\mathcal{L}(f, \lambda, \mu) = C_n(f) + \sum_{s \in S} \sum_{t \in T_s} \lambda_t^{(s)} \left(\sum_{P \in \mathcal{P}_t^s} f_P - R_t^s \right) - \sum_{P \in \mathcal{P}} \mu_P f_P$$

with respect to a particular flow f_{P_1} , $P_1 \in \mathcal{P}_j^\sigma$ and applying the limit $n \rightarrow \infty$, we observe that at the minimizing flow f , for all $P_1, P_2 \in \mathcal{P}_j^\sigma$,

$$\begin{aligned} &\sum_{e \in P_1} c'_e(z_e^{(n)}) \left(\frac{\sum_{P \in \mathcal{P}_j^\sigma: e \in P} f_P}{z_{e,\sigma}^{(n)}} \right)^{n-1} \\ &\leq \sum_{e \in P_2} c'_e(z_e^{(n)}) \left(\frac{\sum_{P \in \mathcal{P}_j^\sigma: e \in P} f_P}{z_{e,\sigma}^{(n)}} \right)^{n-1}. \end{aligned} \quad (10)$$

where

$$z_{e,s}^{(n)} = \left(\sum_{t \in T} \left(\sum_{P \in \mathcal{P}_t^s: e \in P} f_P \right)^n \right)^{1/n} \quad (11)$$

and $z_e^{(n)} = \sum_{s \in S} z_{e,s}^{(n)}$.

Analogously, for each $s \in S$, we define

$$x_{e,j}^{(s)} \triangleq \sum_{e \in P: P \in \mathcal{P}_j^s} f_P.$$

The edge cost at each edge $c_e(z_e^{(n)})$ is divided among each $s \in S$ in proportion to the $z_{e,s}^{(n)}$ max flow from each s through e , i.e. $c_e(z_e) \frac{z_{e,s}^{(n)}}{z_e^{(n)}}$ is the fraction of the cost picked up by the set of flows towards sinks $t \in T_s$. Further, each $t \in T_s$ picks up $\frac{(x_{e,j}^{(s)})^n}{\sum_{t \in T} (x_{e,t}^{(s)})^n}$ fraction of $z_{e,s}^{(n)}$, which in turn is divided among all flows on paths $P \in \mathcal{P}_t^s$ in the ratio $f_P/x_{e,j}^{(s)}$.

Each sink $t \in T$ attempts to selfishly minimize the cost of transmission to itself. Since the costs across different sessions $\{s : t \in T_s\}$ are additive at each edge and cost functions at each edge are convex, minimizing total cost at sink t is the same as minimizing the cost for each session individually. Hence the criterion for local Nash equilibrium for multiple-session multicasts can be summed up in the following definition.

Definition 4.1: A flow f , feasible for the instance (G, c, R) with multiple-multicast sessions S , is in local Nash equilibrium if for all $\sigma \in S$ and $j \in T_\sigma$, for any $P_1, P_2 \in \mathcal{P}_j^\sigma$,

$$\begin{aligned} &\sum_{e \in P_1} \frac{c_e(z_e^{(n)})}{z_e^{(n)}} \left(\frac{\sum_{P \in \mathcal{P}_j^\sigma: e \in P} f_P}{z_{e,\sigma}^{(n)}} \right)^{n-1} \\ &\leq \sum_{e \in P_2} \frac{c_e(z_e^{(n)})}{z_e^{(n)}} \left(\frac{\sum_{P \in \mathcal{P}_j^\sigma: e \in P} f_P}{z_{e,\sigma}^{(n)}} \right)^{n-1}. \end{aligned} \quad (12)$$

We observe the similarity between (10) and (12) analogous to that between (1) and (8). The following corollary follows analogously from the reasoning in Section III:

Corollary 4.1: If for an instance $\mathcal{L}_n = (G, c, R)$, the cost function at each edge e is of the power law form $c_e(z_e^{(n)}) = a_e(z_e^{(n)})^{k+1}$ for any fixed $k \in \mathbb{R}$, $k > 0$, then the cost of flow f at local Nash equilibrium $C_n(f)$ equals the cost $C_n(f^*)$ of the global min-cost flow f^* .

V. DISTRIBUTED ALGORITHMS FOR MIN-COST FLOW

Section III demonstrates that the sum-cost of the edges with any uniform power-law edge cost function under user equilibrium is the same as the min-cost. This result lends itself readily to the construction of a simple non-cooperative optimal min-cost flow routing algorithm for a single-source multicast with network coding. The following section deals with the single-source multicast for sake of simplicity. It is easy to show that due to the separable and additive nature of the costs for the multiple-source multicast, we can run the same algorithm independently over each session to reach the user equilibrium in this case too.

In this section, we develop two algorithms: User Equilibrium with Single Source Multicast (UESSM) and Local Distributed Selfish Routing Algorithm (LDSRA).

UESSM is a non-cooperative decentralized flow-steering algorithm that provably converges to the min-cost flow allocation for the class of convex, monomial edge cost functions defined in Section III. At each receiver, UESSM “flow-steers” among the paths leading to it in order to greedily minimize its cost. This allows us to achieve the min-cost flow with network coding, without having to perform per-flow primal-dual type calculations at every intermediate router.

The Local Distributed Selfish Routing Algorithm (LDSRA) for min-cost routing is a local distributed algorithm where nodes in the network adjust flow fractions based on the local flow and cost information at each node. This is an analog of the Bellman-Ford algorithm, however, in the context of network coding. By using the end-to-end delay experienced by a probe packet as the marginal cost, LDSRA minimizes the total network latency (sum cost) by reallocating flows from the more expensive (greater delay) neighbor toward a cheaper (lower delay) one.

