Yale University
Department of Computer Science

Approximation and Collusion in Multicast Cost Sharing

Joan Feigenbaum[1] Arvind Krishnamurthy[1]
Rahul Sami[1] Scott Shenker[2]

YALEU/DCS/TR-1212
April 22, 2001

[1]Yale Univ., Computer Science, New Haven, CT 06520-8285, {jf, arvind, sami}@cs.yale.edu
[2]ACIRI/ICSI, 1947 Center Street, Berkeley, CA 94704-1198, shenker@icsi.berkeley.edu

Approximation and Collusion in Multicast Cost Sharing

Joan Feigenbaum* Arvind Krishnamurthy* Rahul Sami* Scott Shenker†

Abstract

We investigate multicast cost sharing from both computational and economic points of view. Recent work in economics [MS97] leads naturally to the consideration of two mechanisms: Marginal Cost (MC), which is efficient and strategyproof, and Shapley value (SH), which is budget-balanced and group-strategyproof and, among all mechanisms with these two properties, minimizes the worst-case efficiency loss. Subsequent work in computer science [FPS00] shows that the MC mechanism can be computed by a simple, distributed algorithm that uses only two messages per link of the multicast tree but that computing the SH mechanism seems, in the worst case, to require a number of messages that is quadratic in the size of the multicast tree.

Here, we extend these results in two directions:

- We give a group-strategyproof mechanism that exhibits a tradeoff between the other properties of the Shapley value: It can be computed by a distributed algorithm that uses significantly fewer messages than the natural SH algorithm (exponentially fewer in the worst case), but it might fail to achieve exact budget balance or exact minimum efficiency loss (albeit by a bounded amount).
- We completely characterize the groups that can strategize successfully against the MC mechanism.

Keywords: Approximation, Cost Sharing, Incentive-Compatibility, Internet Algorithms, Mechanism Design, Multicast

*Yale Univ., Computer Science, New Haven, CT 06520-8285, {jf, arvind, sami}@cs.yale.edu

†ACIRI/ICSI, 1947 Center Street, Berkeley, CA 94704-1198, shenker@icsi.berkeley.edu

1 Introduction

Despite their prominent role in some of the more applied areas of computer science, incentives have rarely been an important consideration in traditional algorithm design where, typically, users are assumed either to be *cooperative* (*i.e.*, to follow the prescribed algorithm) or to be *adversaries* who "play against" each other. In contrast, the *selfish* users in game theory are neither cooperative nor adversarial. Although one cannot assume that selfish users will follow the prescribed algorithm, one can assume that they will respond to incentives. Thus, one need not design algorithms that achieve correct results in the face of byzantine behavior on the part of some users, but one does need algorithms that work correctly in the presence of predictably selfish behavior. This type of "correctness" is a primary goal of economic mechanism design, but standard notions of algorithmic efficiency are not.

In short, the economics literature traditionally stressed incentives and downplayed computational complexity, and the theoretical computer science literature traditionally did the opposite. The emergence of the Internet as a standard platform for distributed computation has changed this state of affairs. In particular, the work of Nisan and Ronen [NR99] inspired the design of algorithms for a range of problems, including scheduling, load balancing, shortest paths, and combinatorial auctions, that satisfy both the traditional economic definitions of incentive compatibility and the traditional computer science definitions of efficiency.

One of the problems that has been studied is *multicast cost sharing*, and we continue the study here. Specifically, we address the computational properties of the Shapley Value mechanism for multicast cost sharing and the game-theoretic properties of the Marginal Cost mechanism. We state our results at the end of this section, after a brief review of previous work and development of the necessary terminology and notation.

Algorithmic mechanism design:.

Nisan and Ronen [NR99] focused the attention of the theoretical computer science community on the study of *algorithmic mechanism design* by adding computational efficiency to the set of concerns that must be addressed in the design of incentive-compatible mechanisms. This emerging field of study lies at the intersection of game theory, algorithmics, and distributed computing and is of great interest because of the growth of Internet-enabled commerce, in which users are assumed to be selfish and responsive to well-defined incentives. There are growing bodies of relevant literature on incentive-compatibility questions in distributed computations in both theoretical computer science (see, *e.g.*, [NR99, FPS00, NR00, Nis00, LN00, GHW01, GH00, JV01]) and the "distributed agents" part of AI [MT99, Par99, PU00, San99, ST, Wel93, WWWM].

[NR99] proposes the following (centralized computational) model for the design and analysis of algorithms in which the participants act according to their own self-interest: There are n agents. Each agent i, for $i \in \{1, \ldots, n\}$, has some private information t^i, called its *type*. For each mechanism-design problem, there is an *output specification* that maps each type vector $t = t^1 \ldots t^n$ to a set of allowed outputs $o \in \mathcal{O}$. Agent i's preferences are given by a *valuation function* v^i that assigns a real number $v^i(t^i, o)$ to each possible output o.

A mechanism defines for each agent i a set of strategies A^i. For each input vector $(a^1, \ldots, a^n)$, *i.e.*, the vector in which i "plays" $a^i \in A^i$, the mechanism computes an *output* $o = o(a^1, \ldots, a^n)$ and a *payment vector* $p = (p^1, \ldots, p^n)$, where $p^i = p^i(a^1, \ldots, a^n)$. Each agent therefore seeks to maximize $v^i(t^i, o) + p^i$. A *strategyproof* mechanism is one in which each agent maximizes this quantity by simply giving his type t^i as input regardless of what other agents do.[1] Thus, the mechanism wants all agents to report their private types truthfully in order to achieve optimal resource allocation, and it is allowed to provide incentives for them to do so by paying them.

Succinctly stated, Nisan and Ronen's contribution to the mechanism-design framework is the notion of a (centralized) *polynomial-time mechanism*, *i.e.*, one in which $o(\cdot)$ and the $p^i(\cdot)$'s are polynomial-time computable. They also provide strategyproof, polynomial-time mechanisms for some concrete problems, including shortest paths and task allocation.

[1] The "solution concept" used in [NR99] and other papers on algorithmic mechanism design is *dominant strategies*. This is not the only possible solution concept, but it is the most appropriate one for Internet computation. The reason for this is beyond the scope of this paper; for a detailed discussion, see [FS97].

Network complexity of mechanisms:.

To achieve feasible algorithmic mechanisms within an Internet infrastructure, a centralized computational model does not suffice. After all, if one assumes that massive numbers of far-flung, independent agents are involved in an optimization problem, one cannot reasonably assume that a *single, centralized* mechanism can receive all of the inputs and disseminate the output and payment values for each user in an efficient manner. The first work to address this issue is the multicast cost-sharing paper of Feigenbaum, Papadimitriou, and Shenker [FPS00]. This work puts forth a general concept of "network complexity" that requires a distributed algorithm executed over an interconnection network T to be modest in four respects: the total number of messages that agents send over T (ideally, this should be linear in $|T|$), the maximum number of messages sent over any one link in T (ideally, this should be constant, to avoid "hot spots" altogether), the maximum size of a message, and the local computational burden on agents.[2] This notion of network complexity allows the mechanism designer to evaluate the feasibility of executing the algorithmic mechanism in a decentralized setting. [FPS00] exercises the notion of network complexity by studying the communication requirements of cost-sharing mechanisms for multicast transmissions. It suggests that a distributed algorithm should have maximum message size $poly(m, \log n, \log |T|)$, where m is the maximum size of a numerical input. All algorithms considered here and in [FPS00] satisfy this constraint.

Multicast Transmissions:.

We use the following multicast transmission model, following [FPS00]: There is a user population P residing at a set of network nodes N, which are connected by bidirectional network links L. The multicast flow originates from a source node $\alpha_s \in N$; given any set of receivers $R \subseteq P$, the transmission flows through a *multicast tree* $T(R) \subseteq L$ rooted at α_s and spans the nodes at which users in R reside. It is assumed that there is a *universal* tree $T(P)$ and that, for each subset $R \subseteq P$, the multicast tree $T(R)$ is merely the minimal subtree of $T(P)$ required to reach the elements in R.[3]

Cost-Sharing Mechanisms:.

