

Share the Multicast Payment Fairly

WeiZhao Wang¹, Xiang-Yang Li^{*1}, and Zheng Sun^{**2}

¹ Illinois Institute of Technology, Chicago, IL, USA, wangwei4@iit.edu, xli@cs.iit.edu

² Hong Kong Baptist University, Hong Kong, China, sunz@comp.hkbu.edu.hk

Abstract. Multicast routing uses a structure, either a tree or a mesh, to connect the receivers to the source, thus saving the bandwidth. How to share the cost among the receivers in a certain fair way has been studied widely in literature. When the agents, either the links or the nodes, in the network are selfish, it is desirable to pay the agents in a proper way such that each agent still reveals its true cost, a property known as strategyproofness. Several strategyproof mechanisms have been proposed based on several different multicast structures, and it is natural that the payments instead of the costs should be shared among the receivers. Motivated by this, we study how the payment should be shared among the receivers in a fair way when the payment is computed via a strategyproof mechanism based on some low cost multicast structure. Specifically, when links are selfish agents, based on a strategyproof mechanism whose multicast tree is at most 2 times the optimal, we propose a payment sharing scheme that is $\frac{1}{n^2}$ -budget-balanced, cross-monotonic, and in the core. We also prove that there is no payment sharing scheme that can achieve β -budget-balance and cross-monotonicity for $\beta = \Omega(\frac{1}{n})$. When both the relay agents and the receivers are selfish, we show a negative result: combining a strategyproof mechanism M for the relay agents and a fair sharing scheme ξ^{LST} for the receivers does not necessarily imply a strategyproof mechanism overall.

1 Introduction

Multicast has been a popular technique for supporting group-based applications, such as video-conference and content distribution. Multicast routing often uses a tree to connect the receivers to the source, and every internal node only sends the data to its downstream nodes in the multicast tree, which saves the bandwidth. Traditionally, whenever the source needs to send some data to a subset of the receivers, the multicast routing picks the shortest path tree (also called least cost path tree in [1]) that spans these receivers. This simple approach ignores the fact that the shortest path tree may be arbitrarily worse than the optimal tree with respect to the total cost. Thus, we need to find the optimal tree connecting a given set of receivers with the minimum total cost, a problem known as the *Steiner tree* problem. However, this problem is well-known to be NP-hard, and a sequence of approximation algorithms have been developed for the Steiner tree problems. In literature, there are two types of Steiner trees: the node weighted Steiner tree where only the nodes have costs, and the link weighted Steiner tree where only the links have the costs. In this paper, we will assume that the network is link weighted, *i.e.*, only the links have costs. Notice that most of our results can be extended to the node weighted network without difficulty.

Recently, sharing the cost of the multicast routing among receivers in a “fair” manner has been studied extensively [2, 3]. An assumption made by the cost sharing schemes is that the costs of the links (or nodes) are publicly known. However, this is not the case in many application scenarios. When the links (or nodes) are independent and self-interested agents, several strategyproof mechanisms [1, 4, 5] have been developed so that a proper payment to each agent can be computed efficiently. A multicast mechanism $M = (\mathcal{O}, \mathcal{P})$ consists of two parts: a multicast tree construction method \mathcal{O} that determines which agent will be used in the tree, and a payment scheme \mathcal{P} that determines how much each agent will be paid to compensate its cost incurred. When the payments to the agents needed to be shared among the receivers, we need to design a *payment sharing scheme* instead of the traditional cost sharing scheme. If every receiver is also selfish with a privately known valuation, then a *payment sharing mechanism* is needed to determine which receiver get the multicast data and at what price. Surprisingly enough, several results from cost sharing mechanisms do not carry over to the payment sharing mechanism. In [1], Wang *et al.* first studied how to fairly share, among the set of receivers, the payment of the mechanism that uses the shortest path tree as its multicast tree.

* The research of the author was supported in part by NSF under Grant CCR-0311174.

** The research of the author was supported in part by Grant FRG/03-04/II-21 and Grant RGC HKBU2107/04E.

By assuming that each receiver is willing to pay the computed charge, *i.e.*, its valuation is sufficiently large, they proved that their payment sharing scheme is fair. However, the cost of the shortest path tree could be as large as r times of the optimum, where r is the number of receivers. In this paper, we will study the payment sharing scheme when the payment is computed by a certain mechanism using a multicast tree with a constant approximation ratio, when the network links are selfish agents with privately known costs. We first show that if a payment sharing scheme is cross-monotone and never overcharges, then the total charge to receivers is at most $\Theta(\frac{1}{n})$ of the total payment to the selfish relay links in the worst case. We then present a payment sharing scheme that is in the core and can recover at least $\Theta(\frac{1}{n^2})$ of the total payment to the selfish links. When both the relay agents and the receivers are selfish, we show a negative result: combining a strategyproof mechanism M for the relay agents and a fair sharing scheme ξ^{LST} for the receivers do not necessarily imply a strategyproof mechanism overall.

The rest of the paper is organized as follows. In Section 2, we give necessary preliminaries and review the related previous results. We present our payment sharing scheme in Section 3. In Section 4, we present some negative results on payment-sharing mechanisms when both the relay links and the receivers are selfish. We conclude our paper in Section 5 with possible future works.

2 Preliminaries and Previous Works

2.1 Algorithmic Mechanism Design

In a standard model of algorithm mechanism design, there are n agents $\{1, 2, \dots, n\}$. Each agent $i \in \{1, \dots, n\}$ has some *private* information t_i , called its *type* (*e.g.* its cost to forward a packet in a network environment). All agents' types define a *profile* $t = (t_1, t_2, \dots, t_n)$. Each agent i declares a valid type τ_i^j which may be different from its actual type t_i and all agents' strategies define a declared type vector $\tau = (\tau_1, \dots, \tau_n)$. A mechanism $M = (\mathcal{O}, \mathcal{P})$ is composed of two parts: an output method \mathcal{O} (also called *allocation rule* in some literature) that maps a declared type vector τ to an output o , and a *payment* scheme \mathcal{P} that decides the monetary payment $p_i = \mathcal{P}_i(\tau)$ for every agent i . Each agent i has a valuation function $w_i(t_i, o)$ that expressed its preference over different outcomes. Agent i 's *utility* or called *profit* is $u_i(t_i, o) = w_i(t_i, o) + p_i$, given output o and payment p_i . An agent i is said to be *rational* if it always chooses its strategy τ_i to maximize its utility u_i .

