Group Strategyproof Multicast in Wireless Networks

Ajay Gopinathan, Zongpeng Li, and Baochun Li

Abstract—We study the dissemination of common information from a source to multiple nodes within a multihop wireless network, where nodes are equipped with uniform omni-directional antennas and have a fixed cost per packet transmission. While many nodes may be interested in the dissemination service, their valuation or utility for such a service is usually private information. A desirable routing and charging mechanism encourages truthful utility reports from the nodes. We provide both negative and positive results towards such mechanism design. We show that in order to achieve the group strategyproof property, a compromise in routing optimality or budget-balance is inevitable. In particular, the fraction of optimal routing cost that can be recovered through node charges cannot be significantly higher than $\frac{1}{2}$. To answer the question whether constant-ratio cost recovery is possible, we further apply a primal-dual schema to simultaneously build a routing solution and a cost sharing scheme, and prove that the resulting mechanism is group strategyproof and guarantees $\frac{1}{4}$-approximate cost recovery against an optimal routing scheme.

Index Terms—Mechanism Design, Wireless Networks, Game Theory, Linear Programming, Approximation Algorithms, Theory

1 Introduction

A wireless *ad hoc* network consists of self-organizing wireless nodes, who must cooperate to route data for each other in the absence of a fixed network infrastructure. We consider ad hoc networks in the setting where wireless nodes are both autonomous and selfish. Each node is equipped with an omni-directional antenna, and expends resources such as energy and processing time when routing packets. As such, nodes may demand a fixed payment for each unit of information transmitted. We focus on the case when a subset of nodes are interested in obtaining identical information such as a media streaming service from a designated source node. In this scenario, nodes are faced with two major challenges. First, nodes must compute an efficient *routing solution* that obtains the information from the source node while minimizing transmission costs. A natural and attractive solution to this problem is *multicast*, which is efficient in terms of both bandwidth usage and transmission cost. Second, nodes must also decide on an appropriate and equitable scheme to distribute the multicast cost amongst themselves. Designing *cost-sharing* schemes that adhere to well defined notions of fairness and economic feasibility is a classic problem in economic theory [1]–[3].

Computing appropriate cost-shares becomes especially challenging when nodes are selfish, and the utility obtained for receiving the multicast is private information known only to the node itself. Wireless nodes may then misreport their willingness to pay for the service, in the hope of being charged less. In such a non-cooperative scenario, the goal is to design a mechanism that ensures nodes have no incentive to lie about their utility. Such a mechanism is said to be *strategyproof*. A strategyproof mechanism that is in addition robust against collusion by nodes is said to be *group strategyproof*. Almost all known group strategyproof mechanisms are based on the seminal work of Moulin and Shenker [1]. The crucial ingredient underlying a Moulin-Shenker mechanism is a cost-sharing scheme that is *cross-monotonic*. A cost-sharing scheme is said to be cross-monotonic if the cost share of a node does not increase when the service set containing the node expands. Using a simultaneous Cournot tatonnement process, Moulin and Shenker proved that cross-monotonic cost-shares give rise to group strategyproof mechanisms. Moreover, under reasonable notions of fairness, Immorlica, Mahdian and Mirrokni showed that the converse is true as well [4]. Motivated by this, group strategyproof mechanisms have been fashioned via the design of cross-monotonic cost-sharing schemes for a plethora of games, including minimum spanning tree [5], facility location [6], Steiner forests [7] and multicast in wired networks [8].

While the key to achieving group strategyproofness lies in a cross-monotonic cost-sharing scheme, at first glance, such a property does not seem difficult to achieve. Requiring only cross-monotonicity, it is easy to design a cost-sharing scheme that is either trivial (offering the service for free), or unfair (charging everyone a fixed price that is too high). Indeed, in most practical situations, we simultaneously require the cost-sharing scheme to be *competitive* and *budget-balanced*. A cost sharing scheme is competitive if no subset of nodes is charged more than the optimal cost of serving this subset alone. Such a requirement ensures that there is no threat of secession by some subset of nodes, who may instead choose to obtain the service from another

• *A. Gopinathan and Z. Li are with the Department of Computer Science, University of Calgary, Canada. B. Li is with the Department of Electrical and Computer Engineering, University of Toronto, Canada*
E-mail: {ajay.gopinathan, zongpeng}@ucalgary.ca, bli@eecg.toronto.edu

provider charging less. The budget-balance requirement is natural — a reasonable mechanism should seek to recoup the cost incurred by the routing solution. From a computational perspective, we are further interested in cost sharing schemes that are competitive and budget-balanced with respect to the min-cost routing solution.

In this paper, we design cost sharing schemes for information dissemination in a wireless ad hoc network, when the underlying charging scheme is required to be group strategyproof. Simultaneously, we require the data delivery method employed to be efficient in terms of routing costs. Any efficient routing mechanism should exploit the following two important properties; (1) the *broadcast advantage* inherent in wireless environments, and (2) the *replicable property* of information. A natural data dissemination method that suggests itself is *multicast*. The optimal multicast route ensures that there are no redundant transmissions by any node, thus ensuring the total cost of wireless transmissions is minimized.

We show that cross-monotonic, competitive and budget-balanced cost sharing schemes *do not exist* for multicast in wireless networks. Hence, we relax the budget-balance requirement, to obtain a cross-monotonic, *approximately* budget-balanced cost-sharing scheme[1]. This guarantees a truthful mechanism, to the detriment of the cost recovery ratio. An interesting question is then to find upper and lower bounds on cost recovery. For a wireless network with T multicast receivers, we show that the budget-balance ratio for *any* cross-monotonic cost-sharing scheme is upper bounded by $\frac{1}{2}+O(\frac{1}{T})$, which is asymptotically constant. In the special case of uniform transmission costs, the upper bound is $\frac{2}{3}+O(\frac{1}{T})$. Our result hinges on a pathological network construction, and we employ a probabilistic argument similar to that of Immorlica *et al.* [4] and Li [8]. We complement this upper bound by showing that constant factor budget-balanced schemes are possible. We design an algorithm that computes a 2-approximate routing solution, and show how we can modify this algorithm to guarantee a cost recovery ratio of at least $\frac{1}{4}$ of the total cost of multicast in a wireless ad hoc network. Our technique is based on the primal-dual schema [6], [9], [10], and is unique in that it ensures cross-monotonicity by continuously increasing dual variables, which results in violated dual constraints. This results in an infeasible dual vector. Nevertheless, we show that the recovered cost shares is bounded with respect to the feasible dual.