A. UESSM: User Equilibrium with Single Source Multicast

The implementation of this algorithm, UESSM, assumes *flow routing* between the source and destination, where the source router encodes downstream hop-by-hop routing information into the IP-header, as can be implemented in IPv6. The intermediate routers in the network between the source and sink do not need to maintain state-information locally. All that the intermediate routers need to do is route packets along the outgoing edges corresponding to the hop-by-hop information embedded in each packet and network code across packets of the same type at each instant of time using a random linear code.

Also, each downstream packet aggregates the cost that it has paid along each edge on a particular flow path. For efficiency,

this information need not be carried by every downstream packet, but only by representative packets at each iteration of the algorithm.

Algorithm: UESSM

Initialization: In our implementation, we will choose a $\epsilon > 0$ small enough [cf. Section V-C] such that R/ϵ is a positive integer, and require that all flow rates be at-least ϵ (a “keep-alive” rate). Also, the flow allocations in our implementation are elements from a lattice $\mathcal{L} = \{0, \Delta, 2\Delta, \dots, R\}$, for some fixed $\Delta > 0$ such that R/Δ and ϵ/Δ are positive integers. Thus, we can initialize at any arbitrary point on this lattice. For instance, for each sink $t \in T$ we can initialize at $f_{P'_t} = R - (Q_t - 1)\epsilon$ for some $P'_t \in \mathcal{P}_t$ and $f_P = \epsilon$ for all $P \in \mathcal{P} \setminus \{P'_t\}$.

Step: Now, one of the sinks $t \in T$ is chosen at random. Let us label the paths $P_1, P_2, \dots, P_{Q_1}, P_{Q_1+1}, \dots, P_{Q_1+Q_2}, \dots, P_{Q_1+\dots+Q_{T-1}+1}, \dots, P_{Q_1+\dots+Q_{T-1}+Q_T}$ where Q_i is the number of paths from the source to destination t . For some fixed $\xi > 0$, a receiver t picks a pair of paths $P_{\sum_{i=1}^{t-1} Q_i+l}$ and $P_{\sum_{i=1}^{t-1} Q_i+m}$, for any $l, m = 1, 2, \dots, Q_t$ with $l \neq m$. Denoting

$$P_{t,l} = P_{\sum_{i=1}^{t-1} Q_i+l} \quad (13)$$

$$P_{t,m} = P_{\sum_{i=1}^{t-1} Q_i+m} \quad (14)$$

if

- (a) $d_{P_{t,l}}(f) > d_{P_{t,m}}(f) + \xi$
- (b) $f_{P_{t,l}} \geq \epsilon + \Delta$
- (c) $f_{P_{t,m}} \leq R - \Delta$

then, $f_{P_{t,l}} \leftarrow f_{P_{t,l}} - \Delta$, $f_{P_{t,m}} \leftarrow f_{P_{t,m}} + \Delta$. Conversely, if (a), (b) and (c) hold with $P_{t,l}$ and $P_{t,m}$ interchanged, then $f_{P_{t,m}} \leftarrow f_{P_{t,m}} - \Delta$, $f_{P_{t,l}} \leftarrow f_{P_{t,l}} + \Delta$.

Termination: No user can make any flow switch if and only if $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) \geq -\xi$, $\forall t, l, m$ which are feasible (i.e., (b) and (c) above are satisfied). In other words, at termination, for any receiver t and any pair of flows l, m $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) < -\xi$ if and only if $f_{P_{t,m}} = \epsilon$.

Note that we would like to distinguish between the terms ‘step’ and ‘iteration’ as follows. By a ‘step’, we will mean the sequence of operations defined in the algorithm above. On the other hand by an ‘iteration’ we mean a ‘step’ that results in a flow reallocation. This distinction is made because due to random selection, at a step no flow reallocation may occur. In the rest of the paper, we will only count ‘iterations’.

B. Asynchronous implementation

The implementation of the algorithm above does *not* require synchronous timing between the clocks at the various sink nodes but only requires that the clocks have the same cycle frequency. We assume that the path-delay timescale along the network (for the update of the path costs etc.) is negligible compared to the time-steps in which the algorithm proceeds. Each sink $j \in T$ picks a random delay that is exponentially

distributed before adjusting its flows in the manner outlined in Subsection V-A. Since the exponential distribution is a continuous time-distribution, the collision probability is small. Further, since all flow steering is implemented at the source, the source can be designed to sequentially adjust flows of each sink. This ensures that only one sink adjusts flows at a time in the asynchronous algorithm, thereby retaining the same features as the synchronous implementation. Henceforth, we will denote each source adjustment (reallocation) as an iteration of the algorithm.

C. Convergence of UESSM to the min-cost flow

In this section, we restrict ourselves to edge cost functions of the form $c_e(x) = a_e x^{k+1}$, $k > 0$, as discussed in Section III. From Theorem 3.1, it follows that a global optimum is the same as the cost at a Nash equilibrium.