Each link $l \in L$ has an associated cost $c(l) \geq 0$ that is known by the nodes on each end, and each user i assigns a utility value u_i to receiving the transmission. A cost-sharing mechanism determines which users receive the multicast transmission and how much each receiver is charged. We let $x_i \geq 0$ denote how much user i is charged and σ_i denote whether user i receives the transmission; $\sigma_i = 1$ if the user receives the multicast transmission, and $\sigma_i = 0$ otherwise. We use u to denote the input vector $(u_1, u_2, \cdots, u_{|P|})$. The mechanism M is then a pair of functions $M(u) = (x(u), \sigma(u))$. The receiver set for a given input vector is $R(u) = \{i \mid \sigma_i = 1\}$. A user's individual *welfare* is therefore given by $w_i = \sigma_i u_i - x_i$. The cost of the tree $T(R)$ reaching a set of receivers R is $c(T(R))$, and the overall welfare, or *net worth*, is $NW(R) = u_R - c(T(R))$, where $u_R = \sum_{i \in R} u_i$ and $c(T(R)) = \sum_{l \in T(R)} c(l)$. The overall welfare measures the total benefit of providing the multicast transmission (the sum of the valuations minus the cost).

A cost-sharing mechanism fits into Nisan and Ronen's algorithmic mechanism design framework in the following manner. The private type information is just the user's individual utility for receiving the transmission, $t^i = u_i$. The mechanism computes the output specification $o = \sigma$ and the payment vector $p = -x$ for the following valuation function: $v^i(t, o) = t^i$ if $o_i = 1$ and 0 otherwise. Each user seeks to maximize $v^i(t^i, o) + p^i = \sigma_i u_i - x_i$, which is the user's individual welfare, w_i.

A strategyproof cost-sharing mechanism is one that satisfies the property that $w_i(u) \geq w_i(u|^i \mu_i)$, for all u, i, and μ_i. (Here, $(u|^i \mu_i)_j = u_j$, for $j \neq i$, and $(u|^i \mu_i)_i = \mu_i$.) Strategyproofness does not preclude the possibility of a group of users colluding to improve their individual welfares. Any reported utility profile v can be considered a group strategy for the group $S = \{i \mid v_i \neq u_i\}$. A mechanism M is *group strategyproof* (GSP) if there is no group strategy such that at least one member of the strategizing group improves his

[2]The term "network complexity" does not appear in the preliminary version of this paper that is in the STOC 2001 proceedings. It is defined in Section 2 of the forthcoming journal version, which is available at `http://cs-www.cs.yale.edu/homes/jf/FPS.pdf`.

[3]This approach is consistent with the design philosophy embedded in essentially all current multicast routing proposals (see, for example, [BFC93, DC90, DEF+96, HC99, PLB+99]). [FPS00] actually shows that the alternative, seemingly attractive approach of finding net-worth-maximizing multicast trees in a general directed graph is computationally infeasible.

welfare while the rest of the members do not reduce their welfare. In other words, if M is GSP, the following property holds for all S:

$$\textit{either } w_i(v) = w_i(u) \ \forall i \in S$$

$$\text{or } \exists i \in S \text{ such that } w_i(v) < w_i(u)$$

Following [MS97, FPS00], we also require the mechanism to satisfy the following three basic requirements: No Positive Transfers (NPT): $x_i(u) \geq 0$, or, in other words, the mechanism cannot *pay* receivers to receive the transmission. Voluntary Participation (VP): $w_i(u) \geq 0$; this implies that $x_i = 0$ whenever $\sigma_i = 0$ and that users are always free to not receive the transmission and not be charged. Consumer Sovereignty (CS): $\sigma_i = 1$ if u_i is big enough; this implies that the network cannot exclude any user who is willing to pay a sufficiently large amount regardless of other users' utilities.

In addition to these basic requirements, there are certain other desirable properties that one could expect a cost-sharing mechanism to possess. A cost-sharing mechanism is termed *efficient* if it maximizes the overall welfare, and it is said to be *budget balanced* if the revenue raised from the receivers covers the cost of the transmission exactly.

It is a classical result in game theory [GKL76, Rob79] that a strategyproof cost-sharing mechanism cannot be both budget-balanced and efficient. Moulin and Shenker [MS97] have shown that there is only one strategyproof mechanism, Marginal Cost (MC), that satisfies the basic requirements and is efficient. They have also shown that, while there are many group-strategyproof mechanisms that are budget-balanced but not efficient, the most natural budget-balanced mechanism to consider is the Shapley value (SH), because it minimizes the worst-case efficiency loss. The SH mechanism has the users share the transmission costs in an equitable fashion; the cost of a link is shared equally by all users that receive the transmission through the link.

Our Results:.

The foregoing discussion makes it clear that the computational and game theoretic properties of SH and MC mechanisms are worthy of study. It is easy to see (and is noted in [FPS00]) that both are polynomial-time computable by centralized algorithms. [FPS00] further shows that there is a distributed algorithm that computes MC using only two messages per link. By contrast, [FPS00] notes that the obvious algorithm that computes SH requires $\Omega(|P|\cdot|N|)$ messages in the worst case and shows that, for a natural class of algorithms (called "linear distributed algorithms"), there is an infinite set of instances with $|P| = O(|N|)$ that require $\Omega(|N|^2)$ messages.

The game-theoretic properties of these mechanisms have also been studied. The MC mechanism is known to be strategyproof but is vulnerable to groups of players colluding to improve their welfare. Previous studies did not investigate the nature of collusion needed to succeed in manipulating the mechanism. The SH mechanism, on the other hand, has been shown to be group strategyproof [MS97, M99].

In this paper, we extend previous results on the SH and MC mechanisms in two directions:

- We present a group-strategyproof mechanism that exhibits a tradeoff between the properties of SH: It can be computed by a distributed algorithm that uses significantly fewer messages than the natural SH algorithm (exponentially fewer in the worst case), but it might fail to achieve exact budget balance or exact minimum efficiency loss (albeit by a bounded amount).
- We completely characterize the groups that can strategize successfully against the MC mechanism.

The rest of this paper is organized as follows. In Section 2, we present our group-strategyproof, communication-efficient mechanism and explain why it can be viewed as a step toward the goal of "approximately computing the SH mechanism" in a communication-efficient manner. In Section 3, we present our result on successful collusion against the MC mechanism.

2 Towards approximating the SH mechanism

In view of the evidence given in [FPS00] that exact computation of the SH mechanism requires an unacceptably large number of messages, it is natural to ask the following question: Can one compute an *approximation* to the SH mechanism using an algorithm that sends significantly fewer messages? To approach this question, we must first say what it means to "approximate the SH mechanism" and specify exactly what we mean by a "message."

A multicast cost-sharing mechanism is a pair of functions (σ, x). Thus, one may be tempted to define an approximation of the mechanism as a pair of functions (σ', x') such that σ' approximates σ well (for each u, these are characteristic vectors of subsets of P; so, we may call σ' a good approximation to σ if, for each u, the Hamming distance between the vectors is small), and x' approximates x well (in the sense, say, that the L^p-difference of $x(u)$ and $x'(u)$ is small, for each u, for some p). The mechanism (σ', x'), however, would not be interesting if its game-theoretic properties were completely different from those of (σ, x). In particular, if (σ', x') were not strategyproof, then agents might misreport their utilities; thus, even if (σ, x) and (σ', x') were, for each u, approximately equal as pairs of functions, the resulting equilibria might be very different, *i.e.*, $(\sigma'(v), x'(v))$ might be very far from $(\sigma(v), x(v))$, where v is the reported utility vector when using the approximate mechanism (σ', x'). Thus, we require that our approximate mechanisms retain the strategic properties – strategyproof or group strategyproof – of the mechanism that they are approximating.[4] In addition, if the original mechanism has some property, such as budget balance or efficiency, that does not relate to the underlying strategic behavior of agents but is an important design goal of the mechanism, then we would want the approximate mechanism to approximate that property closely.

Thus, we must define "approximation of a multicast cost-sharing mechanism" not in terms of the closeness of (σ', x') to (σ, x) but in terms of the desired game-theoretic properties properties.