The rest of this paper is organized as follows; in Section 2, we discuss related work. We introduce our network model as well as some game theoretic definitions in Section 3. In Section 4, we argue using a probabilistic method that perfect budget-balance in wireless networks is impossible, and derive upper bounds on cost recovery. We design a primal-dual based algorithm that computes cross-monotonic cost-shares for wireless networks with uniform cost in Section 5, and prove its performance bound, before concluding in Section 6.

1. Relaxing the budget-balanced requirement is equivalent to relaxing the competitiveness property. See Section 3.

2 Related Work

The study and design of group strategyproof mechanisms was initiated by the seminal work of Moulin [2] and Moulin and Shenker [1], in which they showed that the Cournot tatonnement under a cross-monotonic cost-sharing scheme gives rise to mechanisms that are group strategyproof. Further, they show that if the cost function is submodular, then it is possible to achieve cross-monotonic cost sharing with the Shapley value [11]. In a Moulin-Shenker mechanism, the service is offered in the beginning to all interested agents at prices computed using some cost-sharing scheme. Agents that are unwilling to meet the price imposed are removed from the service set, new cost-shares are computed, and the service is offered to the remaining agents. The process repeats until all agents agree to meet the asking price of the mechanism. If the underlying cost-sharing scheme is cross-monotonic, the dominant strategy of every agent, whether acting individually or in conspiracy with other agents, is to report her true valuation for the service. Inspired by their work, group strategyproof mechanisms have been developed for various games through the design of cross-monotonic, competitive and approximately budget-balanced cost-sharing algorithms. The minimum spanning tree [5], the travelling salesman problem [5], facility location [6], single-source rent-or-buy [12] and Steiner forest [7] all constitute combinatorial optimization games for which algorithms have been developed for computing cost shares with the previously stated properties.

With the notable exception of the minimum spanning tree game, a recurring theme in the cost-sharing schemes for the previously mentioned games is the poor budget-balance ratio. Using a novel probabilistic argument, Immorlica *et al.* [4] prove upper bounds on cost recovery for various games, including edge and vertex cover, set cover and the metric facility location game. Further, Immorlica *et al.* showed that under the reasonable assumptions of *no free riders* and *upper continuity*, group strategyproof mechanisms give rise to cross-monotonic cost-sharing schemes. Recently, Mehta *et al.* [13] consider weakening the group strategyproof property with the aim of improving the budget-balance ratio. They propose acyclic mechanisms with exponentially better budget-balance properties for some class of games, but are only weakly group strategyproof. Subsequently, Brenner and Shafer [14] showed how to turn any α-approximation algorithm for a combinatorial optimization problem into an α-budget-balanced acyclic mechanism.

Cross-monotonic cost-sharing for optimal multicast with network coding [15] was studied by Li [8] for both directed and undirected networks. Similar to Immorlica *et al.* [4], Li used a probabilistic technique to show the existence of directed networks for which no cross-monotonic cost-sharing scheme recovers more than $O(\frac{1}{\sqrt{k}})$ of the cost, where k is the number of multicast receivers. For undirected networks, the upper bound was shown to be $O(\frac{1}{2})$. In a preliminary version of this paper [16], we showed that the cross-monotonic cost recovery upper bound for wireless networks was $\frac{1}{2}+O(\frac{1}{\eta})$ where η was a network dependent parameter, and further designed an algorithm for computing a $\frac{1}{8}$-budget-balanced cost-sharing scheme for uniform cost networks. In this paper, we improve these results by deriving cost recovery upper and lower bounds of $\frac{1}{2}+O(\frac{1}{T})$ and $\frac{1}{4}$ respectively, where T is the number of multicast receivers.

3 PRELIMINARIES

In this section, we will introduce the wireless network model we use, and discuss some desirable properties of cost-sharing schemes. We will also show that the optimal multicast cost in wireless networks is not submodular, thus precluding the use of the Shapley value [11] as a viable cost-sharing scheme for group strategyproofness.

3.1 The Network Model

We assume that the wireless networks we study can be modeled by disk graphs with some uniform radius, r. In such graphs, a wireless node u is connected to all nodes whose physical distance from u is less than r. The *broadcast property* of wireless networks means that a transmission by u can be heard by all other nodes within range r of u. We will say v is in the *neighbourhood* of u or is *adjacent* to u if v is within u's transmission radius. Each node charges a fixed price to transmit a unit of information, and we denote the cost of transmitting a unit of information via node u as $c(u)$. We will use $d(u,v)$ to denote the cheapest cost path from node u to node v, including the cost of u's transmission. We assume that there is a distinguished source node s, with identical data to be sent to a set of receivers $\mathcal{T}$. To exploit the replicable property of information and efficiently utilize bandwidth, the data delivery mechanism employed by s will be multicast. Optimal multicast is equivalent to computing the optimal Steiner tree in a network. Since Steiner trees are NP-Hard to compute [17], we will compute an approximately optimal Steiner tree instead.

