On Achieving Group Strategyproof Information Dissemination in Wireless Networks

Ajay Gopinathan and Zongpeng Li
Department of Computer Science
University of Calgary
{ajay.gopinathan,zongpeng}@ucalgary.ca

Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
bli@eecg.toronto.edu

***Abstract*—We study the dissemination of common information from a source to multiple peers within a multihop wireless network, where peers are equipped with uniform omni-directional antennas and have a fixed cost per packet transmission. While many peers 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 peers. 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 peer 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 $\frac{1}{8}$-approximate cost recovery against an optimal routing scheme.**

I. Introduction

Advances in wireless technology have led to its widespread adoption, making it the preferred choice for connectivity in a wide variety of settings. In a wireless *ad hoc* network, autonomous wireless nodes with the same resources and responsibilities, cooperate to route data in the absence of a fixed network infrastructure. We consider a wireless ad hoc network, formed by self-organizing peers, each equipped with an omni-directional antenna with uniform communication range, and a fixed transmission cost per packet. Peers are interested in obtaining information in the form of a data service, from a service provider. A major challenge for the service provider is to compute an efficient *routing solution* that disseminates information to peers while incurring the least transmission costs. Simultaneously, the service provider seeks to exact *payment* from the receiving peers to recover the cost of the routing solution. Designing *cost-sharing* schemes that adhere to well defined notions of fairness and economic feasibility is a central problem in game theory [1]–[3].

Computing appropriate cost-shares becomes especially challenging when peers are self-serving agents, whose utility obtained for receiving the service is known only to the peer itself. These self-interested peers may misreport its 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 peers have no incentive to lie about its utility. Such a mechanism is said to be *strategyproof*. A strategyproof mechanism that is in addition robust against collusion by peers 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 peer does not increase when the service set containing the peer 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] [6], facility location [7] [8], Steiner forests [9] and multicast in wired networks [10].

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 peers 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 peers, who may instead choose to obtain the service from another provider charging less. The budget-balance requirement is natural — the service provider wishes to recoup the cost incurred from 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 *optimal* or least 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 transmission by peer nodes, 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.

We show that the budget-balance ratio for *any* cross-monotonic cost-sharing scheme cannot be significantly higher than $\frac{1}{2}$ in wireless networks. In the case of uniform transmission costs, we further show that the budget-balance ratio is not significantly higher than $\frac{2}{3}$. Our result hinges on a pathological network construction, and we employ a probabilistic argument similar to that of Immorlica *et al.* [4] and Li [10] to derive the upper bound on cost recovery. We complement this upper bound by showing that constant factor budget-balanced schemes are possible. We design an algorithm that computes a 4-approximate routing solution, and show how we can modify this algorithm to guarantee a cost recovery ratio of at least $\frac{1}{8}$ of the total cost of multicast in a wireless ad hoc network. Our technique is based on the primal-dual schema [7], [9], [11], [12], and ensures cross-monotonicity by continuously increasing dual variables, which occasionally 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 II, we discuss related work. We introduce our network model as well as some game theoretic definitions in Section III. In Section IV, 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 V, and prove its performance bound. We discuss our results and conclude in Section VI.

II. 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 easy to achieve cross-monotonic cost sharing, *e.g.*, through the Shapley value method [13]. 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 service provider. 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 cost-sharing algorithms. The minimum spanning tree [6], the travelling salesman problem [6], the facility location game [7], [8], single-source rent-or-buy [7], [14], and Steiner forest [9] all constitute combinatorial optimization games for which algorithms have been developed for computing cross-monotonic cost-shares.

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*, cross-monotonic cost-sharing schemes lead to group strategyproof mechanisms.

In the context of multicast, cross monotonic cost-sharing was also studied by Feigenbaum *et al.* [15]. Assuming a fixed multicast tree, they show that proportional cost-sharing reduces to the Shapley value, which is a submodular function, hence giving rise to group strategyproofness. Multicast was also studied by Penna and Ventre [16]. Using a minimum spanning tree construction on all nodes, they modified the Moulin mechanism to query only leaf nodes in deciding the service set. Hence, only leaf nodes are required to pay for service.