Recall that in UESSM we are restricting each flow to have a rate of at least ϵ . Now, we choose ϵ as follows: Given any $\alpha > 0$, we will choose ϵ such that

- (i) $\epsilon > 0$
- (ii) $\epsilon < R/Q_t \quad \forall t = 1, 2, \dots, |T|$
- (iii) $R/\epsilon \in \mathbb{N}$
- (iv) $|C_n^* - C_n^*(\epsilon)| < \alpha$,

where C_n^* is the optimal cost to \mathcal{L}_n -GLOBAL(G, c, R), and $C_n^*(\epsilon)$ is the optimal cost to \mathcal{L}_n -GLOBAL(G, c, R) under the additional constraint that $f_P \geq \epsilon$, $\forall P \in \mathcal{P}$. Observe that under this restricted simplex, the GLOBAL problem is still convex and differentiable. Further, since the cost functions are differentiable and finite, and the constraint sets are convex, given any $\alpha > 0$, there exists an ϵ such that the above conditions hold. Also, let $f^*(\epsilon)$ be an optimal solution to \mathcal{L}_n -GLOBAL(G, c, R) with the ϵ -restricted convex constraint set.

Next, for any destination $t \in T$ we will formalize the notion of an infinitesimal reallocation of flows from path $P_{t,l}$ to path $P_{t,m}$ as defined in Equations (13),(14). Recall that due to monomial cost edge function and the \mathcal{L}_n -approximation, the global cost function $C_n()$ is differentiable at all points. Accordingly, we can define $\nabla C_n(f) = (\frac{\partial C_n(f)}{\partial f_P})_{P \in \mathcal{P}}$ to be the $|\mathcal{P}|$ -sized vector whose elements are $\frac{\partial C_n(f)}{\partial f_P}$.

Further, we define *direction vectors* \mathcal{E} to be the collection of all vectors of the form $e_{t,l,m} = [0, \dots, 0, -1, 0, \dots, 0, 1, 0, \dots, 0] \in \{-1, 0, 1\}^{|\mathcal{P}|}$ where the $P_{t,l}$ -th element is -1 and the corresponding $P_{t,m}$ -th element is 1 . (Note that the vectors are not necessarily linearly independent – for example, $e_{t,l,m} = -e_{t,m,l}$). Accordingly, an infinitesimal shift of flow from $P_{t,l}$ to $P_{t,m}$ is given by the inner product $\nabla C_n(f)^T \cdot e_{t,l,m}$.

In the following, we will utilize the property that the gradient function $\nabla C_n(f)$ is Lipschitz over the the space of feasible flow vectors f .

Lemma 5.1: $\nabla C_n(f)$ is Lipschitz in the space of feasible flow vectors f with Lipschitz constant $L(\epsilon)$.

Proof: We refer the reader to [28]. ■

Recall that the constraint set (set of feasible flow rates) is described by a convex set where the flows corresponding to each receiver t is constrained to lie on a $|Q_t|$ -dimensional

(scaled) simplex (i.e., for each $t \in T$, $\sum_{l \in Q_t} f_{t,l} = R$, $f_{t,l} \geq 0$). For each f in the constraint set, we denote $\mathcal{E}(f) \subseteq \mathcal{E}$ to be the set of *feasible direction vectors*, where any $e_{t,l} \in \mathcal{E}(f)$ satisfies the following: a Δ shift of flow in the direction $e_{t,l}$ from f leads to f' which is a feasible flow vector.

Lemma 5.2: Fix any $\alpha > 0$. Then, choose the following parameters for the UESSM algorithm:

- (i) Let $\xi = \frac{\alpha}{2^{(k+1)R/|\mathcal{P}|^2}}$, where k is the exponent in the edge cost function ($c_e(z) = a_e(z^{(n)})^{k+1}$),
- (ii) Choose any $\Delta \leq \min\{\epsilon/10, \frac{\xi(k+1)}{2L(\epsilon)}\}$ such that ϵ/Δ is a positive integer, and $L(\epsilon)$ is given in Lemma 5.1.

With the above conditions, the following holds: Suppose that the algorithm UESSM terminates at iteration M . Then, $|C_n(f^{(M)}) - C_n^*(\epsilon)| < \alpha$, where $C_n(f^{(M)})$ is the cost with flow allocation $f^{(M)}$, and $C_n^*(\epsilon)$ is the optimal cost of the convex problem \mathcal{L}_n -GLOBAL(G, c, R) under the constraint that for all $P \in \mathcal{P}$, $f_P \geq \epsilon$.

Proof: Let $f^*(\epsilon)$ be an optimal flow corresponding to solution $C_n^*(\epsilon)$ and $f^{(M)}$ denote the flow in the M -th iteration (termination) of the algorithm.

Since $C_n()$ is convex, it follows from the gradient formula that

$$\begin{aligned} C_n(f^{(M)}) &\geq C_n(f^*(\epsilon)) \\ &\geq C_n(f^{(M)}) + \nabla C_n(f^{(M)})^T \cdot (f^*(\epsilon) - f^{(M)}). \end{aligned} \quad (15)$$

Now, recall $\mathcal{E}(f^{(M)})$ is the set of feasible direction vectors corresponding to flow $f^{(M)}$.

Claim: There exists non-negative $\pi_{t,l,m}$ such that

$$f^*(\epsilon) = f^{(M)} + \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} e_{t,l,m} \quad (16)$$

Proof: For each $t \in T$, we define the vector γ_t to be a $\{0, 1\}$ vector of dimension \mathcal{P} where $\gamma_{t,l} = 1$ for all $l = Q_{t-1} + 1, \dots, Q_t$ and 0 otherwise (i.e., γ_t corresponds to the flows destined for receiver t).

We now decompose $f^*(\epsilon) - f^{(M)} = \sum_{t \in T} (f_t^*(\epsilon) - f_t^{(M)})$ where $f_t^* = f^*(\epsilon) * \gamma_t$, (and similarly for $f_t^{(M)}$) where the $*$ operation corresponds to term-by-term multiplication (thus, $f_t^*(\epsilon)$ corresponds to the flows for receiver t).