3.2 Cross-Monotonic Cost Sharing Schemes

Consider the following problem: a set $\mathcal{U}$ of agents are interested in obtaining a service from a service provider. For any set $\mathcal{S} \subseteq \mathcal{U}$ of agents, there is a *cost* of providing this service to $\mathcal{S}$. Let $C_{OPT}(\mathcal{S})$ denote the optimal (*i.e.* cheapest) cost of serving $\mathcal{S}$. Agents share the cost of obtaining the service, and each agent $i \in \mathcal{U}$ has some *private* valuation, v_i, which is the maximum amount she is willing to pay for the service. In such a scenario, we are interested in designing a *cost-sharing mechanism* that solves the following two problems; (1) deciding the set of agents $\mathcal{S} \subseteq \mathcal{U}$ that should receive the service, and (2) deciding the *cost-share* of agent i in the set $\mathcal{S}$, denoted as

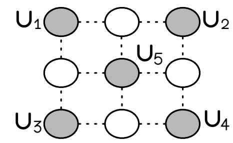

Fig. 1. Multicast in wireless networks is not submodular

$\xi(i,\mathcal{S})$. The mechanism solicits bids b_i from each agent $i \in \mathcal{U}$, computes $\mathcal{S}$, and allocates the cost so that for each $i \in \mathcal{S}$, $\xi(i,\mathcal{S}) \le b_i$. We assume each agent i is selfish, and thus wishes to maximize her utility, which is $u_i = v_i - \xi(i,\mathcal{S})$ if $i \in \mathcal{S}$, and 0 otherwise. Agent i may *lie* about her valuation, and bid $b_i \ne v_i$ if doing so yields higher utility. As such, we require that the cost-sharing mechanism is *truthful*, that is, the *dominant strategy* [3] for every agent is to bid her true valuation $b_i = v_i$. A Moulin-Shenker mechanism [1], coupled with a *cross-monotonic* cost-sharing scheme, ensures that truthful bidding is the dominant strategy for every agent, even when agents are allowed to act in collusion with other agents. Formally, a cross-monotonic cost-sharing scheme for some agent i in the set $\mathcal{A}$ has the following property

$$\xi(i,\mathcal{A}) \le \xi(i,\mathcal{B}) \quad \forall \mathcal{B} \supseteq \mathcal{A} \tag{1}$$

Essentially, an agent i in some service set is guaranteed that her current cost-share will never increase when the service set expands, if a cross-monotonic cost-sharing scheme is used. It is also further desirable that the computed cost-shares possess the following properties:

- **Competitiveness** To ensure agents do not switch to another provider, the cost-sharing scheme should not overcharge users, that is:
$$\textstyle\sum_{i\in\mathcal{S}} \xi(i,\mathcal{S}) \le C_{OPT}(\mathcal{S})$$
- **Budget-balance** The cost-sharing scheme should recover the full cost of the solution, that is:
$$\textstyle\sum_{i\in\mathcal{S}} \xi(i,\mathcal{S}) \ge C_{OPT}(\mathcal{S})$$

However, many games of interest lack cost-sharing schemes that are simultaneously cross-monotonic, competitive and budget-balanced [4]. One can relax the budget-balance requirement, to obtain a competitive, *approximately* budget-balanced scheme. A cost-sharing scheme is said to be competitive and β-budget-balanced for $0 \le \beta \le 1$ if the following holds instead:

$$\beta\, C_{OPT}(\mathcal{S}) \le \sum_{i\in\mathcal{S}} \xi(i,\mathcal{S}) \le C_{OPT}(\mathcal{S}) \tag{2}$$

Alternatively, one may relax the competitiveness requirement instead. A budget-balanced, α-competitive cost sharing scheme for $\alpha \ge 1$ is one that obeys the following

$$C_{OPT}(\mathcal{S}) \le \sum_{i\in\mathcal{S}} \xi(i,\mathcal{S}) \le \alpha\, C_{OPT}(\mathcal{S}) \tag{3}$$

A competitive, β-budget-balanced cost-sharing scheme is equivalent to a budget-balanced, $\frac{1}{\beta}$-competitive cost-sharing scheme. In the sequel, a *cross-monotonic* and *approximately budget-balanced* cost sharing scheme will be taken to also mean one that is, in addition, *competitive*.

3.3 Wireless Multicast Cost is Non-Submodular

One possible approach to cost sharing is to allocate the multicast cost according to the Shapley value [11], which is a well known approach in economic theory for equitable and fair allocation of goods. The Shapley value essentially charges each agent i the *marginal cost* of serving i, in expectation over *all possible orderings* of the set of agents. More formally, the cost share of agent i according to the Shapley value is

$$\xi(i,\mathcal{T}) = \sum_{\mathcal{S}\subseteq\mathcal{T}\setminus\{i\}} \frac{|\mathcal{S}|!(|\mathcal{T}|-|\mathcal{S}|-1)!}{|\mathcal{T}|!}\Big(C_{OPT}(\mathcal{S}\cup\{i\})-C_{OPT}(\mathcal{S})\Big)$$

It turns out that if the cost function $C_{OPT}(.)$ is *submodular*, then the Shapley value is both cross-monotonic and perfectly budget-balanced (see Moulin and Shenker [1] for details). A function f is said to be *submodular* if for all $\mathcal{A}\subset\mathcal{B}$ and for some $i\notin\mathcal{B}$, the following holds

$$f(\mathcal{B}\cup\{i\}) - f(\mathcal{B}) \le f(\mathcal{A}\cup\{i\}) - f(\mathcal{A}) \quad (4)$$

A *cost function* that is submodular intuitively means that the marginal cost of servicing a new agent decreases as the service set expands. Unfortunately, the cost function for multicast in wireless networks is not submodular, thus precluding the use of the Shapley value to compute cost shares that are cross-monotonic and budget-balanced. Consider the example network shown in Fig. 1. Assume that u_3 is the multicast source, and let $c(u_i)=1$ for all i. Then

$$C_{OPT}(\{u_1,u_5\}) = 2 \qquad C_{OPT}(\{u_5\}) = 2$$
$$C_{OPT}(\{u_1,u_4,u_5\}) = 3 \qquad C_{OPT}(\{u_4,u_5\}) = 2$$

Letting $\mathcal{A}=\{u_5\}$, $\mathcal{B}=\{u_4,u_5\}$, $i=u_1$, we can see from (4) that wireless multicast cost is not submodular.

4 Cost Recovery Upper Bound

In this section, we will show that for wireless networks, cost-sharing schemes that distribute the optimal (*i.e.* minimum) multicast routing cost in a cross-monotonic fashion cannot be budget-balanced. We first prove this for a simple topology, which provides an intuition into why the *broadcast advantage* restricts cross-monotonic cost recovery. Subsequently, we generalize the ideas used to show that the upper bound on cost recovery in networks with T multicast receivers is at most $\frac{1}{2}+O(\frac{1}{T})$.