The SH mechanism is group-strategyproof, budget-balanced, and, among all mechanisms with these two properties, the unique one that minimizes the worst-case efficiency loss. We should therefore strive for a group-strategyproof mechanism that has low network complexity and is approximately budget-balanced and approximately efficiency-loss minimizing in the worst case. "Approximately budget-balanced" can be taken to mean that there is a constant $\beta > 1$ such that, for all $c(\cdot)$, $T(P)$, and u:

$$(1/\beta) \cdot C(T(R(u))) \leq \textstyle\sum_{i \in R(u)} x_i(u) \leq \beta \cdot C(T(R(u)))$$

Similarly, "approximately efficiency-loss minimizing" can be taken to mean that there is a constant $\gamma > 1$ such that, for all $c(\cdot)$ and $T(P)$, the worst-case efficiency loss is at most γ times the worst-case efficiency loss suffered by SH. We do not obtain such a mechanism here, but we do make some progress toward the goal; our mechanism is group-strategyproof and fails to achieve exact budget balance and exact minimum efficiency loss by bounded amounts, but the bounds are not constant factors. Furthermore, there is a natural distributed algorithm that computes this mechanism using far fewer messages than appear to be needed for SH computation.

Precise definition of "message":
Throughout this section, we use the term *message* to mean $O(1)$ numerical values, each of size at most $poly(m, \log |P|, \log |N|)$, where m is the maximum size of a numerical input, *i.e.*, a utility value u_i or a link cost $c(l)$. (The size of a value is simply the number of bits needed to write it down.) Recall that this is an acceptable maximum message size, according to the criteria put forth in [FPS00]. Our algorithms will require network nodes to send functions to other network nodes, and these functions will be specified either by their values at $O(|P|)$ points or by their values at $O(\log |P|)$ points. When a node sends such a specification, it adds $O(|P|)$ messages (resp., $O(\log |P|)$ messages) to the network complexity of the algorithm. This use of the term "message" is consistent with its use in [FPS00]. Note that the network complexity is the same for

[4] One could relax this requirement to consider mechanisms in which the strategic behavior required to manipulate the mechanism is "hard to compute" [NR00] or for which the effects of strategic manipulation are completely characterizable and deemed to be tolerable. We have not pursued either of these lines of inquiry in this section. We turn to the question of characterization of strategic manipulations in Section 3 below.

an algorithm that sends a set of numerical values simultaneously as it is for one that sends each value in the set separately; this is necessary if the notion of "network complexity" is to require that both the number of messages and the size of messages be modest.

Previous work on approximation in algorithmic mechanism design:
Nisan and Ronen [NR00] were the first to address the question of approximate computation in algorithmic mechanism design. They considered Vickrey-Clark-Groves (VCG) mechanisms in which optimal outcomes are NP-hard to compute (*e.g.*, combinatorial auctions). They pointed out that, if an optimal outcome is replaced by a computationally tractable approximate outcome, the resulting mechanism may no longer be strategyproof. The above discussion of how we should define "approximating the SH mechanism" and why approximating the pair of functions (σ, x) is not sufficient is based on the analogous observation in our context. [NR00] approaches this problem by developing a notion of "feasible" strategyproofness and describing a broad class of situations in which NP-hard VCG mechanisms have feasibly strategyproof approximations. This approach is not applicable to SH-mechanism approximation for several reasons: SH is not a VCG mechanism; we are not seeking an approximation to an NP-hard optimization problem but rather a communication-efficient approximation to an apparently communication-inefficient, but polynomial-time computable, function; we are interested in network complexity in a distributed computational model, and [NR00] is interested in time complexity in a centralized computational model. Approximate multicast cost-sharing was first addressed by Jain and Vazirani [JV01]. They exhibited a group-strategyproof, approximately budget-balanced,[5] polynomial-time mechanism based on a 2-approximation algorithm for the minimum Steiner-tree problem. Their approach is also not applicable to SH-mechanism approximation, because they are concerned with time complexity in a centralized computational model, their network is a general directed graph (rather than a multicast tree, as it is in our case), and they are not attempting to approximate minimum worst-case efficiency loss. Finally, "competitive-ratio" analysis (a form of approximation) has been studied for a variety of strategyproof auctions [GH00, GHW01, LN00]. In Section 2.1, we review the natural SH algorithm given in [FPS00]. In Section 2.2, we give an alternative SH algorithm that also has unacceptable network complexity but that leads naturally to our approach to approximation. In Sections 2.3, 2.4, and 2.5, we define a new mechanism that has low network complexity, prove that it is group-strategyproof, and obtain bounds on the budget deficit and the efficiency loss.

2.1 The natural multi-pass SH algorithm

The Shapley value mechanism divides the cost of a link l equally between all receivers downstream of l. The mechanism can be characterized by its cost-sharing function $f : 2^P \mapsto \Re^P_{\geq 0}$ ([MS97, M99]). For a receiver set $R \subseteq P$, player i's cost share is $f_i(R)$.

[FPS00] presents a natural, iterative algorithm to compute $\sigma_i(u)$ and $x_i(u)$ for the SH mechanism, using this cost sharing function f. We restate it here:

First, set $\sigma_i^{(1)}(u) = 1$ for all $i \in P$. In the kth iteration, start with a set of receivers $R^{(k)}(u) = \{i | \sigma_i^{(k)} = 1\}$. Then, perform one pass up and down the tree. In the upward pass, compute for each link l the number of players in $R^{(k)}(u)$ downstream of l. In the downward pass, compute the cost shares $x_i^{(k)}(u) = f_i(R^{(k)}(u))$ by adding up the cost shares for each link in the path from each receiver to the root. Then, drop any player in $R^{(k)}(u)$ who cannot afford his cost share, *i.e.*, set $\sigma_i^{(k+1)}(u) = 1$ iff $\sigma_i^{(k)}(u) = 1$ and $x_i^{(k)} \leq u_i$. This process converges in a finite number of iterations, and the resulting values of $x(u)$ and $\sigma(u)$ define the mechanism.

Unfortunately, this algorithm could make as many as $\Omega(|N|)$ passes up and down the tree and send a total of $\Omega(|N| \cdot |P|)$ messages in the worst case. Moreover, [FPS00] contains a corresponding lower bound for a broad family of algorithms: There is an infinite class of inputs, with $|P| = \mathcal{O}(|N|)$, for which any "linear distributed algorithm" that computes SH sends $\Omega(|N|^2)$ messages in the worst case.

[5] The [JV01] definition of approximate budget balance is more stringent than the one we suggest in this section; it does not allow a budget deficit (and also requires, as ours does, a constant-factor bound on the budget surplus).

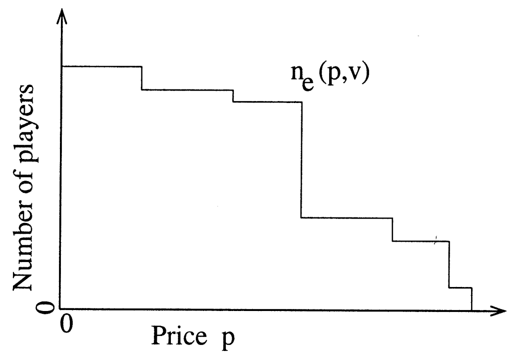

Figure 1: The function $n_e(p, v)$ computed for each edge e

2.2 A one-pass SH algorithm

Our first step toward a more communication-efficient mechanism that has some of the desirable properties of SH is to present a distributed algorithm for SH that makes just one pass up and down the tree. This algorithm still sends $\Omega(|P|)$ messages (simultaneously) across each link in the worst case; so the total number of messages is still $\Omega(|N| \cdot |P|)$, and thus the mechanism is not directly usable. However, we show in Section 2.3 how approximating the functions communicated in this one-pass SH algorithm leads to a new mechanism that can be computed by sending $O(|N| \cdot \log |P|)$ messages total and has other desirable properties.

Let v be the (reported) utility profile. Then, for every edge e in the tree, we compute the following function:

$\mathbf{n}_e(p, v) \stackrel{def}{=}$ *the number of players in the subtree beneath e who are each willing to pay p for the links* above e (*i.e.*, the number of players in this subtree who will not drop out of the receiving set when their cost share for the link edges from the root down to but excluding e is p) .

(We put the utility profile v in explicitly as an argument to allow us to prove group-strategyproofness later; however, in any one run of the algorithm, v is fixed.)

