Approximation and Collusion in Multicast Cost Sharing[1]

Aaron Archer[2]
Cornell University, Operations Research Dept., Ithaca, NY 14853
E-mail: aarcher@orie.cornell.edu

and

Joan Feigenbaum[3] Arvind Krishnamurthy[4] Rahul Sami[5]
Yale University, Computer Science Dept., New Haven, CT 06520-8285
E-mail: jf@cs.yale.edu, arvind@cs.yale.edu, sami@cs.yale.edu

and

Scott Shenker[6]
ICSI, 1947 Center Street, Berkeley, CA 94704-1198
E-mail: shenker@icsi.berkeley.edu

Version: March 22, 2004

We investigate multicast cost sharing from both computational and economic perspectives. Recent work in economics leads 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. Subsequent work in computer science shows that the MC mechanism can be computed with only two modest-sized messages per link of the multicast tree but that computing the SH mechanism for p potential receivers can require $\Omega(p)$ bits of communication per link. We extend these results in two directions. First, we give a group-strategyproof mechanism that exhibits a tradeoff between the other properties of SH: It can be computed with exponentially lower worst-case communication than the SH algorithm, but it might fail to achieve exact budget balance (albeit by a bounded amount). Second, we completely characterize the groups that can strategize successfully against the MC mechanism.

Journal of Economic Literature Classification Entries: C60, C72.

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 *obedient* (*i.e.*, to follow the prescribed algorithm) or to be *adversaries* who "play against" each other. In contrast, the *strategic* users in game theory are neither obedient nor adversarial. Although one cannot assume that strategic users will follow the prescribed algorithm,

[1]Abstract in Proceedings of the 3rd ACM Conference on Electronic Commerce, Tampa FL, October 2001. This work was supported by the DoD University Research Initiative (URI) program administered by the Office of Naval Research under Grant N00014-01-1-0795.

[2]Supported by the Fannie and John Hertz Foundation.

[3]Supported in part by ONR grants N00014-01-1-0795 and N00014-01-1-0447 and NSF grants CCR-0105337, CCR-TC-0208972, ANI-0207399, and ITR-0219018.

[4]Supported in part by NSF grants CCR-9985304, ANI-0207399, and CCR-0209122.

[5]Supported by NSF Career grant CCR-9702980 and ONR grant N00014-01-1-0795.

[6]Supported in part by NSF grants ITR-0081698, ITR-0121555, and ANI-0207399.

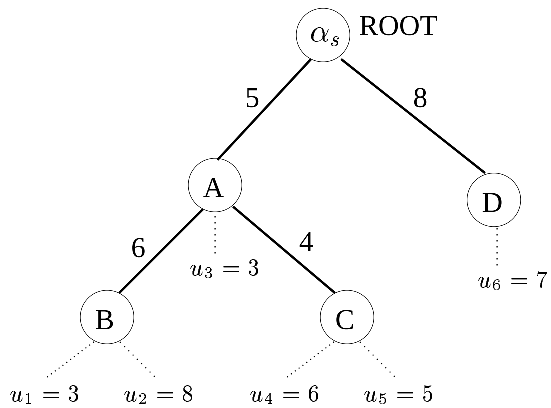

FIG. 1 A multicast cost-sharing problem.

one can assume that they will respond to incentives. Thus, one need not design algorithms that achieve correct results in the face of adversarial 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 (2001) 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. Multicast routing is a technique for transmitting a packet from a single source to multiple receivers without wasting network bandwidth. To achieve transmission efficiency, multicast routing constructs a directed tree that connects the source to all the receivers and sends only one copy of the packet over each link of the directed tree. When a packet reaches a branch point in the tree, it is duplicated and a copy is sent over each downstream link. Multicasting large amounts of data to large groups of receivers is likely to incur significant costs, and these costs need to be covered by payments collected from the receivers. However, receivers cannot be charged more than what they are willing to pay, and the transmission costs of shared network links cannot be attributed to any single receiver. Thus, one must design cost-sharing mechanisms to determine which users receive the transmission and how much they are charged.