4.1 Example topology with $\frac{3}{4}$-budget-balance bound

Consider once again the network shown in Fig. 1. Assume that the source node is u_5, which has zero transmission cost, and let all other nodes have cost $c(u_i)=1$ to transmit a unit of information. Let $\{u_1,u_2,u_3,u_4\}$ be potential multicast receivers, and let $\mathcal{A}_1=\{u_1,u_2\}$, $\mathcal{A}_2=\{u_2,u_3\}$, $\mathcal{A}_3=\{u_3,u_4\}$ and $\mathcal{A}_4=\{u_1,u_4\}$. Now, choose at random any set A_i as the target multicast receiver set. Since the network is symmetric and $\mathcal{A}_i$ is randomly chosen, for *any* given budget-balanced cost sharing scheme, each node in $\mathcal{A}_i$ will pay at most 1/2 *in expectation*. Now consider multicasting to some set $\{\{u_j\}\cup\mathcal{A}_i\}$ for any i and $u_j\notin\mathcal{A}_i$. Since there is a node in $\mathcal{A}_i$ which together with u_j forms a multicast set in which u_j pays at most $\frac{1}{2}$, by cross-monotonicity, u_j will also only pay at most 1/2 when in the superset $\{\{u_j\}\cup\mathcal{A}_i\}$. However, the multicast cost for this set is 2, hence, the cost recovery ratio in expectation is at most $\frac{\frac{1}{2}+\frac{1}{2}+\frac{1}{2}}{2}=\frac{3}{4}$

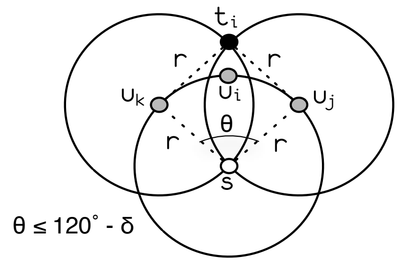

Fig. 2. Placement of relays u_i and potential receivers t_i for topology with budget-balance at most $\frac{1}{2}+O(\frac{1}{T})$

4.2 Poor cost recovery for wireless networks

We now generalize the argument in the previous section to wireless networks that can be modeled by disk graphs with uniform radius. We construct a pathological network that does not admit a cost-sharing scheme that is cross-monotonic and perfectly budget-balanced for any optimal minimum cost multicast.

Theorem 1. *There exists a wireless network that can be modeled by unit disk graphs which does not admit a cost-sharing scheme for optimal multicast that is cross-monotonic and* $(\frac{1}{2}+O(\frac{1}{T})+\epsilon)$*-budget-balanced for any* $\epsilon>0$*, where* T *is the number of potential multicast receivers.*

Proof: Assume that the source node s has transmission cost $c(s)=x$. Arrange relay nodes u_i, $i=1\ldots T$, with $c(u_i)=y$, at equidistant points from each other along the circumference of s's neighbourhood. Fix a relay node u_i, and let u_j and u_k be two other relay nodes with equal distance from u_i, and such that u_j, u_i and u_k subtends an angle of at most $120°-\delta$ for some $\delta>0$ (see Fig. 2). For each such set of nodes u_j, u_i and u_k, place a receiver node t_i at the point outside of s's neighbourhood where the coverage area of u_j and u_k intersect. By construction, t_i is within the neighbourhood of all relay nodes between u_j and u_i, as well as u_i and u_k. Hence, each t_i is connected to $\eta=\lfloor(\frac{120-\delta}{360})T\rfloor=O(T)$ relay nodes. Symmetrically, each relay node is also connected to exactly η receivers. To prove the budget-balance ratio of this network, pick a receiver t' at *random* and include the next η receiver nodes in the clockwise direction, and label this as the target multicast group, $\mathcal{T}$. Denote the set consisting of t' and the next $\eta-1$ nodes in $\mathcal{T}$ as $\mathcal{A}$, and denote by $\mathcal{B}$ the set of η nodes $\mathcal{T}\setminus\{t'\}$. Observe that receivers in $\mathcal{A}$ (resp. $\mathcal{B}$) have one relay node in common, call this $u_{\mathcal{A}}$ (resp. $u_{\mathcal{B}}$). Since $\mathcal{T}$ is picked at random, we can bound the expected cost share of each receiver as follows:

$$\begin{aligned} E[\sum_{t_i\in\mathcal{T}}\xi(t_i,\mathcal{T})] &= E[\sum_{t_i\in\mathcal{B}}\xi(t_i,\mathcal{T})]+E[\xi(t',\mathcal{T})] \\ &\le E[\sum_{t_i\in\mathcal{B}}\xi(t_i,\mathcal{B})]+E[\xi(t',\mathcal{A})] \\ &= \eta\frac{x+y}{\eta}+\frac{x+y}{\eta}=(x+y)(1+\frac{1}{\eta}) \end{aligned}$$

The first equality is from linearity of expectations. The next inequality follows from cross-monotonicity, since $\mathcal{A},\mathcal{B}\subset\mathcal{T}$. Since $C_{OPT}(\mathcal{T})=x+2y$ and $|\mathcal{A}|=|\mathcal{B}|=\eta$, the fraction of cost recovered is at most

$$\frac{(x+y)(1+\frac{1}{\eta})}{x+2y}$$

Setting the cost $x = 0$ and $y = 1$ yields the theorem. □

The proof of Theorem 1 also implies the following:

Corollary 1. *For uniform cost wireless networks that can be modeled by unit disk graphs, there is no cross-monotonic, $(\frac{2}{3} + O(\frac{1}{T}) + \epsilon)$-budget-balanced cost-sharing scheme.*

5 Computing cross-monotonic cost-shares

In this section, we will design an algorithm for computing cost-shares that are cross-monotonic for multicast transmission in wireless ad hoc networks with arbitrary topologies. In particular, we will build a Steiner tree for efficient multicast, and design a scheme to distribute the cost of the Steiner tree to multicast users in a cross-monotonic fashion. Our techniques are grounded in the linear programming based primal-dual framework for approximation algorithms, first introduced by Goemans and Williamson [9]. The primal-dual schema employed not only facilitates the construction of a good routing solution, but also provides a natural means for computing cost shares that are cross-monotonic. The latter can be achieved by treating the variables in the dual solution as cost shares.