Let $\tau_{-i} = (\tau_1, \dots, \tau_{i-1}, \tau_{i+1}, \dots, \tau_n)$, *i.e.*, the declared types of all other agents except i and $\tau^i t_i = (\tau_1, \tau_2, \dots, \tau_{i-1}, t_i, \tau_{i+1}, \dots, \tau_n)$. A mechanism is *strategyproof* if for every agent i , revealing its true type t_i will maximize its utility *regardless* of what other agents do. In this paper, we are only interested in mechanisms $M = (\mathcal{O}, \mathcal{P})$ that satisfy the following three conditions:

1. **Incentive Compatibility (IC)**: \forall agent $i, \forall \tau, w_i(t_i, \mathcal{O}(\tau^i t_i)) + p_i(\tau^i t_i) \geq w_i(t_i, \mathcal{O}(\tau)) + p_i(\tau)$
2. **Individual Rationality (IR)** (a.k.a., Voluntary Participation): Each agent must have a non-negative utility, *i.e.*, $w_i(t_i, \mathcal{O}(\tau^i t_i)) + p_i(\tau^i t_i) \geq 0$.
3. **Polynomial Time Computability (PC)**: \mathcal{O} and \mathcal{P} are computed in polynomial time.

In addition to the above requirements, some other common requirements for a mechanism could be

1. **Consumer Sovereignty (CS)**: If the actions of all other agents are fixed, for every player i , there exists a threshold $\tilde{\tau}_i$ such that player i is guaranteed to get the service when its cost is at most $\tilde{\tau}_i$.
2. **Group Strategyproof (GS)**: No group of players can increase their utilities (or called profits) by colluding and lying about their truthful types.

2.2 Payment Sharing

In multicast transmission, one of the key concerns is how to charge the receivers in a *fair* way. If the relay links are cooperative, *i.e.*, the costs of relay links are publicly known, then we need to share the costs of the multicast tree among receivers fairly. For the fair cost sharing, most of the literatures [2, 6, 7] used the *Equal Link Split Downstream* (ELSD) sharing scheme to charge receivers: the *cost* of each link is shared *equally* among all its downstream receivers. However, if we simply apply the ELSD as our charging scheme to share the payment, it usually is not reasonable in a common sense; for example, it is not cross-monotonic. If the relay links are selfish, then we have to share the payments to these relay links, while the payment to a selfish link should be computed by a certain strategyproof mechanism. In this paper, our focus is the payment sharing instead of the cost sharing studied in literature.

Consider a set U of n players. For a subset $S \subseteq U$ of players, let $\mathbb{P}(S)$ be the total *payment* of providing service to S . A payment sharing scheme is simply a function $\xi_i(S, c)$ with $\xi_i(S, c) = 0$ for $i \notin S$, for every set $S \subseteq U$ of players. While the definition of budget-balance is straightforward, defining fairness is more subtle: many fairness concepts were proposed in the literature, such as *core* and *bargaining set* [8]. We call a charging scheme ξ *reasonable* or *fair* if it satisfies the following criteria.

1. **Budget Balance** (BB): The payment to all relay agents should be shared by the receivers, *i.e.*, $\mathbb{P}(R, c) = \sum_{r_i \in R} \xi_i(R, c)$, where c is the cost vector of the network. When budget-balance cannot be met, we relax it to β -budget-balanced: for the receiver set R , $\beta \cdot \mathbb{P}(R, c) \leq \sum_{i \in R} \xi_i(R, c) \leq \mathbb{P}(R, c)$, for some given parameter $\beta \leq 1$. Equivalently, if we divide the shares by β , we would require that the total payment sharing of the receivers is at least the payment of providing the service, but does not exceed $\frac{1}{\beta}$ of that.
2. **No Positive Transfer** (NPT): Any receiver r_i 's sharing should not be negative. In other words, we do not pay the receiver to receive service.
3. **Cross-monotonicity** (CM): For any two subsets $S \subseteq T$ and $i \in S$, $\xi_i(S, c) \geq \xi_i(T, c)$. In other words, the payment share of a player i should not go up if more players require the service. This is also called *population monotone*.
4. **Fairness under core** (Core): For any subset $S \subseteq R$, $\sum_{i \in S} \xi_i(R, c) \leq \mathbb{P}(S, c)$. In other words, the payment shares paid by any subset of players should not exceed the total payment of providing the service to them alone, hence they have no incentives to secede.

Notice that a budget-balanced and cross-monotonic cost sharing scheme is always in the core. When each receiver q_i has a maximum payment ζ_i it is willing to pay to receive the multicast service, then we have to decide which receivers will get the service and at what price, *i.e.*, we need design a truthful mechanism. This mechanism sometimes is called *payment sharing mechanism*. Several additional properties could be required for a payment sharing mechanism.

1. **Group Strategyproof** For any subset of receiver $S \subseteq R$, they can not collude together such that every receiver does not decrease its utility while at least one receiver increases its utility.
2. **Consumer Sovereignty**(CS): For any receiver q_i , if ζ_i is sufficiently large, then q_i will be selected to receive the service. The payment sharing mechanism cannot arbitrarily exclude any users; the network has to allow users to receive the transmission if they are willing to pay a sufficiently high payment.

It is well-known [2] that a cross-monotonic budget-balanced *cost* sharing scheme ξ implies a group-strategyproof mechanism $M(\xi)$ that determines which receiver will get the service and at what price. Moulin and Shenker [2] also offered a characterization of a whole class of budget-balanced and group strategyproof mechanisms.

2.3 Problem Statement

Given a communication network $G = (V, E, c)$, where $V = \{v_1, \dots, v_n\}$ is the set of communication terminals, $E = \{e_1, e_2, \dots, e_m\}$ is the set of links. Every link e_i in the network has a cost c_i to transmit a unit size of data. Each of a set of terminals $R = \{q_1, q_2, \dots, q_{|R|}\} \subset V$ is willing to receive the data from a source node s . For simplicity, assume that $q_0 = s$ is the source node in one specific multicast and the size of the data is normalized to 1. In order to prevent monopoly, we assume that the network G is bi-connected.

There are several different scenarios, concerning whether the links are selfish and whether every receiver has a willing payment for receiving the data, which could be studied. In this paper, we always assume that each link is a selfish agent who will provide the relay service to receivers only if it gets a certain amount of payment. We assume throughout this paper that the selfish links in the network will not *collude* to improve their profits together. To design a multicast protocol, we first need to design a strategyproof mechanism $M_E = (\mathcal{O}_E, \mathcal{P}_E)$ such that

1. the links selected under the allocation rule \mathcal{O}_E form a topology (a tree, a mesh, a ring, etc) that spans the set of receivers R ;
2. \mathcal{P}_E defines a payment for each relay agent. Here a link is paid according to a strategyproof payment scheme if it is selfish; otherwise its payment will be its actual cost.