In contrast to the previously mentioned work, Li studied cross-monotonic cost-sharing for the *optimal* multicast flow [10]. The optimal flow was computed under assumption that network coding [17], [18] was used for multicast. Li studied directed and undirected networks, and provided upper bounds on cost recovery, as well as algorithms to achieve good budget-balance for both types of network. 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 the present work, we consider wireless networks, which can be modeled as directed networks using a universal transformation [19], but show that constant budget-balance is still possible.

Cost sharing schemes for multicast in wireless networks has previously been studied by Penna and Ventre [20], as well as Bilo *et al.* [21]. Both study optimal multicast in terms of minimum energy usage. Penna and Ventre provide

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

strategyproof cost sharing schemes for multicast in wireless networks, based on the celebrated Vickrey-Clarke-Groves mechanism. Unlike the present work, Penna and Ventre do not provide a method for computing cross-monotonic cost shares. Further, they assume that the communication tree is pre-computed. Our algorithm on the other hand, simultaneously computes a 4-approximate routing scheme while computing cost shares that give rise to group strategyproofness. Bilo *et al.* employ Jain and Vazirani's [6] Steiner tree approximation method to compute a cross-monotonic cost sharing scheme that is $\frac{1}{12}$-budget-balanced for wireless networks, where the cost of transmission between nodes decays exponentially with distance. In contrast, we focus on wireless networks with fixed transmission costs, and our algorithm achieves a $\frac{1}{8}$-budget-balance ratio.

Algorithms for cross-monotonic cost recovery in the literature are primarily based on the primal-dual method [6], [7], [9]. Primal dual methods are attractive in that they allow the construction of the solution by carefully controlling the growth of dual variables. Dual variables correspond to the cost of building the solution, and smoothly increasing this leads to cost-shares that grow in a cross-monotonic fashion. Primal-dual methods were first used by Goemans and Williamson to design approximation algorithms for a class of constrained forest problems [11]. In the context of network design problems, Agrawal, Klein and Ravi [22] extended this method to design approximation algorithms for the Steiner forest problem, of which the Steiner tree is a special case.

III. 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 in wireless networks is not submodular, precluding the use of the Shapley value [13] as a viable cost-sharing scheme for group strategyproofness.

A. The Network Model

We will assume that the wireless networks we study can be modeled by disk graphs with some uniform radius, r. In such graphs, a wireless peer or node[2] 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 has a cost to transmit a unit of information, and we denote a node u's transmission cost 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 will assume that there is a distinguished source node s, with identical data to be sent to a set of receivers T. For example, the source node may be providing media streaming service to 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 [23], [24], we will seek to compute an approximately optimal Steiner tree instead.

[2]we will use peers and nodes interchangeably in the sequel.

B. Group Strategyproof Mechanisms and Cost Sharing Schemes

Consider the following problem: a set U of agents are interested in obtaining a service from a service provider. Each agent $i \in U$ has some private valuation for the service, v_i, and places a bid of b_i for obtaining the service. Each agent is *selfish*, in the sense that she always tries to maximize her utility, $u_i = v_i - b_i$. In such a scenario, the service provider is faced with the following two problems; (1) deciding the set of agents $S \subseteq U$ that should receive the service, and (2) deciding the cost share of agent i in the set S, denoted $\xi(i, S)$. This scenario constitutes a *cooperative game* [1], [3], and is widely applicable in a variety of settings. Agents seeking to maximize their utility may lie about their valuation, and place lower bids. The service provider on the other hand, seeks to recover the cost of serving the agents. A Moulin-Shenker mechanism, coupled with a cross-monotonic cost-sharing scheme, ensures that truthfully reporting v_i by agent i is the dominant strategy, even when agent i is acting in collusion with other agents.

Formally, a cross-monotonic cost-sharing scheme for some agent i in the set A has the following property

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

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.