Our strategy will be to first focus on computing a good multicast routing solution. In particular, the primal-dual algorithm we design will build a 2-approximate solution. However, the dual variables obtained by this algorithm cannot be used directly as a cross-monotonic cost-sharing scheme. Instead, we will modify this algorithm in Section 5.3 to compute cost-shares that are cross-monotonic, and $\frac{1}{4}$-budget-balanced.

5.1 The primal-dual schema

We begin by introducing the primal-dual framework within the context of multicast in wireless networks. We focus on the case when each node has uniform transmission radius, as well as uniform cost to transmit a packet. The former assumption allows us to assume that links between nodes are *symmetric*, while the latter admits $c(u) = c(v), \forall u, v$ without loss of generality.

Before we can describe the algorithm, we will require some definitions and notations. For a set of nodes $\mathcal{S}$, we define the binary function $f : 2^{\mathcal{V}} \to \{0, 1\}$, where V is the set of all nodes in the network, such that for any $\mathcal{S} \subseteq \mathcal{V}$, we have the following:

$$f(\mathcal{S}) = \begin{cases} 1 & \text{if } |\mathcal{S} \cap \mathcal{T}| \geq 1 \text{ and } s \notin \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

That is, $f(\mathcal{S}) = 1$ implies that the set of nodes $\mathcal{S}$ contains at least one multicast receiver, and does *not* contain the source node s. We define a *node cut* of the set $\mathcal{S}$ as the minimal set of nodes $\delta(\mathcal{S})$, such that following conditions hold

- $\mathcal{V} \setminus \delta(\mathcal{S})$ induces a graph such that the set of nodes $\mathcal{S}$ and $\bar{\mathcal{S}}$ are disconnected.
- If $u \in \delta(\mathcal{S})$, then there exists a $w \in \mathcal{S}$ such that u and w are adjacent.

A node cut for $\mathcal{S}$ is essentially the minimal set of nodes that are adjacent to $\mathcal{S}$, and whose removal from the network disconnects $\mathcal{S}$ from the source node.

The definitions of the binary function f together with the notion of a node cut, allows us to succinctly state the problem of computing the optimal Steiner tree in a symmetric, wireless network as a linear integer program (IP) of the following form:

$$\text{Minimize} \quad \sum_{u \in \mathcal{V}} c(u)x(u) \tag{5}$$

Subject To

$$\sum_{u \in \delta(\mathcal{S})} x(u) \geq f(\mathcal{S}) \qquad \forall \mathcal{S} \subseteq \mathcal{V} \tag{5a}$$

$$x(u) \in \{0, 1\} \qquad \forall u \in \mathcal{V} \tag{5b}$$

In IP (5), the binary variable $x(u)$ indicates if node u should be used to transmit information in the optimal Steiner tree connecting s to each node in $\mathcal{T}$. The objective function tries to minimize the total cost of transmitting nodes. The first constraint ensures that for any set of nodes $\mathcal{S}$ for which $f(\mathcal{S}) = 1$, (*i.e.* $\mathcal{S}$ contains at least one receiver but not the source node), at least one of the nodes that disconnects $\mathcal{S}$ and s should be included in the Steiner tree. Since this constraint is stated for all possible sets of nodes, it is easy to see that a solution x that satisfies all the constraints also ensures that each node in $\mathcal{T}$ will be connected to s.

Computing a Steiner tree directly using IP (5) is intractable, since the number of constraints is exponential in the number of nodes in the network. Instead, we will resort to the primal-dual technique based on linear programming theory [9] to solve for an approximately optimal Steiner tree. The primal-dual schema will employ the linear program (LP) relaxation of IP (5), achieved by relaxing the integrality requirement to merely requiring that $x(u) \geq 0$ for all u. This gives us the *primal* linear program, for which we can subsequently formulate the following *dual* linear program:

$$\text{Maximize} \quad \sum_{\mathcal{S} \subseteq \mathcal{V}} f(\mathcal{S})y(\mathcal{S}) \tag{6}$$

Subject To

$$\sum_{y(\mathcal{S}): u \in \delta(\mathcal{S})} y(\mathcal{S}) \leq c(u) \qquad \forall u \in \mathcal{V} \tag{6a}$$

$$y(\mathcal{S}) \geq 0 \qquad \forall \mathcal{S} \subseteq \mathcal{V} \tag{6b}$$

We know from the *weak duality theorem* [18] of linear programming theory, the objective function of the LP relaxation of (5) is lower bounded by the objective function of the dual LP in (6). Recall also that from linear programming duality theory, the optimal solutions to the primal and dual LPs are related via *complementary slackness conditions* [18]. In particular, whenever a constraint of type (6a) holds with equality in the optimal dual solution, then it follows that the corresponding primal variable $x(u) > 0$. The primal-dual schema essentially seeks to exploit the properties just stated. It begins with a feasible dual solution in which $y(\mathcal{S}) = 0$ for all $\mathcal{S} \subseteq \mathcal{V}$, and an infeasible primal solution where $x(u) = 0$ for all $u \in \mathcal{V}$. The algorithm then attempts to construct a

Algorithm 1: Primal-Dual Algorithm

Initialize $x(u) = 0 \ \forall u$, set time $\tau = 0$
Update solution set $\mathcal{W} = \emptyset$

Phase 1:

1) Update set of *active* components $\mathcal{C}$ using Definition 1
2) If $\mathcal{C} = \emptyset$, go to Phase 2.
3) Increase $y(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{C}$ and τ at uniform rate until constraint (6a) becomes tight for some u
4) Set $x(u) = 1$, and add u to $\mathcal{W}$. Repeat step 1).

Phase 2:

1) Examine nodes in $\mathcal{W}$ in an arbitrary, predetermined order.
2) For each $u \in \mathcal{W}$, if graph induced by $\mathcal{W} \cup \mathcal{T} \setminus \{u\}$ does not disconnect s and some $t \in \mathcal{T}$, set $x(u) := 0$, $\mathcal{W} := \mathcal{W} \setminus \{u\}$.