This has been well-studied [4, 1, 5]. In this paper, we concentrate on designing a fair payment sharing scheme ξ when the payment is computed by a mechanism $M = (\mathcal{O}_E, \mathcal{P}_E)$ with the property that \mathcal{O}_E has a constant approximation ratio.

We further assume that each receiver has a valuation of receiving the data from the source. Let ζ_i be the willing payment by receiver q_i , and ζ be the vector of the willing payments of the receivers. A mechanism $M_R = (\mathcal{O}_R, \mathcal{P}_R)$ determines the receivers that receive the service and the prices they pay. If the links are not selfish, then this is the cost sharing problem, for which Jain and Vazirani [9] proposed a $\frac{1}{2}$ -budget-balanced cross-monotonic cost sharing scheme in [9]. In this paper, instead of focusing on the cost sharing, we focus on the payment sharing. For simplicity of our notation, given a payment sharing scheme ξ , we always use $M(\xi)$ to denote its induced mechanism defined in [2].

Given a structure $H \subseteq G$, we use $\omega(H)$ to denote the sum of the costs of all agents in this network. If we change the cost of any agent i (link e_i or node v_i) to c'_i , we denote the new network as $G' = (V, E, c|^i c'_i)$, or simply $c|^i c'_i$. If we remove one agent i from the network, we denote it as $c|^i \infty$. We denote $G \setminus e_i$ as the network without link e_i , and denote $G \setminus v_i$ as the network without node v_i and all its incident links. For the simplicity of notations, we will use only the cost vector c to denote the network $G = (V, E, c)$ if no confusion is caused.

2.4 Previous Works

Strategyproof unicast and the efficient computing of the payment were addressed in [10–13]. Several results were proposed in the literature to deal with multicast in selfish networks [6, 5, 1, 4]. By assuming that the links (or nodes) of a network are selfish agents, a variety of strategyproof mechanisms were studied using different methods to construct the multicast tree.

Sharing the cost of providing service among the set of service receivers has been extensively studied in [2, 6, 7, 14–19] so some fairness is accomplished. Moulin and Shenker [2] showed that a budget-balanced and cross-monotone cost sharing scheme implies a group strategyproof mechanism when the receivers are selfish. Feigenbaum *et al.* [6], by assuming a *fixed* multicast structure, designed a strategyproof mechanism that selects a subset of receivers (each with a privately known willing payment) and then shares the cost of the multicast tree providing the service among the selected receivers so that budget-balance is achieved. Herzog *et al.* [7] considered an axiomatic approach to the problem, analyzing the implications of different distributive notions on the resulting allocations. A variety of cost allocation schemes were proposed and studied in [7]. Feigenbaum *et al.* [15] gave a group-strategyproof mechanism that can be computed with exponentially lower worst-case communication than the Shapley Value algorithm, but it might fail to achieve exact budget balance (albeit by a bounded amount). Feigenbaum *et al.* [16] also studied the message complexity of distributed cost sharing mechanism. They showed that any distributed algorithm, deterministic or randomized, that computes a (or an approximately, resp.) budget-balanced, group-strategyproof multicast cost-sharing mechanism must send $\Omega(n)$ (or $\Omega(\log n)$ resp.) bits over linearly many links in the worst case. Gupta *et al.* [17] gave a cross-monotone, fair cost-sharing method for the Single Source Rent-or-Buy network design game that is also β -budget-balanced, where $\beta = 4.6$. Notice that the traditional multicast game is a special case of the Rent-or-Buy game by letting $M = 1$ in [17]. Devanur *et al.* [18] obtained strategyproof cost allocations for the set cover game and the facility location game. The mechanisms for the set cover game in [18] is not group-strategyproof. Li *et al.* [20] gave for a general multi-set cover game the first group-strategyproof mechanism, which is also $\frac{1}{2^n}$ -budget-balanced. Maximizing profit in strategyproof multicast was studied in [9, 21]. Jain and Vazirani [9] gave a $1/2$ -budget-balanced cross-monotonic cost sharing scheme for the link weighted multicast using a prima-dual approach. Immorlica *et al.* [19] studied the maximum β for a cross-monotonic β -budget-balanced cost sharing schemes for a variety of games.

The only result that studied the payment (instead of cost) sharing among a set of receivers is by Wang *et al.* [1]. They specifically studied how to share the payment to all relay agent, defined by the mechanism using the shortest path tree as the multicast tree, among all receivers. They show that their payment sharing scheme is budget-balanced, cross-monotonic, and without free-rider (NFR). A receiver is called *free-rider* in [1] if its shared payment is less than $\frac{1}{r}$ of the payment of a unicast from the source. Here r is the total number of receivers. Notice that the cost of the shortest path tree could be as large as r times the cheapest cost multicast tree. In this paper, we will study the payment sharing of multicast when the tree is constructed by a certain polynomial-time method with a constant approximation ratio, such as the method by Takahashi and Matsuyama [22].

3 Payment Sharing For Multicast

In this section, we focus on the scenarios when the relay links are selfish. Thus, we study how to share the payment instead of the cost to the links among the receivers in a fair way.

3.1 Tree Construction

In practice, the *shortest path tree* (SPT), which is the union of the shortest pathes from source to all receivers, is most widely used as a multicast tree. We use $SPT(R, c)$ to denote the shortest path tree of a network when the network cost vector is c and receivers set is R . Although SPT is used very often in practice, the weight of $SPT(R, c)$ could be as large as $|R|$ times the optimal. Takahashi and Matsuyama [22] first gave a polynomial time algorithm computing a 2-approximation of the minimum cost Steiner tree (MCST). Then a series of results have been developed to improve the approximation ratio. The current best result is due to Robins and Zelikovsky [23], in which the authors presented a polynomial time method with approximation ratio $1 + \frac{\ln 3}{2}$. Due to its simplicity of construction, we will use algorithm in [22] to construct the multicast tree, and the resulting tree is denoted as $LST(R, c)$.