Now, consider the (scaled and shifted) simplex of feasible flows to receiver 1, i.e., $\mathcal{A}_1 = \{\sum_{l=1}^{Q_1} f_{1,l} = R \text{ and } f_{1,l} \geq \epsilon\}$. Let $v_i, i = 0, 1, 2, \dots, Q_1$ be the vertices of the (scaled and shifted) simplex \mathcal{A}_1 (the n dimensional scaled and shifted simplex has Q_1 vertices, with each vertex having one component equal $R - (Q_1 - 1)\epsilon$ and all other components being ϵ).

As the set is a convex simplex, we have for some $a_i \geq 0$, $\sum_i a_i = 1$,

$$f_1^*(\epsilon) = \sum_{i=0}^{Q_1} a_i v_i$$

Now, we consider two cases:

Case (i): $f_1^{(M)}$ is an interior point of \mathcal{A}_1 .

In this case, all directions vectors $e_{1,l,m}$ at $f^{(M)}$ are feasible, and the existence of non-negative of $\pi_{1,l}$ is immediate. All direction vectors are feasible for the following reason: We have

by the algorithm description (and the explicit construction of the values given in the Lemma statement) that $f^{(M)}$ lies on the Δ lattice and ϵ/Δ and R/Δ are positive integers. Thus, $f^{(M)}$ lies in the interior of \mathcal{A}_1 implies that each component of $f_1^{(M)}$ has flow rate value of at-least $\epsilon + \Delta$, in which case all direction vectors are feasible.

Case (ii): $f_1^{(M)}$ is a boundary point of \mathcal{A}_1 .

Now, note that for a simplex of dimension $Q_1 - 1$, all boundary points can be described by (a) points that lie strictly within the interior of a simplex of dimension k for some $k = 1, \dots, Q_1 - 2$ or (b) the boundary point lies on a vertex of the simplex (i.e, $k = 0$).

For case (a), without loss of generality, let the boundary point be in the interior of a simplex of dimension k with vertices v_0, v_1, \dots, v_k . Then,

$$f_1^{(M)} = \sum_{i=0}^k b_i v_i$$

for non-negative b_i such that $\sum_i b_i = 1$. Thus, we have

$$\begin{aligned} f_1^*(\epsilon) - f_1^{(M)} &= \sum_{i=0}^{Q_1} a_i v_i - \sum_{i=0}^k b_i v_i \\ &= \sum_{i=0}^k a_i v_i + \sum_{i=k+1}^{Q_1} a_i (v_i - v_k) + v_k \left(\sum_{i=k+1}^{Q_1} a_i \right) - \sum_{i=0}^k b_i v_i \end{aligned}$$

Now, let

$$\tilde{f}_1 = \sum_{i=0}^{k-1} a_i v_i + \left(\sum_{i=k}^{Q_1} a_i \right) v_k$$

Then, note that since $\sum_i a_i = 1$, \tilde{f}_1 lies in the k -dimensional simplex with vertices $\{v_i, i = 0, \dots, k\}$. We now have

$$f_1^*(\epsilon) - f_1^{(M)} = (\tilde{f}_1 - f_1^{(M)}) + \sum_{i=k+1}^{Q_1} a_i (v_i - v_k)$$

where, by construction, $a_i \geq 0$ and \tilde{f}_1 and $f_1^{(M)}$ lie on the k -dimension simplex, with $f_1^{(M)}$ in the strict interior of this simplex. Thus, all vectors within the simplex are feasible (i.e., the direction vectors corresponding to both $(v_i - v_k)$ and $(v_k - v_i)$ for $i = 0, 1, \dots, k$ are feasible as $f_1^{(M)}$ in the strict interior), we can choose feasible directions with non-negative weights to move from $f_1^{(M)}$ to \tilde{f}_1 . In other-words, $(\tilde{f}_1 - f_1^{(M)})$ can be expressed as a non-negative weighted sum of feasible direction vectors.

For case (b) where we are terminating at the vertex (say v_0) of the $Q_1 - 1$ dimension simplex, the existence of non-negative π_{t_i} follows because $f_1^*(\epsilon)$ is in the (ϵ) -constrained set, and the feasible directions include all directions of the form $v_i - v_0, i = 1, 2, \dots, Q_1$ which span the simplex set.

The proof is analogous for all other receivers. ■

Thus, from (15) and (16), we have

$$\begin{aligned} C_n(f^{(M)}) &\geq C_n(f^*(\epsilon)) \\ &\geq C_n(f^{(M)}) + \nabla C_n(f^{(M)})^T \cdot \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} e_{t,l,m} \end{aligned}$$

where $\pi_{t,l,m}$ are non-negative and $e_{t,l,m}$ are feasible. From the termination condition of UESSM, we now have that along all feasible directions, $d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \geq -\xi$. This is due to the following reason: Suppose $e_{t,l,m}$ is a feasible direction. This implies a Δ flow reallocation is allowed from flow $f_{P_{t,l}}$ to flow $f_{P_{t,m}}$ at iteration M . However, by the statement of the Lemma, iteration M is the termination step. Thus, UESSM decides not to re-allocate from flow $f_{P_{t,l}}$ to flow $f_{P_{t,m}}$. This can happen due to one of two possibilities: (A) $d_{P_{t,l}}(f^{(M)}) \leq d_{P_{t,m}}(f^{(M)})$ (i.e., $P_{t,l}$ is already a ‘‘cheaper’’ path than $P_{t,m}$, so UESSM does not further decrease the rate along flow $f_{P_{t,l}}$), in which case we have $d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \geq 0 > -\xi$. The other possibility is (B) where $d_{P_{t,m}}(f^{(M)}) \leq d_{P_{t,l}}(f^{(M)}) < d_{P_{t,m}}(f^{(M)}) + \xi$ (i.e., the cost along path $P_{t,l}$ is only ‘‘slightly’’ more expensive than path $P_{t,m}$, and thus, UESSM decides not to switch). In case (B), we have $0 \geq d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \geq -\xi$.