Let us denote $C_{OPT}(S)$ as the cost of the optimal (or cheapest) solution that serves S. It is further desirable that the cost-shares computed possess the following properties

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

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 an approximately budget-balanced scheme. A cost-sharing scheme is said to be β-budget-balanced for $0 \leq \beta \leq 1$ if the following holds instead

$$\beta\, C_{OPT}(S) \leq \sum_{i \in S} \xi(i, S) \leq C_{OPT}(S)$$

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

$$C_{OPT}(S) \leq \sum_{i \in S} \xi(i, S) \leq \alpha\, C_{OPT}(S)$$

Clearly, a β-budget-balanced cost-sharing scheme is equivalent to a $\frac{1}{\beta}$-competitive cost-sharing scheme.

C. Multicast in wireless networks is not submodular

A function f is said to be submodular if for all $A \subset B$ and for some $i \notin A$, the following holds

$$f(B \cup i) - f(i) \leq f(A \cup i) - f(i)$$

A cost function that is submodular intuitively means that the marginal cost of servicing a new agent decreases as the service set expands. Submodular cost functions imply that a cross-monotonic and perfectly budget-balanced cost-sharing scheme exists in the form of the Shapley value [13]. However, we show that the cost function for multicast in wireless networks is not submodular using a simple counterexample.

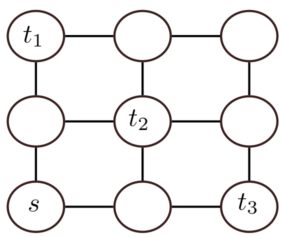

Fig. 1. Multicast in wireless networks is not submodular

Consider the example wireless network shown in Fig. 1. Let $c(u) = 1$ for all nodes u, and let $C_{OPT}(.)$ represent the cost of the optimal multicast for an arbitrary set of receivers. Then

$$\begin{aligned} C_{OPT}(t_2 \cup t_3) &= 2 \\ C_{OPT}(t_2) &= 2 \\ C_{OPT}(t_1 \cup t_2 \cup t_3) &= 3 \\ C_{OPT}(t_1 \cup t_2) &= 2 \end{aligned}$$

Since

$$C_{OPT}(t_2 \cup t_3) - C_{OPT}(t_2) < C_{OPT}(t_1 \cup t_2 \cup t_3) - C_{OPT}(t_1 \cup t_2)$$

and $\{t_2\} \subseteq \{t_1, t_2\}$, the multicast cost function is not submodular in wireless networks.

IV. Upper bounds on cross-monotonic cost recovery

In this section, we will show that there does not exist cross-monotonic cost-sharing schemes that are budget-balanced for all wireless networks. We begin by showing a simple topology that does not admit a cross-monotonic budget-balanced cost-sharing allocation. Subsequently, we generalize the ideas in this topology to show that the upper bound on cost recovery in wireless networks with a uniform radius is $O(\frac{1}{2} + \frac{1}{2\eta})$ where η is a network dependent parameter.

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

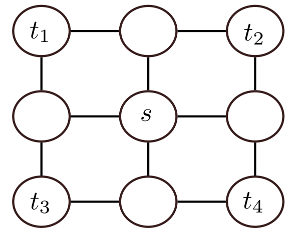

Fig. 2. $\frac{3}{4} + \epsilon$ budget-balance is not possible

Full cost recovery is impossible for any cost allocation scheme that is cross monotonic in wireless networks. We will show this with a simple but representative topology, which will provide us with an intuition into why the *broadcast advantage* restricts cross-monotonic cost recovery. In the subsequent section, we will generalize this idea to general wireless networks with uniform coverage radius.