Algorithm 1 Construct the Steiner Tree (Takahashi and Matsuyama [22])

- 1: Initialize $LST(R, c) = \emptyset$.
 - 2: **repeat**
 - 3: **for** each receive r_i in R **do**
 - 4: Find the least cost path $LCP(s, q_i, c)$ between s and q_i .
 - 5: Find the receiver q_j with the minimum cost of the shortest path $LCP(s, q_j, c)$.
 - 6: Remove q_j from R and add $LCP(s, q_j, c)$ to $LST(R, c)$.
 - 7: Set all links' costs on $LCP(s, q_j, c)$ as 0.
 - 8: **until** R is empty.
 - 9: Output $LST(R, c)$.
-

3.2 Payment Computation

We now briefly review the truthful payment scheme for links when they are selfish. We continue to present some important properties stating the relations of different payment schemes, which are crucial to design our payment sharing scheme. Wang *et al.* [4] gave the truthful payment schemes for tree SPT and LST respectively and a more general framework to design truthful payment schemes for any given multicast structure is given in [1, 5]. We use $\mathcal{P}^{SPT}(R, c)$ and $\mathcal{P}^{LST}(R, c)$ to denote the payment scheme for tree SPT and LST respectively given the set R of receivers. We also use $\mathbb{P}^{SPT}(R, c)$ and $\mathbb{P}^{LST}(R, c)$ to denote the total payment to the links in the network for tree SPT and LST respectively, *i.e.* $\mathbb{P}^{SPT}(R, c) = \sum_{e_k \in E} \mathcal{P}_k^{SPT}(R, c)$ and $\mathbb{P}^{LST}(R, c) = \sum_{e_k \in E} \mathcal{P}_k^{LST}(R, c)$.

For a link $e_k \in SPT(R, c)$, we compute an intermediate payment $p_k^i(c)$ to link e_k for any receiver q_i as $p_k^i(c) = |\text{LCP}(s, q_i, c|^\infty)| - |\text{LCP}(s, q_i, c|^k 0)|$. The final payment to link $e_k \in SPT(R, c)$ is

$$\mathcal{P}_k^{SPT}(R, c) = \max_{q_i \in R} p_k^i(c) \quad (1)$$

For a link $e_k \in LST(R, c)$, the payment $\mathcal{P}_k^{LST}(R, c)$ is computed as follows (see [4] for more detail):

Lemma 1. *Given a network $G = (V, E, c)$, for any link e_k , we have $\mathcal{P}_k^{SPT}(R, c) \leq \mathbb{P}^{LST}(R, c)$.*

PROOF. We prove it by contradiction. For the sake of contradiction, we assume that $\mathcal{P}_k^{SPT}(R, c) > \mathbb{P}^{LST}(R, c)$. Without loss of generality, we assume that $\mathcal{P}_k^{SPT}(R, c) = \mathbb{P}^{LST}(R, c) + \delta$ where $\delta > 0$. We also assume that $\mathcal{P}_k^{SPT}(R, c) = p_k^j(c)$, *i.e.*, $\mathcal{P}_k^{SPT}(R, c) = |\text{LCP}(s, q_j, c|^\infty)| - |\text{LCP}(s, q_j, c|^k 0)|$, where $\text{LCP}(s, q_j, c|^k 0)$ is the shortest path between s and q_j . Let $\hat{c} = c|^k(\mathbb{P}^{LST}(R, c) + \frac{\delta}{2})$. Notice here that $\mathcal{P}_k^{LST}(R, c)$ is the maximum cost that e_k could declare such that it is selected (called cut-value in [5]). Thus we have $e_k \notin LST(R, \hat{c})$. Let $\Pi^{LST}(s, q_j)$ be the path between s and q_j in the tree $LST(R, \hat{c})$, then

Algorithm 2 Payment Scheme $\mathcal{P}^{LST}(R, c)$

- 1: Set $c_k = \infty$ and apply Algorithm 1. Without loss of generality, we assume that the cost vector in the beginning of i th round is $c^{(i)}$ and the path selected in round i is $P(s, q_{\sigma_i})$.
 - 2: **for** each round i **do**
 - 3: Set $c_k^{(i)} = 0$.
 - 4: Let $\text{LCP}_{e_k}(s, q_t, c^{(i)})$ be the path with the smallest weight among all paths between s and receivers in R .
 - 5: Define an intermediate payment $p_k^i(c)$ as $p_k^i(c) = |P(s, q_{\sigma_i})| - |\text{LCP}_{e_k}(s, q_t, c^{(i)})|$.
 - 6: The final payment $\mathcal{P}_k^{LST}(R, c)$ is $\mathcal{P}_k^{LST}(R, c) = \max_{i=1}^{|R|} p_k^i(c)$.
-

$e_k \notin \Pi^{LST}(s, q_j)$. Notice that $\text{LCP}(s, q_j, c|^{k\infty})$ is the shortest path between node s and q_j when link e_k is removed. Thus

$$\begin{aligned} \mathcal{P}_k^{SPT}(R, c) &\leq |\text{LCP}(s, q_j, c|^{k\infty})| \leq |\Pi^{LST}(s, q_j)| \leq |\text{LCP}(s, q_j, \hat{c})| \\ &\leq |\text{LCP}_{e_k}(s, q_j, \hat{c})| \leq \mathbb{P}^{LST}(R, c) + \frac{\delta}{2} < \mathbb{P}^{LST}(R, c) + \delta = \mathcal{P}_k^{SPT}(R, c), \end{aligned}$$

which is a contradiction. This finishes our proof. \square

Similarly, we have the following lemma.

Lemma 2. *Given a network $G = (V, E, c)$, for any link e_k , we have $\mathcal{P}_k^{LST}(R, c) \leq \mathbb{P}^{SPT}(R, c)$.*

PROOF. Recall that if $e_k \notin LST(R, c)$, then $\mathcal{P}_k^{LST}(R, c) = 0$. Thus, we only need to consider the case when $e_k \in LST(R, c)$. Without loss of generality, we assume that $\mathcal{P}_k^{LST}(R, c) = p_k^i(c)$ and $q_{\sigma_i} = q_j$. The tree shown in Figure 1 (a) is the tree in the beginning of iteration i , and the path $\Pi_3 = \text{LCP}_{-e_k}(s, q_j, c)$. Now we discuss by cases:

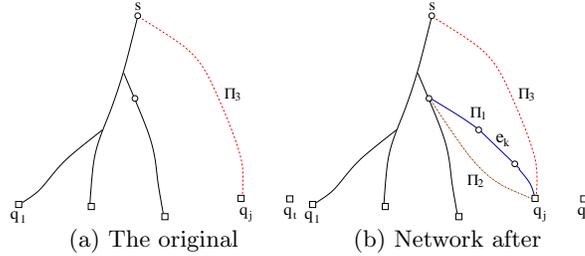


Fig. 1. The payment of LST.

Case 1: $\text{LCP}(s, q_j, c) = \text{LCP}_{-e_k}(s, q_j, c)$, *i.e.*, link e_k is not on the shortest path between s and q_j . In this case, $|\Pi_3| = |\text{LCP}(s, q_j, c)| \leq \omega(SPT(R, c)) \leq \mathbb{P}^{LST}(R, c)$.