Phase 3:

1) Update set of *unsatisfied* components $\mathcal{C}$ using Definition 2
2) If $\mathcal{C} = \emptyset$, go to Phase 4.
3) Increase $y(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{C}$ and τ at uniform rate until constraint (6a) becomes tight for some u
4) Set $x(u) = 1$, and add u to $\mathcal{W}$. Repeat step 1).

Phase 4:

1) Examine nodes in $\mathcal{W}$ in an arbitrary, predetermined order.
2) For each $u \in \mathcal{W}$, if the graph induced by $\mathcal{W} \setminus \{u\}$ does not disconnect s and some $t \in \mathcal{T}$, set $x(u) := 0$, $\mathcal{W} := \mathcal{W} \setminus \{u\}$.

feasible primal solution, by increasing judiciously selected dual variables in a controlled manner. Whenever this dual increase results in a dual constraint of type (6a) holding with equality, we set the corresponding primal variable $x(u) = 1$. This process repeats until the primal solution constructed is feasible. At the end, the primal solution is used to obtain a feasible tree connecting s to all multicast receivers in $\mathcal{T}$. The crucial observation here is that the dual variable $y(\mathcal{S})$ can be interpreted as *cost-shares* for the set of nodes $\mathcal{S}$. We will leverage this property later to compute cost-shares from the dual solution that are cross-monotonic while having a constant budget-balance ratio.

5.2 A 2-approximate Steiner tree construction

We next turn to describing our primal-dual based algorithm for computing an approximately optimal routing solution in wireless networks. We first require some terminology. We will say a constraint is *tight* if (6a) holds with equality for some node u. A node is said to be *open* if $x(u) = 1$. In the beginning, all nodes are said to be closed, that is $x(u) = 0$ for all nodes u. We will require the following definition of a *component*:

Definition 1 *A component is a set of nodes $\mathcal{S}$ that meet the following conditions:*

- *$\mathcal{S}$ is connected, that is for any $u, v \in \mathcal{S}$ there is a path from u to v using only nodes in S,*
- *if $u \in \mathcal{S}$, then either $x(u) = 1$ or $u \in \mathcal{T}$*

A component $\mathcal{S}$ is said to be *satisfied* if $f(S) = 0$, *i.e.*, S includes s. Since $x(u) = 0$ for all u in the beginning of the algorithm, we will initially have $|\mathcal{T}|$ unsatisfied components, each consisting of a receiver $t \in \mathcal{T}$.

Algorithm 1 shows our algorithm for computing a 2-approximate Steiner tree in symmetric wireless networks. Similar to most primal-dual schemas in the literature, it will be necessary to introduce a notion of time, τ into our algorithm. Unlike other primal-dual schemas however, our algorithm consists of 4 phases. The first phase consists of carefully increasing or *growing* selected dual variables, in an attempt to build a routing solution. Growing dual variables leads to opening nodes, due to complementary slackness conditions. Let $\mathcal{W}$ be the set of open nodes at any time during this dual growing phase. Then this phase ends when the set of nodes $\mathcal{W} \cup \mathcal{T}$ induces a subgraph such that the source node is connected to each multicast receiver. This first phase of dual variable growth may however result in more open nodes than is necessary for a feasible routing solution. Hence, the second phase consists of "pruning" these superfluous nodes by removing them from the set $\mathcal{W}$, without sacrificing the feasibility of the solution computed thus far. At the end of the pruning phase, the set $\mathcal{W} \cup \mathcal{T}$ induces a tree, connecting s to $\mathcal{T}$. Note however that dual has only paid for nodes in $\mathcal{W}$, and from Definition 1, we may have tacitly included receivers in the tree $\mathcal{W} \cup \mathcal{T}$ that transmit for "free". Therefore, phase 3 repeats the dual growing process, this time obtaining enough dual payment to pay for transmitting receivers that were not opened in the first dual growing phase. The final phase once again performs node pruning to remove nodes that were unnecessarily added to the solution $\mathcal{W}$ in during the dual growing process in Phase 3. We next describe each phase in detail.

Phase 1: Dual Growth I - In this phase, a component is said to be *active* if it is not satisfied. Let $\mathcal{W}$ consist of the set of open nodes. Initially, at time $\tau = 0$, we have $\mathcal{W} = \emptyset$, and each multicast receiver forms a standalone, active component. For each active component $\mathcal{S}$, we grow its dual variable $y(\mathcal{S})$, *uniformly in time* τ, until some constraint of type (6a) corresponding to some node u goes tight. At this point, we stop dual growth, and open node u by setting $x(u) = 1$ and adding u to $\mathcal{W}$. We then update the set of active components, and begin growing dual variables for this latest set of active components. This phase ends when there are no longer any active components. It is easy to see that at the end of this phase, we will have a single, satisfied component consisting of nodes in the set $\mathcal{W} \cup \mathcal{T}$. Since nodes have uniform transmission radius, the links between nodes are symmetric, and the set of nodes $\mathcal{W} \cup \mathcal{T}$ induces a subgraph where each receiver is connected to the source node. It is crucial to note however, that only the nodes in $\mathcal{W}$ have been "paid" for thus far by the dual variables.

Phase 2: Pruning I - Increasing the dual variable $y(\mathcal{S})$ contributes to reducing the slack in the constraints for all nodes $u \in \delta(\mathcal{S})$. Hence, the dual growing phase essentially opens more nodes than is necessary to connect $\mathcal{S}$ and $\bar{\mathcal{S}}$, and these unnecessary nodes may be pruned. Consequently, in this phase, we will examine only nodes in the set $\mathcal{W} \setminus \mathcal{T}$ for pruning – later we will show that our choice of this set helps to preserve cross-monotonicity when computing cost shares. For each node we examine, we set $x(u) = 0$ and remove u from $\mathcal{W}$ if this can be done without disconnecting any receiver from the source node in the subgraph induced by $\mathcal{W} \cup \mathcal{T}$. Essentially, we try and close any node that is not a multicast receiver, without sacrificing connectivity induced by the union of the remaining set of open nodes and $\mathcal{T}$. Clearly, by the end of the pruning phase, the set of nodes $\mathcal{W} \cup \mathcal{T}$ induces a tree connecting the source node to every receiver.