Consider the network shown in Fig. 2. All nodes have cost $c(u) = 1$ to transmit a unit of information, except for the source, which has zero cost. Firstly, choose at random any three from four potential receivers in $\{t_1, t_2, t_3, t_4\}$ to be in the multicast group. Consider a subset consisting of any two adjacent receivers in the multicast group. Since we choose randomly, each node will pay at most $1/2$ in expectation when only this subset is in the multicast group. By cross monotonicity, the third node will also only pay at most $1/2$. The minimum cost for multicast to any three receivers is 2, and hence, the cost recovery ratio is *at most*

$$\frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{2}}{2} = \frac{3}{4}$$

B. 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 disk graphs with uniform radius, which does not admit a cost-sharing scheme for optimal multicast that is cross-monotonic and* $(\frac{1}{2} + \frac{1}{2\eta} + \epsilon)$*-budget-balanced for any* $\epsilon > 0$*, where* η *is a network dependent parameter such that* $\eta \geq 3$*.*

Proof: Let $N = 2\eta$, and let the source s have $c(s) = x$. We arrange N relay nodes u with $c(u) = y$ at equidistance from each other along the circumference of s's neighbourhood. We next label these relay nodes by numbering them clockwise in ascending order. We next place N potential multicast receivers t_i with $c(t_i) = y$ by reflecting the source node along the axis spanned by node numbered i and $(i + \eta - 1 \mod N)$, for all i. Observe that by construction, each potential multicast receiver has η relays in its neighbourhood. Moreover, each relay node has η potential multicast receivers within its communication range. Additionally, note that N can be accommodated for any arbitrarily large η by scaling r appropriately.

We next prove the poor budget-balance ratio of a network constructed as above. Pick a potential multicast receiver t_i at *random* and include the next η nodes in the clockwise direction, and label this as the target multicast group, T. Observe that the set consisting of the first η nodes in the multicast group (call this A) have one common relay node (call this r_A), while the last η nodes (call this B) share a separate common relay node (call this r_B). Let $\xi(t_i, S)$ be the cost share of node t_i in some subset $S \subseteq T$. Since we pick the multicast group at random, we bound the expected cost share of each node as follows:

$$\begin{aligned} E[\sum_{t_i \in T} \xi(t_i, T)] &= E[\sum_{t_i \in A} \xi(t_i, T)] + E[\sum_{t_i \in T/A} \xi(t_i, T)] \\ &\leq E[\sum_{t_i \in A} \xi(t_i, A)] + E[\sum_{t_i \in T/A} \xi(t_i, B)] \\ &= \eta \frac{x+y}{\eta} + \frac{x+y}{\eta} \\ &= (x+y)(1+\frac{1}{\eta}) \end{aligned}$$

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

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

Setting $x = 0$ and $y = 1$ yields the desired result.

Theorem 1 also implies the following corollary:

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

■

V. Cross-monotonic cost-sharing schemes

A. A budget-balanced scheme for line graphs

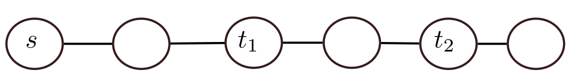

Fig. 3. Cross-monotonic cost-shares are easily found for networks that can be modeled as line graphs

Consider a simple line graph, as shown in Fig. 3. One can easily design a cross-monotonic cost-sharing scheme for such a network. One possibility would be to charge the receiver furthest away from the source the entire cost of the transmission. Such a cost-sharing scheme is cross-monotonic, since adding a receiver to any subset never causes a receiver's cost to increase. Further, it is perfectly budget-balanced. Yet another possibility is a cascading charging scheme. For every receiver t, let $R(t, T)$ be the set of receivers in T closer to s than t. Then, the following cost-sharing scheme

$$\xi(t, T) = d(s, t) - \sum_{t' \in R(t)} d(s, t')$$

which essentially charges each receiver t the added cost of the solution for serving t, is also cross-monotonic and perfectly budget-balanced.

B. A $\frac{1}{8}$-budget-balanced scheme for uniform cost networks

In this section, we will design an algorithm for computing cost-shares that are cross-monotonic for multicast transmission in a wireless ad hoc network. We will build a Steiner tree for efficient dissemination of information in a multicast. However, Steiner trees are NP-Hard to compute [23], [24]. Accordingly, we will design an algorithm that builds a solution that is within a constant factor of the optimal solution. At the same time, the cost-shares we compute will recover a constant fraction of the cost of the solution we construct. We focus on the case when each node has uniform cost to transmit. Without loss of generality, let $c(u) = c(v) = 1, \forall u, v$.