Case 2: $\text{LCP}(s, q_j, c) = \text{LCP}_{e_k}(s, q_j, c)$, *i.e.*, link e_k is on the shortest path between s and q_j . Recall that $\mathcal{P}_k^{SPT}(R, c) = |\text{LCP}_{-e_k}(s, q_j, c)| - |\text{LCP}(s, q_j, c|^{k0})|$, thus we have

$$|\text{LCP}(s, q_j, c|^{k0})| = \sum_{e_i \in \text{LCP}(s, q_j, c|^{k0}) - e_k} c_i \leq \sum_{e_i \in SPT(R, c) - e_k} c_i \leq \sum_{e_i \in SPT(R, c) - e_k} \mathcal{P}_i^{SPT}(R, c)$$

Therefore

$$|\Pi_3| = |\text{LCP}_{-e_k}(s, q_j, c)| = \mathcal{P}_k^{SPT}(R, c) + |\text{LCP}(s, q_j, c|^{k0})| \leq \mathcal{P}_k^{SPT}(R, c) + \sum_{e_i \neq e_k} \mathcal{P}_i^{SPT}(R, c) = \mathbb{P}^{SPT}(R, c)$$

This proves that $|\Pi_3| \leq \mathbb{P}^{SPT}(R, c)$ no matter whether e_k is on the shortest path $\text{LCP}(s, q_j, c)$ or not. Now we consider the iteration i . For simplicity, we assume that $\Pi_1 = \text{LCP}_{e_k}(s, q_t, c^{(i)})$ and $\Pi_2 = P(s, q_{\sigma_i})$. From the assumption that $\mathcal{P}_k^{LST}(R, c) = p_k^i(c)$, we have

$$\mathcal{P}_k^{LST}(R, c) = p_k^i(c) = |P(s, q_{\sigma_i})| - |\text{LCP}_{e_k}(s, q_t, c^{(i)})| = |\Pi_2| - |\Pi_1| \leq |\Pi_3| - |\Pi_1| \leq |\Pi_3| \leq \mathbb{P}^{SPT}(R, c)$$

This finishes our proof. \square

Theorem 1. Given a n -nodes network $G = (V, E, c)$, $\frac{\mathbb{P}^{LST}(R, c)}{n} \leq \mathbb{P}^{SPT}(R, c) \leq n \cdot \mathbb{P}^{LST}(R, c)$.

3.3 Payment Sharing Scheme

Recall that, when the links in the network are selfish, we should give links some payments that are typically larger than their costs. Thus, we need to study how to share the payment instead of cost of the multicast tree among the receivers in a fair way, which we call *payment sharing*. In this situation, even for the tree SPT, the ELSD sharing scheme is not a reasonable sharing scheme [1]. Then a reasonable payment sharing scheme is given in [1]. Our new payment sharing scheme is built upon that payment sharing scheme. Thus, for completeness of our presentation, we review their payment sharing scheme for the payment computed by the mechanism (SPT, \mathcal{P}) . The basic idea of their payment sharing scheme is as follows. Notice that a final payment to an agent j is the maximum of payments p_j^i by all receivers. Since different receivers may have different “views” on the payment agent j deserves to receive, the final payment \mathcal{P}_j should be shared *proportionally* to their views, not *equally* among them as what we do for cost-sharing. Figure 3.3 illustrates the payment sharing scheme that follows. Without loss of generality, assume that the link agent j (denoted

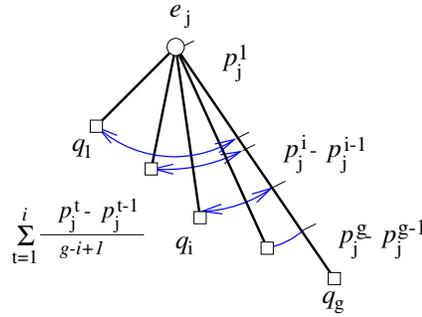


Fig. 2. Share the payment to service providers among receivers fairly.

by a circle) has g downstream receivers, say q_1, q_2, \dots, q_g , and the payments of the receivers to j are $0 \leq p_j^1 \leq p_j^2 \leq \dots \leq p_j^g$, *i.e.*, $p_j = p_j^g$. We then divide the payment p_j into n portions: $p_j^1, p_j^2 - p_j^1, \dots, p_j^i - p_j^{i-1}, \dots, p_j^g - p_j^{g-1}$. Each portion $p_j^i - p_j^{i-1}$ is then equally shared among the last $g - i + 1$ elements, which have the largest $g - i + 1$ payments. Regarding the payment sharing scheme $\xi_i^{SPT}(R, d)$, following theorem is proved in [1].

Theorem 2 (Wang, Li *et al.* [1]). The payment sharing scheme $\xi_i^{SPT}(R, d)$ is reasonable, *i.e.*, satisfies BB, NPT, NFR and CM.

For a mechanism based on tree LST, if a payment sharing scheme ξ is β -budget-balanced and cross-monotonic, then we have

Theorem 3. If a payment sharing scheme ξ is β -budget-balanced and cross-monotonic, then $\beta = O(\frac{1}{n})$. Here n is the size of the network.

PROOF. We prove it by presenting a network example here. The network and the costs of the edges are shown in Figure 3. There are n nodes between v_4 and q_1 . The cost of link $v_i v_{i+1}$ is ϵ , for $5 \leq i \leq n + 3$. Let ξ^{LST} be a payment sharing scheme for the mechanism (LST, \mathcal{P}^{LST}) that is β -budget-balanced and cross-monotonic. Then from the β -budget-balance property we have $\xi_1^{LST}(q_1, c) \leq \mathcal{P}_1(q_1, c) = 2.6$ and $\xi_2^{LST}(q_2, c) \leq \mathcal{P}_1(q_2, c) = 2.9$. When the receiver set is $q_1 \cup q_2$, the cross-monotonicity property implies that $\xi_1^{LST}(q_1 \cup q_2, c) + \xi_2^{LST}(q_1 \cup q_2, c) \leq \xi_1^{LST}(q_1, c) + \xi_2^{LST}(q_2, c) = 5.5$. Notice that $\mathcal{P}(q_1 \cup q_2) = 6 + 0.5 \cdot n$. Thus, $\beta \leq \frac{\xi_1^{LST}(q_1 \cup q_2, c) + \xi_2^{LST}(q_1 \cup q_2, c)}{\mathcal{P}(q_1 \cup q_2)} = \frac{5.5}{6 + 0.5 \cdot n} = O(\frac{1}{n})$. This finishes our proof. \square

The above theorem shows the limitations on the payment sharing scheme when the payment is computed by mechanism (LST, \mathcal{P}) . In the following, we present a payment sharing scheme for LST that achieves $\frac{1}{n^2}$ -budget-balance and cross-monotonicity.