Phase 3: Dual Growth II - One can view the dual growing in Phase 1 in the following way - each component seeks to open a path of nodes to the source by paying for them. However, the dual variable of a component only contributes to node cuts adjacent to a component. This means that any point in the algorithm, all nodes in a component have been paid for, with the *possible exception* of the multicast receivers themselves. In order to remedy this situation, we define a component in this phase differently. Specifically, a component is instead now defined in the following way:

Definition 2 *A component is a set of nodes $\mathcal{S}$ that meet the following conditions:*

- *$\mathcal{S}$ is connected, that is for any $u, v \in \mathcal{S}$ there is a path from u to v using only nodes in S,*
- *if $u \in \mathcal{S}$, then $x(u) = 1$.*

That is, a component in this phase consists strictly of connected nodes in $\mathcal{W}$. The dual growing process in this phase then proceeds in a similar fashion to Phase 1. Once again, we grow the dual variables for unsatisfied components uniformly in time. Observe however that now unsatisfied components must be adjacent to at least one multicast receiver, and hence the increase in dual variables during this phase is bounded by the uniform cost $c(u)$. Once again, whenever a constraint of type (6a) goes tight for some node u, we stop the dual growing process, set $x(u) = 1$ and add u to $\mathcal{W}$. This process repeats until there are no longer any active components.

Phase 4: Pruning II - At the end of Phase 3, we are guaranteed that the set of nodes $\mathcal{W}$ induces a subgraph that connects s to each receiver in $\mathcal{T}$. Since the brief dual growing process in Phase 3 may have once again opened superfluous nodes, we prune unnecessary nodes in $\mathcal{W}$ without disconnecting any receiver from the source. The final solution $\mathcal{W}$ is thus a tree connecting s to each multicast receiver in $\mathcal{T}$.

The next theorem shows that the final solution $\mathcal{W}$ forms a 2-approximate Steiner tree connecting s and $\mathcal{T}$.

Theorem 2. $\sum_{u \in \mathcal{W}} c(u) \leq \sum_{S \subseteq V} 2y(\mathcal{S})$

Proof: There are two crucial ideas behind this lemma. The first is that dual variables for components grow at uniform rate. Second, the primal solution $\mathcal{W}$ induces a tree, and thus every receiver has exactly one path to the source. Consider any point in time during the dual growing phase on the network graph. Only active components are increasing their duals. Increasing the dual variable of component $\mathcal{S}$ never pays for nodes in the component, only to nodes in the node cut set adjacent to $\mathcal{S}$. Now, let us shrink each active component into a single node, and remove all other nodes outside of these components that do not appear in the final solution $\mathcal{W}$. The resulting graph (call it $\mathcal{H}$) now consists of nodes that are either active components, or nodes that will be paid for at some time in the future. Let us define the *degree* of an active component $\mathcal{S}$, as the nodes in $\mathcal{H}$ adjacent to $\mathcal{S}$, and let us denote it by $\deg(\mathcal{S})$. We claim that the average degree of all the active components is not more than 2. To see that this is indeed the case, recall that the final solution $\mathcal{W}$ is a tree. Therefore, every node in $\mathcal{H}$ adjacent to an active node, must either be in a path from the active node to the source, or to another active node. If an active node has paths in $\mathcal{H}$ to more than one active node, than each of those active nodes must have degree of 1 (otherwise, we would have redundant paths to active nodes). By definition, since nodes in $\mathcal{W}$ are open, its corresponding constraint must be tight, and so we get

$$\sum_{u \in \mathcal{W}} c(u) = \sum_{u \in \mathcal{W}} \Big(\sum_{\mathcal{S}: u \in \delta(\mathcal{S})} y(\mathcal{S}) \Big) = \sum_{\mathcal{S} \subseteq \mathcal{V}} \Big(\sum_{u: u \in \mathcal{W} \cap \delta(\mathcal{S})} y(\mathcal{S}) \Big)$$
$$= \sum_{\mathcal{S} \subseteq \mathcal{V}} \Big(\deg(\mathcal{S}) y(\mathcal{S}) \Big)$$

But from the previous argument, we know that average degree for all active components is at most 2, so we get

$$\sum_{u \in \mathcal{W}} c(u) \leq \sum_{\mathcal{S} \subseteq \mathcal{V}} 2y(\mathcal{S})$$

From the duality theorem in linear programming [18], the above implies that the solution $\mathcal{W}$ constructed is within a factor of 2 of the optimal Steiner tree. □

5.3 The cross-monotonic cost-sharing scheme

We next describe how to compute cross-monotonic cost shares for Steiner tree based information dissemination in wireless networks. In order to do so, we need to modify Algorithm 1. The crucial observation that we will employ is the following: the dual variable $y(\mathcal{S})$ can be interpreted as the cost of connecting the set $\mathcal{S}$ to $\bar{\mathcal{S}}$. Intuitively, cross-monotonic cost-shares can then be obtained through equitable distribution of the cost $y(\mathcal{S})$ to receivers $\mathcal{S} \cap \mathcal{T}$, coupled with the "smooth" increase of dual variables $y(\mathcal{S})$ during the dual growing phase of the primal-dual algorithm. A smooth growth of dual variables essentially avoids sudden arbitrary increases in the dual variables, which would harm cross-monotonicity. Unfortunately, the dual-growth of Phase 1 is *not* smooth, in the sense that dual variables cease to increase when components are satisfied.

In order to preserve cross-monotonicity, we need to ensure that dual variables do not cease to increase prematurely in Phase 1. Let $d(s,t)$ be the shortest path cost from the source node to receiver t. Let $\mathcal{S}^\tau$ be the set of nodes in component $\mathcal{S}$ at time τ. In our modified primal-dual algorithm, a component $\mathcal{S}$ is *active* at time τ in Phase 1 if the following condition holds

$$\max_{t \in \mathcal{S}^\tau \cap T} d(s,t) \geq \tau \tag{7}$$

Here, we depart from the usual primal-dual schema by continuing to increasing dual variables for satisfied components, as long as the condition in (7) holds. We will call a component's contribution after it becomes satisfied as its *ghost contribution* [6]. Note that at all times in the algorithm, at most one component's dual increase can be viewed as ghost contribution, namely, the component containing the source.