Now, because the edge cost function is of the form $a_e(z^{(n)})^{k+1}, k > 0$, we have

$$\nabla C_n(f^{(M)})^T \cdot e_{t,l,m} = (k+1) (d_{P_{t,m}}(f) - d_{P_{t,l}}(f))$$

Thus, from the termination condition, we have

$$\nabla C_n(f^{(M)})^T \cdot \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} e_{t,l,m} \geq -(k+1) \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} \xi$$

(In other words, at termination, along all feasible directions, the negative gradient is small).

Now, due to the fact that $\pi_{t,l,m}$ are bounded by R (as the space is bounded) and the fact that $|\mathcal{E}(f^{(M)})|$ is finite (because the number of paths are finite, an upper bound is $2|\mathcal{P}|^2$), we can choose ξ as in the Lemma statement to ensure that the difference in cost is no more than α . ■

Lemma 5.3: For a given $\alpha > 0$, let us fix $\xi = \frac{\alpha}{2(k+1)R|\mathcal{P}|^2}$. Choose any (strictly) positive $\Delta \leq \frac{\xi(k+1)}{2L}$ such that ϵ/Δ is a positive integer, and $L = L(\epsilon)$ is given in Lemma 5.1. Further, choose $\beta = \Delta((k+1)\xi - \Delta L)$.

Suppose that at iteration M , there exists a user $t \in T$, $P_{t,l} = P_{\sum_{i=1}^{t-1} Q_i + l}$ for some $l = 1, 2, \dots, Q_t$ and $P_{t,m} = P_{\sum_{i=1}^t Q_i + m}$, for some $m = 1, 2, \dots, Q_t$ (i.e. $P_{t,l}, P_{t,m} \in \mathcal{P}_t$) such that $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) < -\xi$ and $f_{P_{t,m}} \geq \epsilon + \Delta$ (i.e., a Δ flow switch is feasible).

Then, we have that a flow switch of Δ from flows $P_{t,m}$ to $P_{t,l}$ ensures that $C_n^{(M+1)} - C_n^{(M)} < -\beta < 0$.

Proof: To prove that a flow readjustment will cause a reduction in the overall cost function, we will borrow some results from the proof of convergence of constant step-size descent algorithms. Specifically, we will use the techniques in [22, Props 1.2.3, A.24] to demonstrate that if the gradient of the cost function is Lipschitz over the state space of flows, then if the difference of marginal costs between the paths are outside a ball of size ξ , the net cost reduction following a $\Delta \leq \frac{\xi(k+1)}{2L}$ readjustment of flows will be at least by $\beta > 0$, where $\beta = \Delta((k+1)\xi - \Delta L)$.

Note that by considering a flow reallocation from $P_{t,m}$ to $P_{t,l}$, the direction of descent $e_{t,m,l} = [0, 0, \dots, 0, -1, 0 \dots 0, 1, 0 \dots 0]$, where elements -1 and 1

correspond to paths $P_{t,m}$ and $P_{t,l}$ in the $|\mathcal{P}|$ length vector $e_{t,m,l}$.

Now,

$$\begin{aligned} C_n(f) &= \sum_{e \in A} a_e(z_e)^{k+1} \\ &= \sum_{e \in P} a_e(z_e)^{k+1} + \sum_{e \in A \setminus P} a_e(z_e)^{k+1}. \end{aligned}$$

Thus, differentiating with respect to flow f_P where $P \in \mathcal{P}_t$

$$\begin{aligned} \frac{\partial C_n(f)}{\partial f_P} &= \frac{\partial}{\partial f_P} \sum_{e \in P} a_e(z_e)^{k+1} + 0 \\ &= \sum_{e \in P} a_e(k+1)(z_e)^k \left(\frac{x_{e,t}}{z_e} \right)^{n-1}. \quad (17) \end{aligned}$$

Also, observe that from (7) and (17), for the considered class of cost functions $c_e(z_e) = a_e z_e^{k+1}$,

$$\begin{aligned} \nabla C_n(f)^T \cdot e_{t,m,l} &= \frac{\partial C_n}{\partial f_{P_{t,l}}} - \frac{\partial C_n}{\partial f_{P_{t,m}}} \\ &= (k+1)(d_{P_{t,l}}(f) - d_{P_{t,m}}(f)) \quad (18) \\ &< -(k+1)\xi \quad (19) \end{aligned}$$

where the last step follows from the lemma flow condition.

From the descent lemma [22, Prop A.24], we have that if L is the Lipschitz constant for $\nabla C_n(f)$ over the space of f , then

$$\begin{aligned} C_n(f + \Delta e_{t,m,l}) - C_n(f) &\leq \Delta \nabla C_n(f)^T \cdot e_{t,m,l} + \frac{1}{2} \Delta^2 L \|e_{t,m,l}\|^2 \\ &< \Delta \left(-(k+1)\xi + \frac{1}{2} \Delta L \|e_{t,m,l}\|^2 \right) \\ &\leq \Delta (-(k+1)\xi + \Delta L) \end{aligned}$$

Choosing $\Delta \leq \frac{\xi(k+1)}{2L}$, (such that ϵ/Δ is a positive integer), we have the desired result. \blacksquare

We are now ready to state the main result of this section.