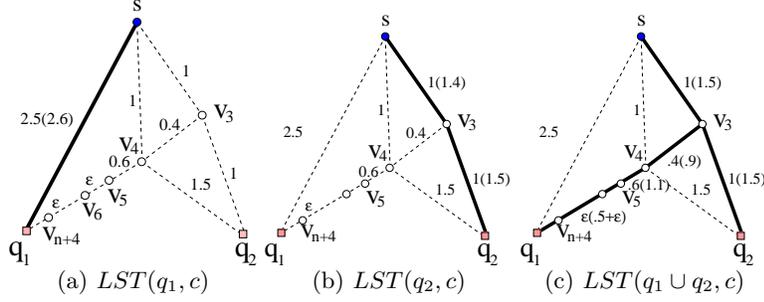


Fig. 3. A bad example of payment sharing of LST.

Algorithm 3 Payment Sharing Scheme for LST.

- 1: Compute the payment sharing $\xi_k^{SPT}(R, c)$.
 - 2: For each receiver q_k , set $\xi_k^{LST}(R, c) = \frac{\xi_k^{SPT}(R, c)}{n}$, where n is the number of the nodes in G .
-

Theorem 4. *The payment sharing scheme 3 satisfies NPT, CM, $\frac{1}{n^2}$ -budget-balance, and is in the core.*

PROOF. Recall that $\xi_k^{LST}(R, c) = \frac{\xi_k^{SPT}(R, c)}{n}$. From Theorem 2, we obtain that $\xi^{LST}(R, c)$ satisfies NPT and CM directly. Thus, we only need to prove that $\xi^{LST}(R, c)$ is $\frac{1}{n^2}$ -budget-balance and in the core.

To prove the $\frac{1}{n^2}$ -budget-balance property, we need to show that $\frac{\mathbb{P}^{LST}(R, c)}{n^2} \leq \sum_i \xi_i^{LST}(R, c) \leq \mathbb{P}^{LST}(R, c)$. From Theorem 1, we have $\frac{\mathbb{P}^{LST}(R, c)}{n^2} \leq \frac{\mathbb{P}^{SPT}(R, c)}{n} \leq \mathbb{P}^{LST}(R, c)$. From Theorem 2, we know that $\xi^{SPT}(R, c)$ satisfies BB. Thus, $\frac{\mathbb{P}^{SPT}(R, c)}{n} = \frac{\sum_i \xi_i^{SPT}(R, c)}{n} = \sum_i \xi_i^{LST}(R, c)$. Consequently, we have $\frac{\mathbb{P}^{LST}(R, c)}{n^2} \leq \sum_i \xi_i^{LST}(R, c) \leq \mathbb{P}^{LST}(R, c)$. This proves that the payment sharing scheme is $\frac{1}{n^2}$ -budget-balanced.

We then show that it is also in the core. This is due to the fact that $\sum_i \xi_i^{LST}(R, c) = \sum_i \frac{\xi_i^{SPT}(R, c)}{n} = \frac{\mathbb{P}^{SPT}(R, c)}{n} \leq \mathbb{P}^{LST}(R, c)$, since ξ^{SPT} is budget-balanced. \square

Notice that there is a gap between the upper bound $O(\frac{1}{n})$ and lower bound $\Omega(\frac{1}{n^2})$ on β for β -budget-balanced cross-monotonic payment sharing scheme. A possible future work is to close the gap.

3.4 Satisfy Budget Balance with γ -Relaxed Core

In Section 3.3, we present a payment sharing scheme $\xi^{LST}(R, c)$ that is $\frac{1}{n^2}$ -budget balanced, cross-monotonic, and in the core. This payment sharing scheme will most likely run to deficit. However, under certain circumstances, there is no outsider who will afford the portion of the payment to the link that is not covered by the receivers. Thus, sometimes we would like to achieve the budget balance property while sacrifice some other properties such as core. Recall that a payment sharing scheme satisfies the core property if, for any receiver set $S \in R$, $\sum_{q_i \in S} \xi_i(R, c) \leq \mathcal{P}(S, c)$. We generalize the core property as follows. A payment sharing scheme ξ is γ -relaxed core if, for any receiver set $S \in R$, $\sum_{q_i \in S} \xi_i(R, c) \leq \gamma \cdot \mathcal{P}(S, c)$. Following theorem shows that a payment sharing scheme that is α -budget balanced and core implies a payment sharing scheme that is budget balanced and $\frac{1}{\alpha}$ -relaxed core.

Theorem 5. *For any payment sharing scheme ξ that is α -budget-balanced and core, then payment sharing scheme $\hat{\xi}$*

$$\hat{\xi}_i(R, c) = \xi_i(R, c) \cdot \frac{\mathcal{P}(R, c)}{\sum_{q_i \in R} \xi_i(R, c)}$$

is budget balanced and $\frac{1}{\alpha}$ -relaxed core.

PROOF. It is obvious that $\hat{\xi}$ is budget-balanced. Since the payment sharing scheme is α -budget-balanced, we have $\sum_{q_i \in R} \xi_i(R, c) \geq \alpha \cdot \mathcal{P}(R, c)$. Thus, $\hat{\xi}_i(R, c) = \xi_i(R, c) \cdot \frac{\mathcal{P}(R, c)}{\sum_{q_i \in R} \xi_i(R, c)} \leq \frac{\xi_i(R, c)}{\alpha}$. For any receiver set $S \in R$, we have $\sum_{q_i \in S} \hat{\xi}_i(R, c) \leq \frac{\sum_{q_i \in S} \xi_i(R, c)}{\alpha} \leq \frac{\mathcal{P}(S, c)}{\alpha}$, which implies that $\hat{\xi}$ is budget balanced and $\frac{1}{\alpha}$ -relaxed core. This finishes our proof. \square

4 Selfish Relay Links and Receivers

So far, we assume that the receivers will pay the fair amount of sharing of payment to receive data using multicast. In other words, each receiver q_i is assumed to pay its sharing according to some payment sharing scheme ξ . In practice, each individual receiver often has a maximum valuation indicating how much it is willing to pay to receive the information from the source. A receiver will choose to receive the information if and only if the charge by the source or by the relay agents is at most its valuation. Furthermore, receiver could also be *non-cooperative* and selfish: it will always maximize its profit by manipulating its reported valuation, should it be possible. This makes the multicast design even harder when both the relay agents and the receivers could be selfish. It is well-known that a cost sharing scheme satisfying CM implies a *group-strategyproof* mechanism [2]. Thus, when each receiver q_i has a valuation ζ_i for receiving the data, *i.e.*, it is willing to pay at most ζ_i for the data, the first intuition is that we can design a payment sharing mechanism as follows.