We begin with some required definitions. For a set of nodes S, we use the proper binary function $f(S)$ in the following sense

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

Hence, $f(S) = 1$ if S contains at least one multicast receiver and does not contain the source node s. We define a *node cut* of the set S as the set of nodes $\delta(S)$, where the following conditions hold

- $V - \delta(S)$ induces a graph where $\bar{S}$ is disconnected from S
- If $u \in \delta(S)$, then there exists a $w \in S$ such that u and w are adjacent.

A node cut is then the set of nodes adjacent to some other set S, which if removed disconnects S from the source.

The optimal Steiner tree in a wireless network with the set of nodes V can be computed by solving an integer program (IP) of the following form:

$$\text{Minimize} \quad \sum_{u \in V} c(u)x(u) \tag{2}$$

Subject To

$$\sum_{u \in \delta(S)} x(u) \geq f(S) \quad \forall S \subseteq V \tag{2a}$$

$$x(u) \in \{0, 1\} \tag{2b}$$

In IP (2), the binary variable $x(u)$ indicates if node u is used to transmit information in the optimal Steiner network. The objective function tries to minimize the cost of the transmitting nodes, while ensuring that each receiver $t \in T$ is connected to the source s via a path of transmitting nodes.

Since the optimal Steiner tree problem is intractable, we will employ the primal-dual schema [11] to solve for an approximately optimal Steiner tree. The primal-dual schema will employ the linear program (LP) relaxation of IP (2), achieved by first relaxing the integrality requirement to $x(u) \geq 0$ for all u, and forming the subsequent dual LP:

$$\text{Maximize} \qquad \sum_{S \subseteq V} f(S)y(S) \tag{3}$$

Subject To

$$\sum_{y(S):u \in \delta(S)} y(S) \leq c(u) \qquad \forall u \in V \tag{3a}$$

$$y(S) \geq 0 \tag{3b}$$

The dual variables $y(S)$ can be seen as cost-shares for the set of nodes S. Our algorithm works in the following way; we begin with an infeasible primal IP by setting $x(u) = 0$ for all u. In line with most primal-dual approximation methods, we will build a primal solution that is feasible by increasing the dual variables judiciously. By complementary slackness, $x(u) = 1$ implies that the corresponding constraint of (3a) must hold with equality. Using this intuition, we will set $x(u) = 1$ whenever the corresponding constraint of (3a) holds with equality due to the dual vector y.

We will design two algorithms based on the ideas just described. The first algorithm computes cross-monotonic cost-shares for Steiner trees in wireless networks, and we will refer to it as algorithm CSTW. Algorithm CSTW ensures cross-monotonicity by increasing dual variables in a smooth fashion, even if it results in violating dual constraints. The result is an infeasible dual, which makes the task of bounding the approximation factor of the primal solution difficult. Therefore, we will at the same time design another algorithm, STW, that will be used to create the Steiner tree. We then bound the cost-shares computed by CSTW against the primal solution constructed by algorithm STW to obtain our budget-balance ratio.

We begin by describing algorithm STW. To describe STW, we require some terminology. We will say a constraint is *tight* if (3a) holds with equality for some node u. A node is said to be *paid for* or *opened* if $x(u) = 1$. We will use the term *component* to refer to a set of nodes. In the beginning of the algorithm, we will have $|T|$ components, each consisting of a receiver $t \in T$. We will associate a notion of time, τ, with our algorithm. A component S is said to be *satisfied* if $f(S) = 0$, i.e., S includes the source node s. Let $d(s,t)$ be the shortest path cost from the source node to receiver t. Let S^τ be the set of nodes in component S at time τ. We will say a component S is *active* at time τ if S has not been satisfied at time τ.