Note that this definition requires that the cost from the leaves through e has already been adjusted for. A single definition is sufficient for this purpose, because the SH mechanism does not distinguish between receivers downstream of e when sharing the cost of e or its ancestors; all such receivers pay the same amount for these links. For each edge, we compute this function at all prices p. The function $n_e(p, v)$ is monotonically decreasing with p, and, for any given utility profile v, can be represented with at most $|P|$ points with coordinates (p_i, n_i) corresponding to the "corners" in the graph of $n_e(p, v)$ in Figure 1. We use this *list-of-points* representation of $n_e(p, v)$ in our algorithm.

The statement of the multicast mechanism problem allows for players at intermediate (non-leaf) nodes; however, to simplify the discussion, we can treat each of these players as if it were a child node with one player and parent edge-cost zero. Thus, we assume, without loss of generality, that all players are at leaf nodes only.

The function $n_e(p, v)$ is computed at the node α_e below e in the tree. The computation is easy if α_e is a leaf node, because one can sort the utilities of the players at α_e, divide the cost C_e between the highest $1, 2, \cdots k$ players at this node, and compute the unspent utility in each case.

If α_e is not a leaf node, we have to include the functions reported by child nodes in this calculation. Suppose we are at node α_e and have received the functions $n_{e_i}(p, v)$ from all the child edges $E = \{e_1, e_2, \cdots e_r\}$ of e. We can compute $n_e(p, v)$ in two steps:

- **Step 1**: First, we compute a function

$$m_e(p,v) = \sum_{i=1}^{r} n_{e_i}(p,v)$$

Intuitively, $m_e(p,v)$ is the number of players beneath e who are willing to pay p each towards the cost from the root down to (and *including*) e. This is apparent from the definition of $n_{e_i}(p,v)$. If each $n_{e_i}(.)$ is specified as a sorted list of points, we can compute $m_e(.)$ by merging the lists and adding up the numbers of players.

- **Step 2**: Now, we have to account for the cost C_e of the link e to compute the function $n_e(p,v)$. For any p such that $pm_e(p,v) \geq C_e$, we have

$$n_e(p - \frac{C_e}{m_e(p,v)}, v) \;\geq\; m_e(p,v)\;, \tag{1}$$

because the $m_e(p,v)$ players who were willing to pay p for the path including e can share the cost of e. Equation 1 need not be a strict equality because it is possible that, for a price $q < p$, the larger set of size $m_e(q,v)$ has

$$q - \frac{C_e}{m_e(q,v)} \geq p - \frac{C_e}{m_e(p,v)}$$

and so could also support the price $p' = p - (C_e/m_e(p,v))$ each for the links above e. However, the value of $n_e(p,v)$ must correspond to $m_e(p',v)$ for some $p' > p$, because every player beneath e who receives the transmission pays an equal amount for the edge e. It follows that

$$n_e(p,v) = \max_{\left\{p' - \frac{C_e}{m_e(p',v)} \geq p\right\}} m_e(p',v) \tag{2}$$

When the RHS of Equation 2 is undefined, $n_e(p,v) = 0$. Given a list of points $(p^{(i)}, m^{(i)})$ corresponding to $m_e(.)$, we can compute $n_e(.)$ through the following procedure: For each point $(p^{(i)}, m^{(i)})$, we get the transformed point $(p^{(i)} - (C_e/m^{(i)}), m^{(i)})$. We then sort the list of these transformed points and throw away any point that is dominated by a higher m_i at the same or higher price.

In this manner, we can inductively compute $n_e(.)$ for all edges, until we reach the root. At the root, we can combine the functions received from the root's children to get $m_{root}(.)$ Because there are no further costs to be shared, it follows that there are $m = m_{root}(0,v)$ players who are willing to share the costs up to the root. Also, there is no set of more than m players that can support the cost up to the root, and so m is the size of the unique greatest fixed-point set computed by the Shapley value mechanism.

Now, we have to compute the prices charged to each player. Assuming that the nodes have stored the functions $n_e(.)$ on the way up the tree, we compute the prices on the way down as follows: For each edge e, we let x_e be the cost share of any receiver below e for the path down to (but not including) e. Then, $x_{root} = 0$ and, if e has child edges $e_1, e_2, \cdots e_k$,

$$x_{e_i} = x_e + \frac{C_e}{n_e(x_e,v)} \tag{3}$$

We descend the tree in this manner until we get a price x_i for every player i. Then, we include i in $R(v)$ iff $x_i \leq v_i$, and if included i pays x_i.

The following two lemmas show that this one-pass algorithm computes the SH mechanism.

Lemma 1 *The outcome computed by this algorithm is budget-balanced.*

Proof: By definition, there are exactly $n_e(x_e, v)$ players beneath e who can pay x_e for the path down to but excluding e. It follows that

$$\sum_i n_{e_i}(x_i, v) = m_e(x_i, v) = n_e(x_e, v)$$

Using this inductively until we reach the leaves, we can show that there are $n_e(x_e, v)$ players downstream of e in the receiving set chosen by the algorithm, *i.e.*, with $x_i \leq v_i$. Equation 3 then shows that the cost of each edge is exactly balanced, and hence the overall mechanism is budget balanced. □

Lemma 2 *The receiving set computed by this algorithm is the same as the receiving set computed by the iterative Shapley value mechanism.*

Proof: By *Lemma 1* we know that the set $R(v)$ constructed can bear the cost of the transmitting to $R(v)$. Let $\overline{R}(v)$ be the receiving set chosen by the iterative Shapley value mechanism. Because $\overline{R}(v)$ is the greatest fixed point, $\overline{R}(v) \supseteq R(v)$.

We show that $\overline{R}(v) = R(v)$ as follows: let $\overline{x}_e(v)$ be the cost shares of individual receivers for the path down to but excluding e corresponding to the receiver set $\overline{R}(v)$. Let $\overline{n}_e(v)$ be the number of receivers below e in this outcome. By induction, we can show that Steps 1 and 2 of the algorithm described in this section maintain the property

$$n_e(\overline{x}_e(v), v) \geq \overline{n}_e(v)$$

Because this is true at the root, it follows that $|R(v)| \geq |\overline{R}(v)|$. Hence $R(v) = \overline{R}(v)$. □

The two algorithms (one-pass and iterative) are both budget-balanced, with the same receiver set and the same cost-sharing function; thus they both compute the SH mechanism.

2.3 A communication-efficient approximation of $\mathbf{n_e}(\cdot)$

The algorithm for the Shapley value mechanism described in the previous section makes only one pass up and down the tree. However, in the worst case, the function $n_e(.)$ passed up edge e requires $|P|$ points (p_i, n_i) to represent it, which is undesirable.

We can achieve an exponential saving in the worst-case number of messages by passing a small **approximate representation of** $n_e(.)$ instead. What should this approximation look like? Firstly, we would like to **underestimate** $n_e(.)$ at every point so that we can still compute a feasible receiving set in one pass.

For each edge e, instead of $n_e(p, v)$, the mechanism uses an under-approximation $\tilde{n}_e(p, v)$. The approximation we choose is simple and is illustrated in *Figure 2*. For some parameter $\alpha > 1$, we round down all values of $n(p, v)$ to the closest power of α. The resulting function $\tilde{n}_e(p, v)$ has at most $(\log|P|/\log\alpha)$ "corners," and so it can be represented by a list of $\mathcal{O}(\log|P|)$ points.

We compute $\tilde{n}_e(p, v)$ by using the following modified versions of Steps 1 and 2 of the one-pass algorithm:

- **Step 1'**: Compute

$$\hat{m}_e(p, v) = \sum_{e_i} \tilde{n}_{e_i}(p, v)$$

(This step is unchanged - we do an exact summation - but the input functions are approximate.)