Algorithm 4 Payment Sharing Mechanism for Tree LST

- 1: $S \leftarrow R$, where R is the set of possible receivers.
 - 2: **repeat**
 - 3: Construct the tree $LST(S, c)$.
 - 4: For each receiver $q_i \in S$, we compute the payment sharing $\xi_i^{LST}(S, c)$ based on the declared cost of all possible relay agents.
 - 5: For each receiver $q_i \in S$, the receiver q_i is removed from S if $\xi_i^{LST}(S, c) > \zeta_i$, *i.e.*, $S \leftarrow S - \{q_i\}$ if $\xi_i(S, c) > \zeta_i$.
 - 6: **until** no receiver is removed in this round
 - 7: All remaining receivers, say $\hat{R} \subseteq R$ will receive the multicast data and pay a sharing $\xi_i^{LST}(\hat{R}, d) \leq \zeta_i$.
-

Intuitively, when a relay agent is *not* in the multicast tree LST , the relay agent could not get positive utility no matter what cost it declares. On the other hand, when a relay agent is in the relay agent could manipulate its declared cost, and in consequence the payment sharing for its downstream receivers, such that it is still in the final tree. There are two ways that a relay agent could change the payment sharing of its downstream receivers: reporting a higher cost or lower cost. Following theorem shows the negative result about payment sharing mechanism described by Algorithm 4.

Theorem 6. *Payment sharing mechanism 4 is not strategyproof, and moreover, some links may have incentives to lie up and lie down.*

PROOF. We first show by example that some links may have incentives to lie up. Figure 4 illustrates such an example of reporting a lower cost. Here the private valuations of receivers q_1 and q_2 are 2.4 and 3.4

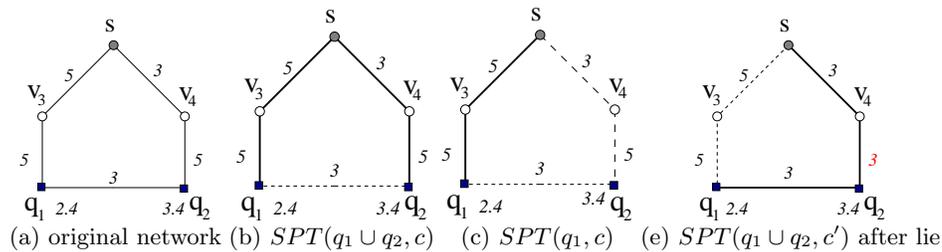


Fig. 4. A relay agent could lie down its cost to improve its utility when use payment sharing mechanism 4.

respectively. The true cost of links are $c(sv_3) = 5$, $c(sv_4) = 3$, $c(v_3q_1) = 5$, $c(v_4q_2) = 5$, and $c(q_1q_2) = 3$. Observe that when link v_4q_2 truthfully reports its cost as 5, the multicast tree is formed by links sv_3 , v_3q_1 , sv_4 and v_4q_2 . Figure 4 (b) illustrates the tree SPT constructed based on the reported cost. In addition, the payments to each individual links are $p_{sv_4} = c(sv_3) + c(v_3q_1) + c(q_1q_2) - c(v_4q_2) = 8$, $p_{v_4q_2} = 10$, $p_{sv_3} = 6$, $p_{v_3q_1} = 6$, and the payments to all other links are 0. Consider two receivers q_1 and q_2 . The total payment

shared by receiver q_1 is $\frac{p_{sv_3} + p_{v_3q_1}}{5} = \frac{12}{5} = 2.4$, which is not larger than its willing payment 2.4. Similarly, the total payment shared by q_2 is $\frac{p_{sv_4} + p_{v_4q_2}}{5} = 185$, which is larger than its maximum willing payment 3.4. Consequently, the receiver q_2 will *not* join the multicast (illustrated by Figure 4 (c)). In other words, link v_4q_2 gets payment 0.

Let's see what happens if link v_4q_2 lies its cost down to $3 < c(v_4q_2)$ (illustrated by Figure 4 (d)). Figure 4 (e) shows the multicast tree constructed in this scenario. It is easy to see that when link v_4q_2 reported its cost as 3, the payments to each individual links are $p_{sv_4} = 10$, $p_{v_4q_2} = 10$, $p_{q_1q_2} = 4$, and the payments to all other links are 0. Let's see what is the sharing of receiver q_2 in this case. It is not difficult to see that $p_{sv_4}^{q_2} = 10$, $p_{sv_4}^{q_1} = 4$, $p_{v_4q_2}^{q_2} = 10$, $p_{v_4q_2}^{q_1} = 4$. Thus, the sharing of receiver q_2 to the payment 10 to link sv_4 is $\frac{p_{sv_4}^{q_1} + (p_{sv_4}^{q_2} - p_{sv_4}^{q_1})}{5} = 1.6$. Similar calculation shows that the charge to receiver q_2 will be $\frac{16}{5} = 3.2$ and the charge to receiver q_1 is $\frac{7}{5} = 1.4$. Thus, both receivers q_1 and q_2 will join the multicast now. Then, the link v_4q_2 gets a payment 10 when it lies its cost down to 3.

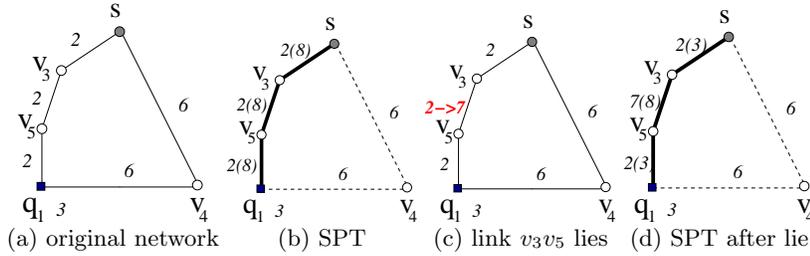


Fig. 5. A relay agent could also decrease its utility when it lies down cost.