Theorem 5.1: Choose the parameters α , ξ and Δ given in Lemma 5.2. Then, UESSM converges in a finite number of iterations (at iteration M), with the termination condition satisfying $|C_n(f^{(M)}) - C_n^*| < 2\alpha$, where $C_n(f^{(M)})$ is the cost with flow allocation $f^{(M)}$, and C_n^* is the optimal cost of the convex problem $\mathcal{L}_n\text{-GLOBAL}(G, c, R)$.

Proof: For each flow allocation f that is not at the terminal condition, by the description of UESSM, there exists at least one user $t \in T$ and some pair of paths $P_{t,l}, P_{t,m} \in \mathcal{P}_t$ such that $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) > -\xi$ for which a Δ flow reallocation is feasible.

Then, we have from Lemma 5.3 any feasible flow reallocation will reduce the sum cost by at least $\beta > 0$, i.e. $C_n(f^{(s+1)}) - C_n(f^{(s)}) \leq -\beta < 0$ at iteration s . Hence, at each iteration until the termination condition is reached, the cost function decreases by at least β . The initial cost $C_n(f^{(0)})$ of the iterations is positive bounded and the cost is non-negative. This implies that the algorithm will terminate in a finite number of iterations. Finally, from Lemma 5.2 the termination condition satisfies $|C_n(f^{(M)}) - C_n^*(\epsilon)| < \alpha$.

Further, note that we have chosen ϵ such that $|C_n^* - C_n^*(\epsilon)| < \alpha$ holds. Hence, by the triangle inequality, $|C_n(f^{(M)}) - C_n^*| < 2\alpha$. \blacksquare

Although, UESSM converges and has provably good convergence properties, UESSM requires the source to maintain path information for all paths from the source to the destinations. This motivates the design of a local distributed algorithm where nodes adjust flow fractions based on the local flow and cost information at each node. We present such an algorithm in the following subsection.

D. Local Distributed Selfish Routing Algorithm (LDSRA) for Min-Cost Routing

Local routing algorithms for cost minimization have been studied in the past in the context of ad hoc network routing protocols, such as STARA [25], [26]. Such an algorithm can be implemented with an exponential-forgetting estimation as in STARA to estimate marginal costs from the source to each downstream node in the network and adjust fractional allocation of flows at each node so as to minimize the local marginal cost.

The algorithm proceeds in two phases. In the first phase, each node s identifies a set of neighbors N_s^k to reach destination k . Also, each node intermittently transmits probe packets along N_s^k , which accumulate marginal costs along the paths, and the feedback from these are used to estimate the marginal cost $D_{s,n}^k(t)$ from s to k along each particular neighbor $n \in N_s^k$ at time t . The second phase is the flow reallocation phase. Each node compares the estimated marginal costs of flows to a particular destination and then shifts flow allocation by a fraction Δ from the neighbor with higher marginal cost to one with lower marginal cost.

It can be shown that under steady state a flow allocation is at a user equilibrium (cf. Definition 3.1) if and only if all utilized paths from each node to each sink have equal marginal costs (see for instance Lemma 6.1 [26]). Since the above flow reallocation phase achieves the latter objective, under steady state it will also be at a user equilibrium. Further, if all edge costs are monomials, from Theorem 3.1 it follows that the above flow reallocation will have globally minimum cost in steady state.

In our simulations discussed next, we assume that the rate at which nodes reallocate flows is much slower than the rate at which probe packets are generated and cost estimates are gathered by each node. This allows us to assume, for purposes of simulation, that the estimates $D_{s,n}^k(t)$ at each node s are ideal. Future work will focus on designing stochastic-approximation based algorithms for joint estimation and rate allocation in this distributed framework, as well as analyzing its convergence and the rate of convergence under these conditions similar to the analysis in [27].

E. Simulation results

We simulate UESSM over the classic 7-node butterfly network in [1], [2] with the edge costs shown in Figure 1 for a rate 1 multicast session from source S_1 to destinations D_1 and

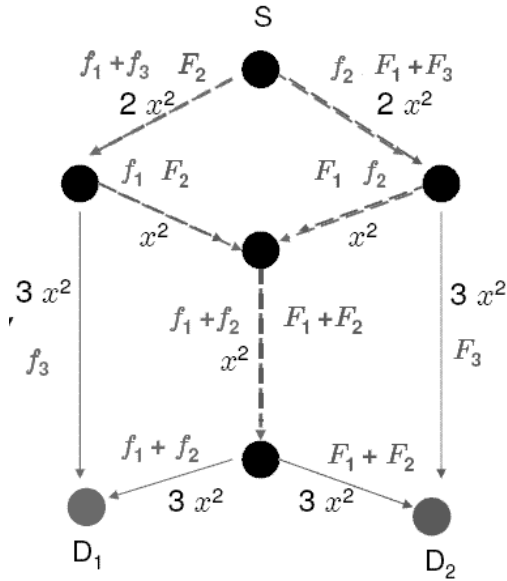


Fig. 1. 7-node Butterfly network.