- **Step 2'**: First, adjust for cost C_e as before

$$\hat{n}_e(p, v) = \max_{\left\{p' - \frac{C_e}{\hat{m}_e(p', v)} \geq p\right\}} \hat{m}_e(p', v)$$

Then, approximate the function $\hat{n}_e(.)$ by $\tilde{n}_e(.)$:

$$\tilde{n}_e(p, v) = \alpha^{\lfloor \log_\alpha \hat{n}_e(p, v) \rfloor}$$

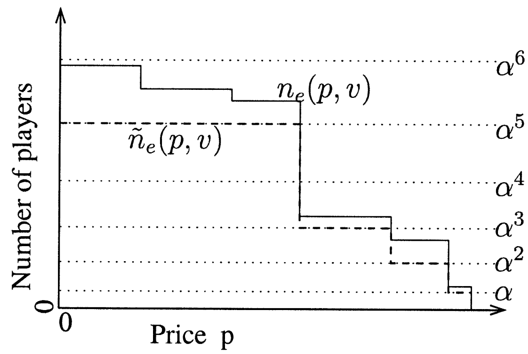

Figure 2: Approximation to $n_e(p, v)$.

With the list-of-points representation, this is easily done by dropping elements of the list that do not change $\tilde{n}_e(p, v)$.

On the way down, we compute

$$x_{e_i} = x_e + \frac{C_e}{\tilde{n}_e(x_e, v)}$$

We now have a situation in which the number of receivers i downstream of edge e is potentially greater than $\tilde{n}_e(x_e, v)$, because $\tilde{n}_e(.)$ is an under-approximation. Define a mechanism (called Mechanism SF, for "step function") by including all the potential receivers and making receiver i pay x_i as before. Note that mechanism SF does not achieve exact budget balance - there may be a budget surplus.

2.4 Group strategyproofness of mechanism SF

Notation

Throughout this section, we use $u = (u_1, u_2, \cdots, u_n)$ to indicate the true utility profile of the players. Recall that $v|^i r_i$ denotes the utility profile $(v_1, v_2 \cdots, v_{i-1}, r_i, v_{i+1}, \cdots, v_n)$, *i.e.*, the utility vector v perturbed by replacing v_i by r_i.

Now, let v be the reported utility profile. Then $S = \{i \mid u_i \neq v_i\}$ is the strategizing group. This strategy is **successful** if no member of S has a lower welfare as a result of the strategy, and at least one member has a higher welfare as a result of the strategy:

$$\forall i \in S \;\; w_i(v) \geq w_i(v)$$

$$\exists j \in S \text{ such that } w_j(v) > w_j(v)$$

We prove that mechanism SF is group strategyproof in three steps: First, we prove that, if there is a successful (individual or group) strategy, there is a successful strategy v in which all colluding players *raise* their utility *i.e.* $v_i \geq u_i$. This is intuitive, because if a player receives the transmission, she is not hurt by raising her utility further. Next, we show that a receiver has no strategic value in raising her utility: if $x_i \leq u_i < v_i$, then the outcome of the mechanism is unchanged in moving from strategy v to $v|^i u_i$. Finally, we combine these two results to show that a successful strategy against mechanism SF cannot exist.[6]

For the first part, we formalize our argument that it is sufficient to consider strategies in which all members raise their utilities. The key to this is showing that the following monotonicity property holds:

[6] We also have an alternate proof of this result that first shows that mechanism SF can be described in terms of an underlying cross-monotonic cost-sharing function and then relies on the result in [M99] that all such cost-sharing mechanisms are group-strategyproof. For space reasons, we give only the more direct proof here.

Lemma 3 Monotonicity: *Let u be a utility profile and v be the perturbed profile obtained by increasing one element of u ($v = u|^i v_i$, where $v_i > u_i$). Then, the following properties hold:*

(i). $\forall e, x \ \ \tilde{n}_e(x, v) \geq \tilde{n}_e(x, u)$

(ii). $\forall j \in P \ \ x_j(v) \leq x_j(u)$

(iii). $\tilde{R}(v) \supseteq \tilde{R}(u)$

(Here $x_j(v)$ is the ask price *computed for player j in the downward pass.)*

Proof: Note that our approximation technique has the property that, if $\hat{n}_e(x, v) \geq \hat{n}_e(x, u)$, then $\tilde{n}_e(x, v) \geq \tilde{n}_e(x, u)$. Statement *(i)* is then immediately true at the leaves and follows by induction at non-leaf nodes. Because the cost of any link e is divided among $\hat{n}_e(x_e, v)$ players, statement *(ii)* follows from statement *(i)*. Finally, because the utilities are the same (or higher in the case of player j), statement *(ii)* implies statement *(iii)*. □

Lemma 3 suggests that, for any successful strategy v, we can get a successful strategy v' by raising v_i to u_i whenever $v_i < u_i$. However, we first have the technical detail of eliminating non-receivers from the strategizing group:

Lemma 4 *Let v be a strategy for group S. Suppose $i \in S$ and $i \notin \tilde{R}(v)$. Let v' be the strategy $v|^i u_i$. Then, $x_j(v') \geq x_j(v)$ for all $j \in P$.*

Proof: Since $i \notin \tilde{R}(v)$, $x_i(v) > v_i > u_i$. (When $v_i < u_i$, the statement follows directly from Lemma 3.) We can show inductively that $\tilde{n}_e(x_e(v), v') = \tilde{n}_e(x_e(v), v)$ and the statement follows. □

Combining the last two results, we get our result:

Lemma 5 *Suppose a group S has a successful strategy v. Further, assume that all members of S receive the transmission with strategy v. Then, S has a successful strategy v' where $v'_i \geq u_i$.*

Proof: By lemma 4, we can assume that all members of S receive the transmission with strategy v. Define a sequence of strategies

$$v = v^{(0)}, v^{(1)}, \cdots v^{(n-1)}, v^{(n)} = v'$$

where $v^{(k)} = v^{(k-1)}|^k u_k$ if $u_k > v_k$, $v^{(k)} = v^{(k-1)}$ otherwise. The monotonicity property implies that if $v^{(k-1)}$ is a successful strategy, so is $v^{(k)}$. □

Now, we prove that, if a receiver i raises his utility, the solution is not altered:

Lemma 6 *Let u be a utility profile and let v be the perturbed profile obtained by increasing one element of u ($v = u|^i v_i$, where $v_i > u_i$). If $u_i \geq x_i(u)$, then*

$$\forall e, \forall x < x_e(u) \ \ \tilde{n}_e(x, v) = \tilde{n}_e(x, u)$$

Proof: It is obviously true at the leaves, because the utility v_i only affects the value of $\tilde{n}_{leaf}(.)$ at prices above $u_i \geq x_i$. (This is a result of our *pointwise* approximation scheme; not all approximations would have this property.) Also, because of the monotonically decreasing nature of $\tilde{n}_e(.)$, this property is maintained by Steps 1' and 2' as we move up the tree. □

A corollary of lemma 6 is that, when the conditions of the lemma hold, the output of the mechanism is identical for inputs u and v. This follows from the fact that $\tilde{n}_e(.)$ is not evaluated at prices above $x_e(u)$ on the way down, and so inductively $x_e(v) = x_e(u)$ for all edges e. Hence, each player gets the same ask price $x_i(v) = x_i(u)$.

We can now prove the main result:

Theorem 1 *Mechanism SF is group strategyproof.*

Proof: Assume the opposite, *i.e.*, that there is a successful group strategy against mechanism SF. Then, by lemma 5, there is a group strategy v for some set S, where every member of S receives the transmission after the strategy. Define the sequence of strategies:

$$v = v^{(0)}, v^{(1)}, \cdots v^{(n-1)}, v^{(n)} = u$$

where $v^{(k)} = v^{(k-1)}|^k u_k$. It follows from lemma 6 that if $v^{(k-1)}$ is a successful strategy for S, so is $v^{(k)}$. This implies that u is a successful strategy, which is a contradiction. □

2.5 Bounds on budget deficit and efficiency loss

In this section, we present a simple modification of mechanism SF, called SSF (for "scaled SF"), and prove bounds on its budget deficit and loss of net worth with respect to the SH mechanism. This mechanism works as follows:

Mechanism SSF:
Let h_e be the height of link e in the tree. Then, define the *scaled cost* $C^\alpha(e)$ of the link e to be $C(e)/(\alpha^{h_e})$. Run mechanism SF assuming link costs $C^\alpha(e)$ instead of $C(e)$, to compute a receiver set $R^\alpha(u)$ and cost shares $x_i^\alpha(u)$.