Let $\mathcal{S}^\tau(t)$ be the component that t is a member of at time τ, and let $\phi(t)$ be the time in the algorithm when some component containing t first becomes satisfied. Our cost-sharing scheme can then be expressed as follows

$$\xi(t, \mathcal{T}) = \int_{\tau=0}^{\phi(t)} \frac{1}{|\mathcal{S}^\tau(t) \cap \mathcal{T}|} d\tau \quad (8)$$

Lemma 1. *The cost-sharing scheme of* (8) *is cross-monotonic.*

Proof: Consider the dual growing process in Phase 1 of the modified primal-dual algorithm. Adding receivers can only lead to more receivers being in the same component, which leads to less cost per receiver, since from (8), the cost of every component is shared equally. Further, adding receivers can only cause other receivers to be satisfied earlier. Crucially, during Phase 1, a component's dual continues to grow even after the component is connected, for time at least as long as the shortest path cost from every receiver in the component to the source. This continuous growth mimics the behaviour of dual growth when any arbitrary subset of receivers is present in the multicast set. Hence, adding a receiver can never cause another receiver's cost to increase. This *smooth* growth of cost-shares leads to cross-monotonic cost-shares in Phase 1. Due to the choice of nodes pruned in Phase 2, the subset of receivers whose cost increases due to the dual growing process in Phase 3 can only be paying to open other receiver nodes. Hence, removing any receiver whose cost does not increase during this phase cannot decrease the extra cost accrued by the other receivers, thus preserving cross-monotonicity. □

Next, we bound the cost-sharing scheme of (8) against the feasible dual vector y of the unmodified version of Algorithm 1.

Lemma 2. $\sum_{t \in \mathcal{S}} \xi(t, \mathcal{T}) \geq \frac{1}{2} y(\mathcal{S})$ *for every component* $\mathcal{S}$.

Proof: Without loss of generality, assume that component $\mathcal{S}$ has a single receiver, since the cost of a component is shared equally between receivers in a component. Let τ_1 and τ_2 be the time when $\mathcal{S}$ first becomes satisfied under the unmodified and modified versions of Algorithm 1 respectively. Clearly $\tau_1 \leq \tau_2$, since components can only get satisfied earlier due to the ghost contribution. If $\tau_1 = \tau_2$, this means that $\mathcal{S}$ did not get satisfied due to ghost contribution, and the lemma holds trivially. Now let $\tau_1 + \delta = \tau_2$ for some $\delta > 0$. Since the ghost component and $\mathcal{S}$ are growing at uniform rate, $y(\mathcal{S}) \geq \delta$ at time τ_1. The cost share of t is therefore at least δ, while the total cost to connect t in the unmodified algorithm is at most 2δ, thus proving the lemma. □

Theorem 2 together with Lemma 2 immediately implies the following theorem.

Theorem 3. *The modified primal-dual algorithm computes cost-shares that are cross-monotonic and* $\frac{1}{4}$*-budget-balanced for the optimal Steiner tree in uniform cost wireless networks.*

6 CONCLUSION

Ensuring a mechanism is group strategyproof invariably entails the design of cost sharing schemes that are cross-monotonic. In this paper, we showed that cross-monotonic cost sharing schemes that balance the budget do not exist for multicast in wireless networks, and derived upper bounds on the cost recovery ratio that are asymptotically constant. On the positive side, we designed a primal-dual based algorithm that guarantees a constant budget-balance ratio when transmission costs are uniform. An important question is whether the gap between the upper and lowed bounds on cost recovery derived here can be decreased. Another direction is to consider the case when receivers dynamically leave and join the multicast session. It is interesting to see if the primal-dual approach can be adapted for this more realistic scenario. We intend to pursue these directions of research in our future work.

REFERENCES

[1] H. Moulin and S. Shenker, "Strategyproof sharing of submodular costs: budget balance versus efficiency," *Economic Theory*, vol. 18, pp. 511–533, 2001.
[2] H. Moulin, "Incremental cost sharing: Characterization by coalition strategy-proofness," *Social Choice and Welfare*, vol. 16, pp. 279–320, 1999.
[3] N. Nisan, T. Roughgarden, E. Tardos, and V. V. (Eds.), *Algorithmic Game Theory*. Cambridge University Press, 2007.
[4] N. Immorlica, M. Mahdian, and V. S. Mirrokni, "Limitations of cross-monotonic cost sharing schemes," in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.
[5] K. Jain and V. Vazirani, "Applications of approximation algorithms to cooperative games," in *Proc. ACM Symposium on Theory of Computing (STOC)*, 2001.
[6] M. Pal and E. Tardos, "Group strategy proof mechanisms via primal-dual algorithms," in *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.
[7] J. Konemann, S. Leonardi, and G. Schafer, "A group-strategyproof mechanism for Steiner forests," in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.
[8] Z. Li, "Cross-monotonic multicast," in *Proc. IEEE INFOCOM*, 2008.
[9] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1992.
[10] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag, 2001.
[11] L. Shapley, "A value for N-person games," *Contributions to the Theory of Games*, pp. 31–40, 1953.
[12] A. Gupta, A. Srinivasan, and E. Tardos, "Cost-sharing mechanisms for network design," in *Proc. Int'l Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, 2004.
[13] A. Mehta, T. Roughgarden, and M. Sundararajan, "Beyond moulin mechanisms," *Games and Economic Behavior*, vol. 67, no. 1, pp. 125–155, 2009.
[14] J. Brenner and G. Schafer, "Cooperative cost sharing via incremental mechanisms," in *Preprint 650, DFG Research Center Matheon, Germany*, 2009.
[15] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Info. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
[16] A. Gopinathan and Z. Li, "On achieving groupstrategyproof information dissemination in wireless networks," in *Proc. ICST GameNets*, 2009.
[17] M. Thimm, "On the approximability of the Steiner tree problem," in *Mathematical Foundations of Computer Science 2001, Springer LNCS 2136, 678-689*, 2001.
[18] C. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Dover Publications, 1998.