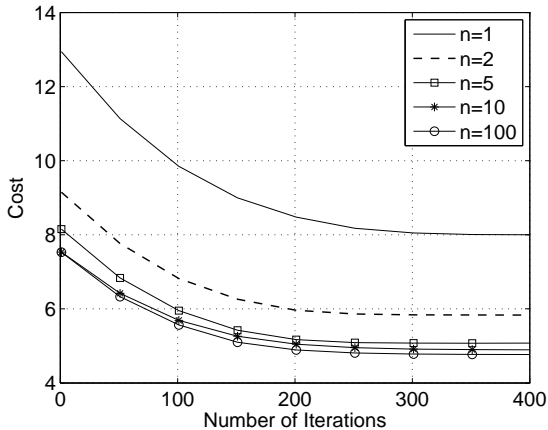


Fig. 2. UESSM Algorithm: Comparing \mathcal{L}_n approximation for $n = 1, 2, 5, 10, 100$.

D_2 . The links are marked with the edge cost functions $c_e(x)$. In this example, $\mathcal{P}_1 = \{f_1, f_2, f_3\}$ and $\mathcal{P}_2 = \{F_1, F_2, F_3\}$.

We first study how $C_n(f)$ changes with increasing values of n in the \mathcal{L}_n -approximation to the max function. The trajectories for 100 representative UESSM runs with $\Delta = 0.01$ with varying values of n are plotted in Figure 2. The $n = 1$ case corresponds to multicast without network coding and has a much higher sum-cost than that achieved by the \mathcal{L}_{100} -approximation, which is very close to the cost with using the non-differentiable max function in $\text{GLOBAL}(G, c, R)$. However, we note that there is not much gain in going from $n = 10$ to $n = 100$. This suggests that the \mathcal{L}_n -approximation works well for even small values of n . Recall that we have bounded the minimum value of $n(\delta)$ given an approximation error target $\delta > 0$ in Remark 3.1.

We have also shown error bars corresponding to one standard deviation about the mean, with random initial condi-

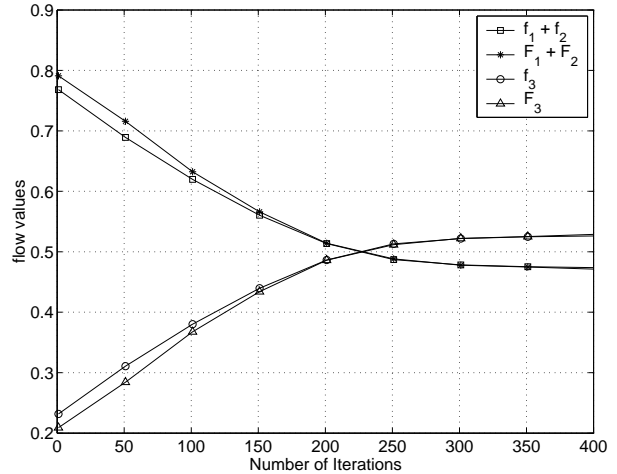
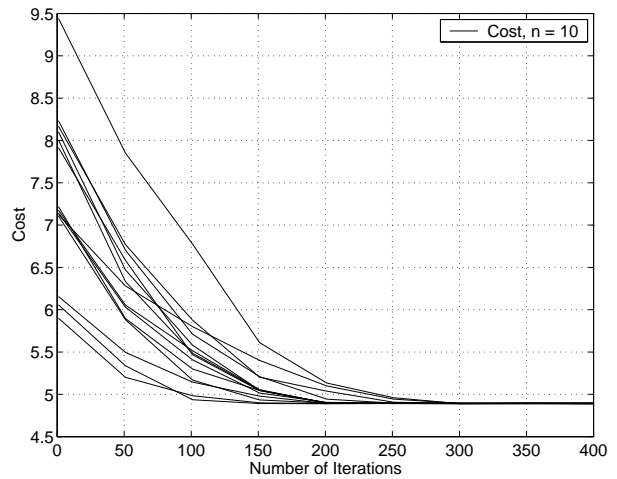


Fig. 3. UESSM Algorithm trajectories: Sum costs and flows for the Butterfly network, \mathcal{L}_{10} -GLOBAL(G, c, R), $\Delta = 0.01$.

tions. We observe that, irrespective of initial conditions, the simulation sum-cost trajectories converges to the mean with progressively small variance. Typical trajectories of flow rates through various paths for the Butterfly network are presented in Figure 3 with a step-size of $\Delta = 0.01$.

We next provide simulation results with the LDSRA algorithm for the same Butterfly network. The costs under two \mathcal{L}_n -approximations ($n = 1, 10$) are plotted in Figure 4. We note that as expected, \mathcal{L}_n cost decreases as n increases. Further, a comparison of Figures 2 and 4 verifies that both algorithms converge to the same sum-cost. Also, we compare the flows through the central edge in Figure 5 and observe that the equilibrium state corresponds to the symmetric min-sum cost flow allocation.

VI. CONCLUSION

In this work, we have presented a cost splitting rule at each link for the min-cost problem using network coding and demonstrated that under this rule, the sum-cost across the network at user equilibrium is the same as the min-cost subject to the condition that all edges satisfy a uniform monomial cost function. Further, based on this result, we present two selfish min-cost routing algorithms - UESSM and LDSRA - which

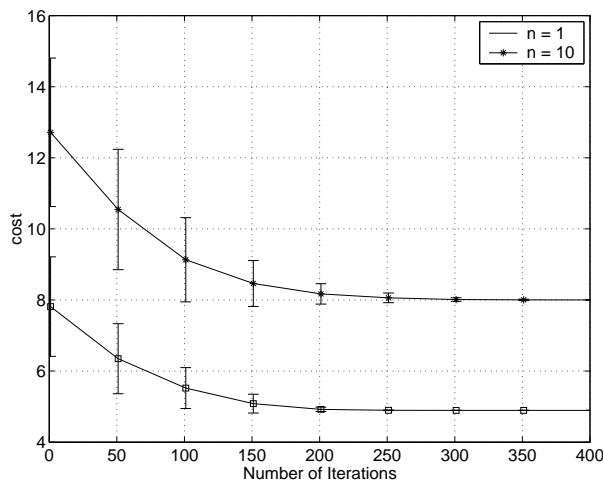


Fig. 4. Butterfly Network with the LDSRA Algorithm: \mathcal{L}_n approximation for $n = 1, 10$.