Algorithm STW proceeds in iterations. In each iteration, we increase or *grow* dual variables of every *active* component uniformly in time. An iteration ends when one or more constraints of type (3a) for some set of nodes R become tight. At this point, we stop dual growth temporarily, and we open each node $u \in R$ by setting $x(u) = 1$. For each open node u that is adjacent to some component S, we will add u to the membership of S. If two or more components have contributed to the same node in R, or to adjacent nodes in R, we merge these components into a new single component. Similarly, when two or more components are satisfied, we merge these components into a single larger component as well. The next iteration of the algorithm then begins. This dual growing phase ends there are no longer any active components.

One can view the dual growing phase 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 receivers themselves. Therefore, at the end of the algorithm, to ensure connectivity, we set $x(t) = 1$ for all $t \in T$, assuming this has not already occurred. This is required to ensure all receivers are connected to the source. Let us make this more concrete with an example. Assume that components S_1 and S_2, with receivers t_1 and t_2, merge into S' by sharing the cost of opening some node m. Assume without loss of generality that at a later time S' is satisfied, with a path from s to t_2. Now t_1's path to s requires t_2 to be transmitting. Hence, opening the receivers at the end of the dual growing phase is necessary to ensure primal feasibility. However, this has consequences for the approximation factor of the algorithm, which we discuss later.

A by-product of the primal-dual schema we have just described is that at the end of the dual growing phase, we will have a number of superfluously open nodes. To ensure a good primal solution, we will examine these open nodes and close them, as long as this can be done without disconnecting some receiver from the source. That is, we set $x(u) = 0$ for some open node u if after doing so, every $t \in T$ still has a path of open nodes to the source. When processing nodes for pruning, we examine them in the *reverse order in which they were opened* during the course of the algorithm. When there are ties, we break them in an arbitrary but fixed manner (for example, by using node ids). The algorithm terminates at the end of this pruning phase. We denote the set of open nodes in the primal solution at the end of the algorithm W. It is easy to see that due to the pruning phase, W will be a tree connecting s to all $t \in T$, such that each $t \in T$ will have exactly one path to the source.

Let us now analyze the performance of algorithm STW. First, note that at the end of the algorithm, we will have a primal feasible solution. This must be so, since the dual growing phase only ends when all components are connected to the source. Further, opening receivers ensures that this connection requirement is met, and the pruning stage ensures it is never violated. Let $W' = W - (W \cap T)$, i.e., W' is the primal solution without the set of open receivers. The next lemma shows that excluding receivers, the dual y computed by algorithm STW pays for at least half of the cost of the solution constructed, excluding the cost of receivers.

Lemma 1. $\sum_{u \in W'} 1 \leq \sum_{S \subseteq V} 2y(S)$

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 is a tree, and 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 S never pays for nodes in the component, only to nodes that are node cuts adjacent to 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 W'. The resulting graph (call it 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 S, as the nodes in H adjacent to S, and let us denote it $\deg(S)$. 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 is a tree. Therefore, every node in 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 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 W' are open, its corresponding constraint must be tight, and so we get

$$\begin{aligned}\sum_{u \in W'} 1 &= \sum_{u \in W'} \Big(\sum_{S: u \in \delta(S)} y(S) \Big) \\ &= \sum_{S \subseteq V} \Big(\sum_{u: u \in W' \cap \delta(S)} y(S) \Big) \\ &= \sum_{S \subseteq V} \Big(\deg(S) y(S) \Big)\end{aligned}$$

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

$$\sum_{u \in W'} 1 \leq \sum_{S \subseteq V} 2y(S)$$

■

Recall that algorithm STW open receivers "for free" at the end of the dual growing phase, that is, these receivers were not paid for by the dual computed. The next lemma shows that this weakens the approximation ratio of our algorithm by a factor of at most 2.