We then show by example that a relay agent could also improve its utility by reporting a cost higher than its true cost. The intuition is as follows: reporting a higher cost such that it is still in LST and SPT, which in turn could decrease the payment to other relay agents. Consequently, the payment sharing of its downstream receivers is then decreased. Thus, the relay agent could be selected now when the new payment sharing of a receiver is smaller than its valuation. Consider the network shown in Figure 5 (a). Assume that the valuation of receiver q_1 is 3. The tree $LST(q_1, c) = SPT(q_1, c)$ and is shown in Figure 5 (b). The payment to each link is shown beside its cost. Notice that $\xi_1^{LST}(q_1, c) = \frac{\xi_1^{SPT}(q_1, c)}{5} = 4.8 > 3$. Thus no receiver will receive the service and the multicast tree is empty. Therefore, link v_3v_5 's utility is 0 when reporting its cost truthfully. Assume that it lies its cost as 7 and c' is the cost vector. The tree $LST(q_1, c') = SPT(q_1, c')$ and is shown in Figure 5 (b), the payment to every link is shown beside its cost. $\xi_1^{LST}(q_1, c') = \frac{\xi_1^{SPT}(q_1, c')}{5} = 2.8 < 3$. Thus receiver q_1 receives the service, and link v_3v_5 is in the multicast tree who will receive a payment 8. Consequently, link v_3v_5 has utility 6, which is larger than its utility 0 by reporting its cost truthfully. \square

The above theorem shows that combining two strategyproof mechanisms (one for selfish relay links, and one for selfish receivers) does not necessarily imply a strategyproof mechanism for the situation when both are simultaneously selfish.

5 Conclusion

Sharing the multicast cost among the receivers in a certain fair way has been studied widely in the literature. In this paper we studied how the payment should be shared among the receivers when the payment is computed by a mechanism $M = (LST, \mathcal{P})$. Here, LST is the multicast tree construction method that can achieve the cost at most 2 times the optimum. We described a payment sharing scheme that is $\frac{1}{n^2}$ -budget-balanced, cross-monotonic, and in the core. We also proved that there is no payment sharing scheme that is β -budget-balanced and cross-monotonic for $\beta = \Omega(\frac{1}{n})$. When both the relay agents and the receivers are selfish, we showed a negative results: combining the strategyproof mechanism M and the fair payment sharing scheme $\xi^{LST}(\cdot)$ does not necessarily imply a strategyproof mechanism.

There are two future research directions. The first one is, for the β -budget-balanced and cross-monotonic payment sharing scheme for $M = (LST, \mathcal{P})$, to close the gap between the upper bound $\frac{1}{n}$ on β and the

achievable lower bound $\frac{1}{n^2}$. The second direction is to design an overall strategyproof mechanism $M = (\mathcal{O}, \mathcal{P})$ that will form an approximately efficient multicast tree, decide the payment to each relay links, determine which receiver will receive the data and at what price. We have to make sure that both the relay links and the receivers will maximize their profit when they report their cost (or willing payment) truthfully.

References

1. Wang, W., Li, X.Y., Sun, Z., Wang, Y.: Design multicast protocols for non-cooperative networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Communication Society (INFOCOM). (2005) To appear.
2. Moulin, H., Shenker, S.: Strategyproof sharing of submodular costs: Budget balance versus efficiency. In: Economic Theory. Volume 18. (2001) 511–533
3. Feigenbaum, J., Papadimitriou, C.H., Shenker, S.: Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences* **63** (2001) 21–41
4. Wang, W., Li, X.Y., Wang, Y.: Truthful multicast in selfish wireless networks. In: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking. (2004)
5. Kao, M.Y., Li, X.Y., Wang, W.: Towards truthful mechanisms for binary selection problems: A general design framework. In: ACM EC, accepted for publication. (2005)
6. Feigenbaum, J., Papadimitriou, C.H., Shenker, S.: Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences* **63** (2001) 21–41
7. Herzog, S., Shenker, S., Estrin, D.: Sharing the “cost” of multicast trees: an axiomatic analysis. *IEEE/ACM Transactions on Networking* **5** (1997) 847–860
8. Osborne, M.J., Rubinstein, A.: A course in game theory. The MIT Press (2002)
9. Jain, K., Vazirani, V.V.: Applications of approximation algorithms to cooperative games. In: Proceedings of the 3rd ACM Conference on Electronic Commerce. (2001) 364–372
10. Nisan, N., Ronen, A.: Algorithmic mechanism design. In: Proceedings of the 31st Annual ACM Symposium on Theory of Computing. (1999) 129–140
11. Feigenbaum, J., Papadimitriou, C., Sami, R., Shenker, S.: A BGP-based mechanism for lowest-cost routing. In: Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing. (2002) 173–182
12. Hershberger, J., Suri, S.: Vickrey pricing in network routing: Fast payment computation. In: Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science. (2001) 252–259
13. Wang, W., Li, X.Y.: Truthful low-cost unicast in selfish wireless networks. In: Proceedings of the 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks. (2004)
14. Libman, L., Orda, A.: Atomic resource sharing in noncooperative networks. *Telecommunication Systems* **17** (2001) 385–409
15. Archer, A., Feigenbaum, J., Krishnamurthy, A., Sami, R., Shenker, S.: Approximation and collusion in multicast cost sharing. *Games and Economic Behavior* **47** (2004) 36–71
16. Feigenbaum, J., Krishnamurthy, A., Sami, R., Shenker, S.: Hardness results for multicast cost sharing (extended abstract). *Theoretical Computer Science* **304** (2003) 215 – 236
17. Gupta, A., Srinivasan, A., Tardos, E.: Cost sharing mechanisms for network design. In: Proceedings of APPROX04. (2004)
18. Devanur, N.R., Mihail, M., Vazirani, V.V.: Strategyproof cost-sharing mechanisms for set cover and facility location games. In: ACM Electronic Commerce (EC03). (2003)
19. Immorlica, N., Mahdian, M., Mirrokni, V.S.: Lower bounds for cost sharing and group-strategyproof mechanisms. In: ACM SODA. (2004)
20. Li, X.Y., Sun, Z., Wang, W.: Cost sharing and strategyproof mechanisms for set cover games. In: Proceedings of the 22nd International Symposium on Theoretical Aspects of Computer Science. (2005) To appear.
21. Chawla, S., Kitchin, D., Rajan, U., Ravi, R., Sinha, A.: Profit maximizing mechanisms for the extended multicasting games. Technical Report CMU-CS-02-164, Carnegie Mellon University (2002)
22. Takahashi, H., Matsuyama, A.: An approximate solution for the steiner problem in graphs. *Mathematical Japonica* **24** (1980) 573–577
23. Robins, G., Zelikovsky, A.: Improved steiner tree approximation in graphs. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California, United States, Society for Industrial and Applied Mathematics (2000) 770–779