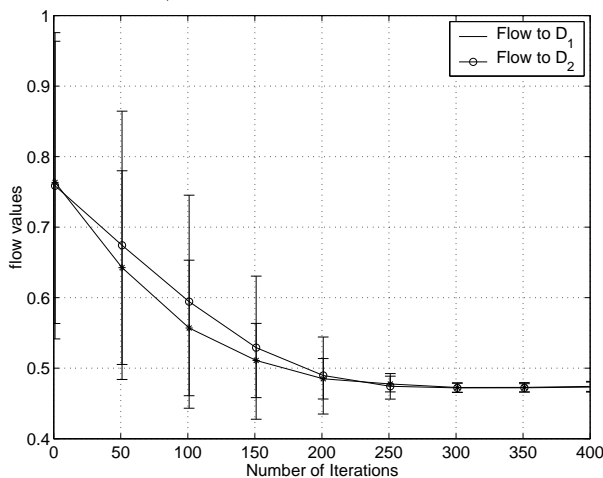


Fig. 5. Butterfly Network with the LDSRA Algorithm: Flow allocation to central edge.

have desired performance in simulations. Additionally, we prove that UESSM converges to the min-cost flow allocation for any network topology.

Observe that in our discussion of multicast with many sources, we restricted the mixing of data to only between flows from a particular sink. However, note that mixing between flows from different sources would involve designing a network code for the many-sources many-sinks problem. It is known that optimal code-design for such a case is NP-Hard [20]. Thus, the design and analysis of approximation algorithms for network coding with multiple-sources multicasting simultaneously would be an important area of future research.

REFERENCES

- [1] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum cost multicast: A decentralized approach based on network coding," in *Proc. IEEE Infocom 2005*, March 2005.
- [2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, 2000.
- [3] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. 2003 International Symposium on Information Theory*. IEEE, 2003.

- [4] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *41st Allerton Annual Conference on Communication, Control, and Computing*, Monticello, IL, October 2003.
- [5] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *41st Allerton Annual Conference on Communication, Control, and Computing*, Monticello, IL, October 2003.
- [6] 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.
- [7] 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.
- [8] 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)*, October 2004.
- [9] R. Johari and J. Tsitsiklis, "Efficiency loss in a network resource allocation game," *Mathematics of Operations Research*, vol. 29, no. 3, pp. 407–435, 2004.
- [10] D. Acemoglu and A. Ozdaglar, "Flow control, routing and performance from service provider viewpoint," *MIT-LIDS-WP-1696 Technical Report*, MIT-LIDS, December 2003.
- [11] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operations Research Society*, vol. 49, pp. 237–252, 1998.
- [12] K. Jain, V. V. Vazirani, and Y. Yu, "Market equilibria for homothetic, quasi-concave utilities and economies of scale in production," in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, January 2005.
- [13] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.
- [14] T. Roughgarden and E. Tardos, "How bad is selfish routing?" *Journal of the ACM*, vol. 49, no. 2, pp. 236–259, March 2002.
- [15] C. Papadimitriou, "Algorithms, games, and the Internet," in *Proc. 33rd Annual ACM Symposium on the Theory of Computing*, 2001, pp. 749–753.
- [16] S. C. Dafermos and F. T. Sparrow, "The traffic assignment problem for a general network," *Journal of Research of the National Bureau of Standards, Series B*, vol. 73B, no. 2, pp. 91–118, 1969.
- [17] M. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*. Yale University Press, 1956.
- [18] T. Roughgarden, "The price of anarchy is independent of the network topology," in *Annual ACM Symposium on the Theory of Computing*, 2002, pp. 428–437.
- [19] J. Correa, A. Schulz, and N. Stier-Moses, "Selfish routing in capacitated networks," *Mathematics of Operations Research*, vol. 29, no. 4, pp. 961–976, November 2004.
- [20] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proc. ACM-SIAM Symposium on Discrete Algorithms SODA-2004*, January 2004.
- [21] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE/ACM Transactions on Networking*, pp. 274–285, April 2004.
- [22] D. P. Bertsekas, *Nonlinear programming*. Belmont, MA: Athena Scientific, 1995.
- [23] L. Kleinrock, *Queueing Systems*. John Wiley and Sons, 1976, vol. 2.
- [24] D. Gamarnik and A. Zeevi, "Validity of heavy traffic steady-state approximations in generalized Jackson networks," *Annals of Applied Probability (to appear)*, vol. 16, no. 1, pp. 56–90, 2006.
- [25] P. Gupta and P. R. Kumar, "A system and traffic dependent adaptive routing algorithm for ad hoc networks," in *Proc. IEEE 36th Conf. on Decision and Control*, San Diego, 1997, pp. 2375–2380.
- [26] P. Gupta, "Design and performance analysis of wireless networks," Ph.D. dissertation, University of Illinois at Urbana-Champaign, August 2000.
- [27] V. Borkar and P. R. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 382–396, March 2003.
- [28] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," Bell Labs Technical Report, Tech. Rep., August 2005.