Lemma 2. $\sum_{u \in W} 1 \leq \sum_{S \subseteq V} 4y(S)$

Proof: Let $k = |T|$. If a receiver $t \in W$ is opened after the dual growing phase, then it must be that t is required to connect some other receiver t' to the source, and at some point in the algorithm, say time τ_m, the components containing t and t' were merged. In the worst case, we have to open $k-1$ receivers in the primal without having paid for it in the dual, with receiver t_1 required to connect t_2, which in turn is required to connect t_3, and so on. Let each of the receivers t_i be in some component S_i when they merged. Let W'' be the nodes in W' that have been paid for at the time of the merge. The cost of the solution at time τ_m is then

$$(\sum_{u \in W''} 1) + k - 1$$

But we know from lemma 1 that

$$\sum_{u \in W''} 1 \leq 2 \sum_{i=1}^{k} y(S_i)$$

Let α be the new bound we seek. Then

$$2 \sum_{i=1}^{k} y(S_i) + k - 1 \leq \alpha \sum_{i=1}^{k} y(S_i)$$

Due to uniform growth rate and cost, $y(S_i) \geq \frac{1}{2}$ And so at worst, we get

$$\alpha(\frac{k}{2}) \geq 2(\frac{k}{2}) + k - 1$$

and so $\alpha \geq 4 - \frac{2}{k}$. Setting $\alpha = 4$ yields the bound stated in the lemma. ■

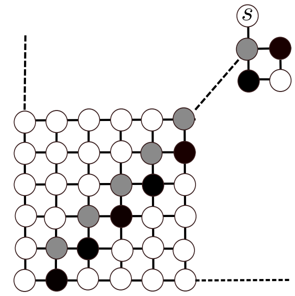

Fig. 4. An almost tight example for Lemma 2

It turns out that Lemma 2 is essentially almost tight. Consider the grid network as shown in Fig. 4. Every node's immediate neighbours are its node cuts. In Fig. 4, the black nodes are receivers, who wish to connect to the node marked s. In algorithm STW, in the beginning, each node is its own component. At time $\tau = 0.5$, all grey nodes have been paid for (since each node costs 1), and are declared open. This causes all components to merge into a single component. This component will be satisfied at time $\tau = 1.5$, when the algorithm ends. Assume that there are k receivers. Then the total dual is $k(\frac{1}{2}) + 1$, while the actual cost of the solution is $2k - 1$. Hence, the ratio of dual to the solution constructed is

$$\frac{k+2}{4k-2}$$

which goes to $\frac{1}{4}$ for very large k.

Lemma 2 also immediately implies the following corollary

Corollary 2. *Algorithm STW is a 4-approximation algorithm to the Steiner tree problem in wireless networks with uniform cost.*

Proof: By LP duality, the optimal primal solution is lower bounded by the dual solution. Therefore, lemma 2 immediately implies that algorithm STW constructs a 4-approximate solution. ■

While algorithm STW creates a viable primal solution, we cannot use its dual as cost-shares. Ceasing to increase the dual variable of a satisfied component means the dual does not grow smoothly, and may not preserve cross-monotonicity. Instead, we will use algorithm CSTW to compute cross-monotonic cost-shares. Algorithm CSTW mirrors STW in every way, with the following exception; a component S is said to be active at time τ as long as the following condition holds

$$\max_{t\in S^{\tau}\cap T} d(s,t) \geq \tau \qquad (4)$$

Let the primal solution built by CSTW be the set of nodes Z. Note that we depart from the usual primal-dual schema, by increasing dual variables for satisfied components, as long as the condition in (4) holds. We will call a component's contribution after it becomes satisfied as its *ghost contribution* [7]. Note that at all times in the algorithm, only one component is contributing ghost shares, namely, the component containing the source.

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

$$\xi(t,T) = \int_{\tau=0}^{\phi(t)} \frac{1}{S^{\tau}(t)} d\tau \qquad (5)$$

Lemma 3. *The cost-sharing scheme described in* (5) *is cross-monotonic.*

Proof: Adding receivers can only lead to more receivers being in the same component, which leads to less cost per receiver, since the cost of that component is shared. Crucially, 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. ■

Next, we bound the cost-sharing scheme of (5) against the feasible dual vector y of algorithm STW.