Lemma 7 *Mechanism SSF is group strategyproof.*

Proof: The player's utility does not affect the scaled costs, and mechanism SF is group-strategyproof for any tree costs. □

Let $R(u)$ be the receiver set in the (exact) Shapley value mechanism. We now show that $R^\alpha(u) \supseteq R(u)$.

Lemma 8 *Let $\tilde{n}_e^\alpha(x, u)$ be the surplus utility distribution computed by mechanism SSF. Let $n_e(x, u)$ and x_e be defined as in the exact Shapley value algorithm. Then,*

$$\forall e \quad \tilde{n}_e^\alpha(x_e, u) \geq \frac{n_e(x_e, u)}{\alpha^{h_e}}$$

Proof: We prove the statement by induction on h_e. For $h_e = 1$ (a leaf edge), it is true because of our approximation method. Suppose the statement is true for all edges of height no more than r, and $h_e = r+1$. Let $\{e_1, e_2, ..e_k\}$ be the child edges of e. By the inductive assumption, $\tilde{n}_{e_i}^\alpha(x, u) \geq (n_{e_i}(x_{e_i}, u)/\alpha^r$. It follows that

$$\hat{m}_e^\alpha(x_{e_i}, u) = \sum_{i=1}^{k} \tilde{n}_{e_i}^\alpha(x_{e_i}, u) \tag{4}$$

$$\geq \frac{n_e(x_e, u)}{\alpha^r} \tag{5}$$

From the computation of the ask prices x_e and x_{e_i}, we know

$$x_e = x_{e_i} + \frac{C_e}{n_e(x_e, u)}$$

Let

$$x' = x_{e_i} - \frac{C^\alpha(e)}{\hat{m}_e^\alpha(x_{e_i}, u)}$$

Then, $x' \geq x_e$ follows from 5. Also, following *Step 2'* of mechanism SF, we have

$$\hat{n}_e^\alpha(x', u) \quad \geq \quad \hat{m}_e^\alpha(x_{e_i}, u)$$

and because $x' \geq x_e$, $\hat{n}_e^\alpha(x_e, u) \geq \hat{n}_e^\alpha(x', u)$. Finally, in passing from $\hat{n}^\alpha(.)$ to $\tilde{n}^\alpha(.)$, we get

$$\begin{aligned} \tilde{n}_e^\alpha(x_e, u) &\geq \frac{\hat{n}_e^\alpha(x_e, u)}{\alpha} \\ \tilde{n}_e^\alpha(x_e, u) &\geq \frac{n_e(x_e, u)}{\alpha^{r+1}} \end{aligned}$$

And thus the statement is proved by induction. □

Lemma 9 $R^\alpha(u) \supseteq R(u)$.

Proof: Using Lemma 8,

$$\frac{C^\alpha(e)}{\tilde{n}_e^\alpha(x_e, u)} \leq \frac{C(e)}{n_e(x_e, u)}$$

and we can show inductively that $x_e^\alpha \leq x_e$ for all links e. Because this is true at the leaves, it follows that $R^\alpha(u) \supseteq R(u)$ □

Bounding the budget deficit:. Unlike mechanism SF, which is balanced or runs a surplus, mechanism SSF may generate a budget deficit (but never a surplus). However, the deficit (as a fraction of the cost) can be bounded in terms of α and the height h of the tree:

Theorem 2

$$\frac{C(T(R^\alpha(u)))}{\alpha^h} \leq \sum_{i \in R^\alpha(u)} x_i^\alpha(u) \leq C(T(R^\alpha(u)))$$

Proof: Let $X = \sum_{i \in R^\alpha(u)} x_i^\alpha(u)$. Because mechanism SF never runs a deficit, $X \geq C^\alpha(T(R^\alpha(u))) \geq \frac{C(T(R^\alpha(u)))}{\alpha^h}$.

The right-hand side inequality can be proved by induction on the tree height. □

Bounding the worst-case efficiency loss:. Let T^α and T be the multicast trees corresponding to the receiver sets $R^\alpha(u)$ and $R(u)$ respectively. Then, T^α can be written as a disjoint union of trees, $T^\alpha = T \cup T_1 \cup T_2 \cup \cdots \cup T_r$. The corresponding relation for the receiver set is $R^\alpha(u) = R(u) \cup R_1 \cup R_2 \cup \cdots \cup R_r$, where R_i is the subset of players in $R^\alpha(u)$ who are attached to some node in T_i. Some of these subtrees may have negative efficiency, and so the overall efficiency of the SSF mechanism may be less than the efficiency of the Shapley value. However, we can bound the worst-case efficiency loss (with respect to the exact Shapley value) in terms of the total utility $U = \sum_{i \in P} u_i$:

Theorem 3

$$NW(R^\alpha(u)) \geq NW(R(u)) - (\alpha^h - 1)U$$

Proof: The efficiency of the receiver set $R^\alpha(u)$ is

$$\begin{aligned} NW(R^\alpha(u)) &= \sum_{i \in R^\alpha(u)} u_i - C(T(R^\alpha(u))) \\ &= NW(R(u)) + \sum_{j=1}^{r} NW(R_j) \end{aligned}$$

Now, for any subtree T_j of T^α,

$$U(T_j) = \sum_{i \in T_j} u_i \geq C^\alpha(T_j) \geq \frac{C(T_j)}{\alpha^h} \implies NW(T_j) \geq -(\alpha^h - 1)U(T_j)$$

and hence

$$\begin{aligned} NW(R^\alpha(u)) &\geq NW(R(u)) - (\alpha^h - 1)\sum_{j=1}^{r} U(T_j) \\ NW(R^\alpha(u)) &\geq NW(R(u)) - (\alpha^h - 1)U(R^\alpha(u)) \\ &\geq NW(R(u)) - (\alpha^h - 1)U \end{aligned}$$

□

To summarize, Mechanism SSF uses $O(\log_\alpha n)$ messages per link, incurs a cost of at most α^h times the revenue collected, and has an efficiency loss of at most $(\alpha^h - 1)U$ with respect to the SH mechanism.

For example, when $|P| = 100,000$ and $h = 5$, the best algorithm known for the SH mechanism would require about 100,000 messages per link in the worst case. Setting $\alpha = 1.03$, mechanism SSF requires fewer than 400 messages per link, has a budget deficit of at most 14% of the tree cost, and has a worst-case efficiency loss of at most 16% of the total utility. As another example, when $|P| = 10^6$ and $h = 10$, we can use $\alpha = 1.02$ to achieve a worst-case deficit of 19%, and worst-case efficiency loss of 22% of the total utility, with 700 messages per link or use $\alpha = 1.04$ to obtain corresponding bounds of 33% and 48% with about 350 messages per link.

3 Group strategies that succeed against the MC mechanism

[FPS00] gives a low-network-complexity algorithm for the Marginal-Cost mechanism for multicast cost sharing. The algorithm itself highlights interesting features of the mechanism, which we describe here:

Given an input utility profile u, the receiving set is the unique *maximal efficient set* of nodes. To compute this set, for each node $n \in N$, we recursively compute its *welfare* $W(n)$ as

$$W(n) = \left(\sum_{\substack{c \in C(n) \\ W(c) >= 0}} W(c) \right) - C_e$$

where $C(n)$ is the set of children of n in the tree and C_e is the cost of the edge linking n to its parent node. Then, the maximal efficient set $R(u)$ is the set of all players i such that every node on the path from i to the root has nonnegative welfare.

Another way to view this is as follows: The algorithm partitions the node set N into a forest $F(U) = \{T_1(u), T_2(u), \cdots, T_k(u)\}$. An edge from the original tree is included in the forest iff the child node has nonnegative welfare. This is illustrated in Figure 3. $R(u)$ is then the set of players at nodes in the subtree $T_1(u)$ containing the root.