Lemma 4. *For every component S, we have $\sum_{t\in S}\xi(t,T) \geq \frac{1}{2}y(S)$.*

Proof: Without loss of generality, assume that component 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 S becomes satisfied under algorithms CSTW and STW 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 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 S are growing at uniform rate, $y(S) \geq \delta$ at time τ_1. The cost share of t is therefore at least δ, while the total cost to connect t in algorithm STW is at most 2δ, thus proving the lemma. ■

Lemmas 2, 3 and 4 then allow us to state the following theorem.

Theorem 2. *Algorithm CSTW computes cost-shares that are cross-monotonic and $\frac{1}{8}$-budget-balanced for building a Steiner tree in wireless networks with uniform cost.*

VI. 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. We further showed that the upper bound on cost recovery is not significantly higher than $\frac{1}{2}$ and $\frac{2}{3}$ respectively for wireless networks with heterogeneous and uniform transmission costs. On the positive side, we designed an 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. In one direction, we seek a better network construction that results in worse upper bounds on the budget-balance ratio. In the other, it may be possible that the primal-dual algorithm we presented can be improved for better cost recovery. At the same time, it is interesting to see if our algorithm can be further modified to compute cross-monotonic cost shares for wireless networks with heterogeneous transmission costs. 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 *Proceedings of the 16th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

[5] K. Kent and D. Skorin-Kapov, "Population monotonic cost allocation on MSTs," in *Operational Research Proceedings KOI*, 1996.

[6] K. Jain and V. Vazirani, "Applications of approximation algorithms to cooperative games," in *Proceedings of the 33rd annual ACM Symposium on Theory of Computing (STOC)*, 2001.

[7] M. Pal and E. Tardos, "Group strategy proof mechanisms via primal-dual algorithms," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

[8] S. Leonardi and G. Schaefer, "Cross-monotonic cost-sharing methods for connected facility location games," in *Proceedings of the 5th ACM conference on Electronic Commerce (EC)*, 2004.

[9] J. Konemann, S. Leonardi, and G. Schafer, "A group-strategyproof mechanism for Steiner forests," in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

[10] Z. Li, "Cross-monotonic multicast," in *Proceedings of IEEE INFOCOM*, 2008.

[11] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," in *Proceedings of the 3rd annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1992.

[12] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag, 2001.

[13] L. Shapley, "A value for N-person games," *Contributions to the theory of games*, pp. 31–40, 1953.

[14] A. Gupta, A. Srinivasan, and E. Tardos, "Cost-sharing mechanisms for network design," in *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, 2004.

[15] J. Feigenbaum, C. Papadimitriou, and S. Shenker, "Sharing the cost of multicast transmissions," *Journal of Computer and System Sciences*, vol. 63, pp. 21–41, 2001.

[16] P. Penna and C. Ventre, "Free-riders in Steiner tree cost-sharing games," in *Proceedings of 12th International Colloquium on Structural Information And Communication Complexity*, 2005.

[17] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[18] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 782–795, 2003.

[19] Z. Li, B. Li, and M. Wang, "Optimization models for streaming in multihop wireless networks," in *Proceedings of IEEE ICCCN*, 2007.

[20] P. Penna and C. Ventre, "Sharing the cost of multicast transmissions in wireless networks," in *Proceedings of the 11th International Colloquium of Structural Information and Communication Complexity (SIROCCO)*, 2004.

[21] V. Bilo, M. Flammini, G. Melideo, L. Moscardelli, and A. Navarra, "Sharing the cost of multicast transmissions in wireless networks," *Theoretical Computer Science*, vol. 369, no. 1-3, pp. 269 – 284, 2006.

[22] A. Agrawal, P. Klein, and R. Ravi, "When trees collide: An approximation algorithm for thegeneralized steiner problem on networks," *SIAM Journal on Computing*, vol. 24, pp. 440–456, 1995.

[23] M. Thimm, "On the approximability of the Steiner tree problem," in *Mathematical Foundations of Computer Science 2001, Springer LNCS 2136, 678-689*, 2001.

[24] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing Steiner trees," in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.