Once $F(u)$ has been computed, for each player i, define $X(i, u)$ to be the node with minimum welfare value in the path from i to its root in its partition $T_l(u)$. Then, the payment $x_i(u)$ of each player i is defined as

$$\begin{aligned} x_i(u) = &\quad \max(0, u_i - W(X(i,u))) &\quad \forall i \in R(u) \\ x_i(u) = &\quad 0 &\quad \forall i \notin R(u) \end{aligned}$$

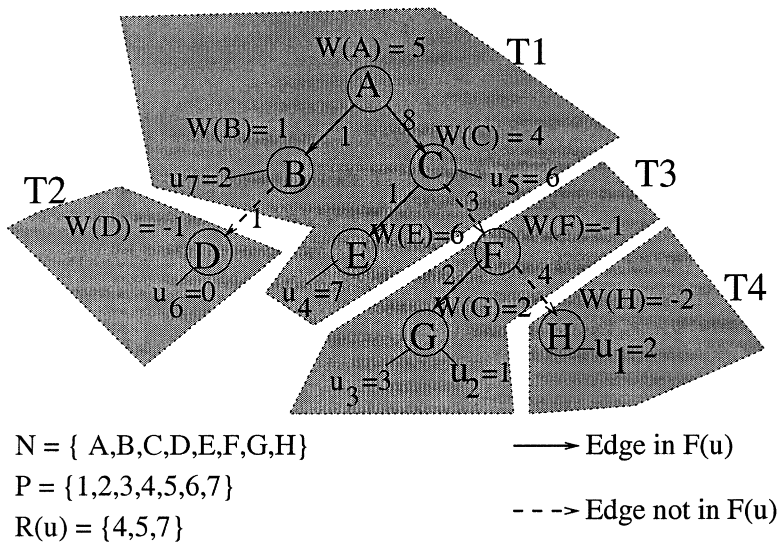

Figure 3: Forest induced by MC mechanism

If there are multiple nodes on the path with the same welfare value, we choose $X(i,u)$ to be the one among them nearest to i; this does not alter the payment, but it simplifies our later results on when a coalition can be successful. We will use this characterization of the receiving set and payments in terms of $F(u)$ and $X(i,u)$ in our analysis of group strategies against the MC mechanism.

Recall that a strategy v for a group S is a successful group strategy at a given utility profile u if $\forall i \in S\ w_i(v) \geq w_i(u)$, and $w_j(v) > w_j(u)$ for some $j \in S$. In other words, a successful group strategy is one that benefits at least one member of the coalition and harms none of the members of the coalition. If the group S has only two members, we call it a **successful pair strategy**.

It is well known that the MC mechanism is not group strategyproof for the multicast cost-sharing problem. However, it isn't clear in general which forms of collusion would result in successful manipulation. Here we examine this in detail by asking two questions. First, for which utility profiles is MC group strategyproof? Second, for a utility profile u where MC is not GSP, what do the successful strategies look like?

These questions suggest a general line of inquiry within algorithmic mechanism design that is worthy of further study. Recall that, in our discussion in Section 2 of what it means to "approximate the SH mechanism," we insisted that an approximate mechanism be group-strategyproof. We noted, however, that some form of "approximate group-strategyproofness" might be acceptable. That is, one may be quite willing to deploy a mechanism that is known not to be group-strategyproof if the groups that could strategize successfully and their effects on the other parties and resources involved were precisely characterizable and deemed to be acceptable. For example, in multicast cost sharing, a multicast-service provider may be willing to use such a mechanism if successful groups did not cut deeply into his profits. Our results on the MC mechanism cannot be put to practical use in this way, but they exemplify a type of characterization that, for other mechanisms, may be usable in practice.

Preliminaries. To start with, we restate two lemmas from [MS97] that we use repeatedly.

Let u be the true utility profile, and let $u' = u|^i u_i'$ *i.e.* the ith player reports a different utility, everyone else is truthful. If $u_i' > u_i$, then

Lemma 10 *(no other player is hurt)*

$$w_j(u') \geq w_j(u) \quad \forall j \neq i$$

Further, if player i would have received the service with his truthful utility ($i \in R(u)$), then

Lemma 11 *(player i is not hurt by raising his utility)*

$$i \in R(u) \implies x_i(u') = x_i(u)$$

From these results, we can prove that we only need to consider strategies in which players *increase* their reported utility:

Lemma 12 *Let u be the true utility profile, and let v be a reported utility profile. Define a strategy v' by*

$$v'_i = \max(v_i, u_i)$$

Then, for all $i \in P$, $w_i(v') \geq w_i(v)$.

Proof: We can increase the elements of v one at a time, and at each stage use Lemma 10 and the strategyproofness of the mechanism to show that no player's utility is reduced. □

The definition of a successful strategy under GSP allows some members of the coalition not to receive the broadcast. To avoid complicating the later proofs, we prove that setting the utility of any zero-welfare player to 0 does not reduce the welfare of any other player.

Lemma 13 *Let u be the true utility profile. Let v be the reported utility profile, and require that $\forall i \; v_i \geq u_i$. Also assume that no player has negative welfare with this strategy: $w_i(v) \geq 0$. Let $j \in P$ be such that $j \notin R(v)$ or $x_j(v) = v_j$. Then, if we construct a utility profile $v' = v|^j 0$ by setting utility of j to 0,*

$$w_i(v') = w_i(v) \quad \forall i \in P$$

Proof: Note that $W(P, v') = W(P - j, v)$. If $j \notin R(v)$, then $j \notin R(v')$. The definition of MC payment for player i is

$$x_i(v) = v_i - W(P, v) + W(P - i, v) \tag{6}$$

By submodularity of cost, which implies supermodularity of welfare surplus [MS97, page 13],

$$W(P, v) - W(P - j, v) \geq W(P - i, v) - W(P - j - i, v)$$

If $j \notin R(v)$, then the LHS is 0. If $j \in R(v)$, then

$$x_j(v) = v_j \implies W(P, v) - W(P - j, v) = 0$$

Therefore, in either case, the LHS of Equation 6 is 0 implying its RHS is also 0, so equality holds. Rearranging terms gives us

$$W(P, v) - W(P - i, v) = W(P - j, v) - W(P - j - i, v) \tag{7}$$

Consider three cases:

- **Case 1**: $i \in R(v)$ and $x_i(v) < v_i$
  In this case, Equation 7 implies that $i \in R(v')$ and $x_i(v) = x_i(v')$

- **Case 2**: $i \in R(v)$ and $x_i(v) = v_i$
  If $i \in R(v')$, then as in Case 1 $x_i(v) = x_i(v')$. If $i \notin R(v')$, then $w_i(v') = 0$. By assumption, $v_i \geq u_i$ and so $w_i(v) \leq 0$. But we also assumed that $w_i(v) \geq 0$, $w_i(v) = 0$.

- **Case 3**: $i \notin R(v)$
  Then, $i \notin R(v')$ and $w_i(v) = w_i(v') = 0$;

In all three cases, $w_i(v) = w_i(v')$. □

Suppose a group S has a successful strategy v. Through repeated application of Lemma 12 and Lemma 13, we can construct a successful strategy v' for a subset S' of S, such that every member of S' receives positive welfare:

Lemma 14 *For a true utility profile u, suppose coalition S has a successful strategy v. Then, there exists $S' \subseteq S$, such that S' has a successful strategy v' and $\forall i \in S'$ $w_i(v') > 0$.*

Proof: By Lemma 12, we can assume $v_i \geq u_i$, and hence the conditions of Lemma 13 hold. First, suppose that S had only one member j such that $w_j(v) = 0$. Then, setting $S' = S - \{j\}$, $v' = v|^j u_j$, we can use Lemma 13 and Lemma 10 to show that v' is a successful strategy for S', and no member of S' has zero welfare with this strategy.

In the general case, when there are multiple members of S with zero welfare, we can repeatedly apply this construction and show that the number of zero-welfare members of the coalition decreases with each iteration. □

This result allows us to restrict our attention to coalitions in which every member of the coalition has positive welfare (and hence receives the transmission) after the successful strategy.

Theorem 4 *Let u be the true utility profile, and let $P'(u) \subseteq P$ be the set of players who do not maximize their welfare at u: $P'(u) = \{i \in P \mid w_i(u) < u_i\}$. Then, the MC mechanism is GSP at utility profile u* iff *for every player $i \in P'$ the following condition is satisfied:*

> **There is no player j in the same component of $F(u)$ as i such that j is in the subtree rooted at $X(i, u)$.**

Proof: If part: Assume the conditions of Theorem 4 hold. Suppose S was a set of players with a successful strategy v. By Lemma 14, we can choose S and v such that every member of S has positive welfare with input v. We consider two cases:

- **Case 1:** $S \subseteq R(u)$
  Because v is a successful strategy, there exists a $j \in S$ for which $w_j(v) > w_j(u)$. This implies that $j \in P'(u)$ and so, by assumption, no other descendant of $X(j, u)$ participates in the group strategy. It follows that

  $$x_j(u) = u_j - W(X(j, u)) = x_j(v)$$

  which contradicts the claim that $w_j(v) > w_j(u)$.

- **Case 2:** $S \not\subseteq R(u)$
  In this case, consider the members of $S - R(u)$. They may be in different trees of $F(u)$; however, we can always find a member $i \in S - R(u)$ such that there is no other member of S in the subtree beneath $X(i, u)$. Now, $i \in R(v)$ requires $x_i(v) > u_i$, which contradicts the assumption that v is a successful strategy.

Only if part: Let S be a successful coalition with strategy v. Then, there exist $j \in S$ that strictly benefits from the strategy, *i.e.*, $w_j(v) > w_j(u)$. Now,

$$\begin{array}{rcl} x_j(u) & < & x_j(v) \\ u - W(X(j, u)) & > & v - W(X(j, v)) \end{array}$$

For this to happen, there must be another member of S in the subtree rooted at $X(j, u)$. We can again consider two cases $S \subseteq R(u)$ and $S \not\subseteq R(u)$ and show that in either case the conditions stated in the theorem must be violated. $\square$

Next, in the cases that the mechanism is not GSP, we can characterize the possible successful group strategies with the following results:

Theorem 5 *Consider a true utility profile u. Suppose coalition S has a successful strategy v. Then there exists a pair of players $i, j \in S$ such that i and j have a successful pair strategy v'.*

Furthermore, i and j are in the same set $T_l(u)$ in the forest, and j is in the subtree rooted at $X(i, u)$.

Proof: By lemma 14,we can assume that all members of S receive the transmission under v. Again, we consider two cases:

- **Case 1:** $S \subseteq R(u)$
  In this case, there is an $i \in S \cap R(u)$ such that $w_i(v) > w_i(u)$. Let A be the nearest ancestor node of i that is also an ancestor of at least one other member j of S. Now, the welfare of every node between i and A is at least $w_i(v)$ under profile v, and hence under u. Therefore, if j claims a utility $v'_j = v_j + L$, for large L, then $w_i(v') \geq w_i(v)$, for $v' = u|^j v_j$. Further, because j is also in $R(u)$, by Lemma 11 $w_j(v') = w_j(u)$, and so v' is a successful strategy for $\{i, j\}$.

- **Case 2:** $S \not\subseteq R(u)$
  In this case, as in Theorem 4, we can find a player $i \in S - R(u)$ such that all members of S that are descendants of $X(i, u)$ are also in the same tree of $F(u)$ as i. Choose a closest pair j, k in this subtree, and let A be their nearest common ancestor. Both j and k have positive welfare with strategy v. It follows that and all nodes in the paths from i to A and from j to A have positive welfare in u. Now, setting $v'_j = L$ and $v'_k = L$ for large L gives us a successful strategy for j and k.

$\square$

Theorem 6 *Any superset of a coalition that can violate GSP at a profile u can also violate GSP at a profile u*

Proof: Suppose $S \subseteq P$ has a strategy v that gives equal or higher welfare to all members of S. By lemma 14, we can assume that every member of S receives the transmission with this strategy. Now, let $v'_i = \max(v_i, u_i) \forall i \in P$. By repeated application of lemma 10,

$$w_i(v') \geq w_i(u) \forall i \notin S$$

Also, by using the fact that all elements of S have positive welfare and applying lemma 11 and lemma 10,

$$w_i(v') \geq w_i(v) \forall i \in S$$

Since v is a successful strategy for S, $w_i(v) \geq w_i(u) \forall i \in S$. Now, v' is a strategy for any $S' \supseteq S$; we have just shown that no player does worse under v' than under u. Because v is successful, there is some $j \in S$ such that

$$w_j(v) > w_j(u) \implies w_j(v') > w_j(u)$$

It follows that v' is a successful strategy for S'. $\square$

Together, Theorems 5 and 6 tell us that the successful pair strategies at utility profile u completely describe all the successful coalitions at u.

References

[BFC93] A. Ballardie, and P. Francis, and J. Crowcroft. Core Based Trees (CBT). In *Proc. ACM Sigcomm '93*, ACM Press, 1993.

[DC90] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems* **8**, 1990.

[DEF+96] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. *ACM/IEEE Transactions on Networking* **54**, 1996.

[GKL76] J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics* **6**, 1976.

[HC99] H. Holbrook and D. Cheriton. IP multicast channels: EXPRESS support for large-scale single-source applications, In *Proc of Sigcomm '99*, ACM Press, 1999.

[FS97] E. Friedman, and S. Shenker. Learning and Implementation in the Internet. Available in preprint form at `http://www.aciri.org/shenker/decent.ps`

[FPS00] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. To appear in *Journal of Computer and System Sciences*, special issue on Internet Algorithms. Extended abstract in *Proc. 32-nd ACM Symposium on the Theory of Computing*, ACM Press, 2000.

[GHW01] A. Goldberg, J. Hartline, and A. Wright. Competitive auctions and digital goods. In *Proc. 12th ACM/SIAM Symp. on Discrete Algorithms*, ACM Press/SIAM, 2001.

[GH00] A. Goldberg and J. Hartline. Competitive Auctions for Multicast Content. Manuscript, October 2000.

[JV01] K. Jain and V. Vazirani. Applications of approximation to cooperative games. To appear in *Proc. 33rd ACM Symposium on the Theory of Computing*, ACM Press, 2001.

[LN00] R. Lavi and N. Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proc. 2nd ACM Conference on Electronic Commerce (EC-00)*, ACM Press, 2000.

[M99] H. Moulin. Incremental cost sharing: characterization by strategyproofness. *Social Choice and Welfare* **16**, 1999.

[MS97] H. Moulin and S. Shenker. Strategyproof Sharing of Submodular Costs: Budget Balance Versus Efficiency. To appear in *Economic Theory*. Available in preprint form at `http://www.aciri.org/shenker/cost.ps`

[MT99] D. Monderer and M. Tennenholtz. Distributed Games: From Mechanisms to Protocols. In *Proc. 16-th National Conf. on Artificial Intelligence (AAAI 1999)*, 1999.

[Nis00] N. Nisan. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conference on Electronic Commerce (EC-00)*, ACM Press, 2000.

[NR99] N. Nisan and A. Ronen. Algorithmic mechanism design. To appear in *Games and Economic Behavior*. Extended abstract in *Proc. 31-st Annual ACM Symposium on the Theory of Computing*, ACM Press, 1999.

[NR00] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *Proc. 2nd ACM Conference on Electronic Commerce (EC-00)*, ACM Press, 2000.

[Par99] D. Parkes. ibundle: An efficient ascending price bundle auction. In *Proc. 1st ACM Conference on Electronic Commerce (EC-99)*, ACM Press, 1999.

[PU00] D. Parkes and L. Ungar. Iterative combinatorial auctions: Theory and practice. In *Proc. 17-th National Conf. on Artificial Intelligence (AAAI 2000)*, 2000.

[PLB+99] R. Perlman, C.-Y. Lee, A. Ballardie, J. Crowcroft., Z. Wang, T. Maufer, C. Diot, and M. Green. Simple multicast: A design for simple low-overhead multicast. IETF Internet Draft (Work in Progress), 1999.

[Rob79] K. Roberts. The Characterization of Implementable Choice Rules. In *Aggregation and Revelation of Preferences*, North-Holland, Amsterdam, 1979.

[San99] T. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent systems: A Modern Introduction to Distributed Artificial Intelligence.* MIT Press, Cambridge, MA, 1999.

[ST] Y. Shoham and M. Tennenholtz. On Rational Computability and Communication Complexity. To appear in *Games and Economic Behavior.*

[Wel93] M. Wellman. A market-oriented programming environment and its applications to distributed multicommodity flow problems. *Journal of AI Research* **1**, 1993.

[WWWM] M. Wellman, W. Walsh, P. Wurman, and J. Mackie-Mason. Auction protocols for decentralized scheduling. To appear in *Games and Economic Behavior.*