Figure 1 depicts an instance of the multicast cost-sharing problem. There are six potential receivers, each located at a particular node of the multicast tree and each having a certain *utility value* for receiving the multicast transmission. For example, the notation $u_1 = 3$ beside the leftmost node on the second level from the top means that potential receiver number 1 is located at this node and is willing to pay at most 3 to receive the transmission. The numerical values on the links represent the costs of sending the transmission over those links. The source of the transmission is the root node at the top level of the tree. If $R \subseteq \{1, \ldots, 6\}$ is the set of actual receivers, then the transmission will be sent only to the nodes of the tree at which members of R are located. The total cost of this transmission will be the sum of the costs of the links in the smallest subtree that contains these nodes and the root. For example, if $R = \{2, 3, 4\}$, then the total cost of the transmission would be 15. The role of a *cost-sharing mechanism* is to determine, for each instance, what the receiver-set R should be and how much each member of R should be charged.

The multicast cost-sharing problem has been studied extensively in recent years, first from a networking perspective (Herzog *et al.*, 1997), then from a mechanism-design perspective (Moulin and Shenker, 2001), and most recently from an algorithmic perspective (Feigenbaum *et al.*, 2001; Feigenbaum *et al.*, 2002; Adler and Rubenstein, 2002; Jain and Vazirani, 2001; Fiat *et al.*, 2002). Computationally efficient cost-sharing algorithms are desirable because the computational resources of the multicast infrastructure (*i.e.*, link bandwidth and nodes' memory and CPU cycles) must be used to compute them; the *raison d'etre* of this infrastructure is to deliver content efficiently, not to do cost-sharing, and hence the latter must not consume enough resources to interfere with the former. All of the cost-sharing mechanisms in the existing literature have two basic properties: No Positive Transfers (NPT), which means that the mechanism cannot pay receivers to accept the transmission, and Voluntary Participation (VP), which means that no receiver can be forced to pay more than his utility value. The mechanisms that we present in this paper will satisfy these basic properties as well.

In addition to NPT and VP, 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 (*i.e.*, the sum of the receivers' utilities minus the total transmission cost), and it is said to be *budget-balanced* if the revenue raised from the receivers covers the total cost of the transmission exactly.

It is a classical result in game theory (Green and Laffont, 1979) that a strategyproof cost-sharing mechanism cannot be both budget-balanced and efficient. Moulin and Shenker (2001) 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 welfare 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 that link.

For the instance shown in Figure 1, the MC mechanism computes the receiver-set R to be $\{1, 2, 3, 4, 5\}$ resulting in a total transmission cost of 15 and overall welfare of 10. The SH mechanism does not include potential receiver number 1 in the receiver set, because this receiver's utility is not sufficient to cover an equitable share of the transmission cost, and hence computes R to be $\{2, 3, 4, 5\}$.

Our Results: The foregoing discussion makes it clear that the computational and game-theoretic properties of the SH and MC mechanisms are both worthy of study. It is easy to see (and is noted in Feigenbaum *et al.* (2001)) that both are polynomial-time computable by centralized algorithms. Feigenbaum *et al.* (2001) have further shown that there is a distributed algorithm that computes MC using only two messages per link. By contrast, Feigenbaum *et al.* (2003) shows that computing the SH mechanism requires, in the worst case, that $\Omega(|P|)$ bits be sent over $\Omega(|N|)$ links, where P is the set of potential receivers, and N is the set of tree nodes.

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 (Moulin and Shenker, 2001; Moulin, 1999).

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 an algorithm that is more communication-efficient than the natural SH algorithm (exponentially more so in the worst case), but it might fail to achieve exact budget balance or exact minimum welfare loss (albeit by a bounded amount).
- We completely characterize the groups that can strategize successfully against the MC mechanism and the conditions under which they can do so.

The rest of this paper is organized as follows. Section 2 provides necessary terminology and notation from algorithmic mechanism design and multicast cost sharing and explains what it means to "approximate" an algorithmic mechanism. In Section 3, 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 4, we present our result on successful collusion against the MC mechanism. Section 5 contains open problems.

2. TECHNICAL PRELIMINARIES

In this section, we review the basics of algorithmic mechanism design and multicast cost sharing. We also formulate the notion of "approximately computing a mechanism" that will be used in Section 3 below and comment on some aspects of our computational and strategic models.

2.1. Algorithmic Mechanism Design

The purpose of this section is to review the basics of algorithmic mechanism design. Readers already familiar with this area should skip to the next section.

In designing efficient, distributed algorithms and network protocols, computer scientists typically assume either that computational agents are *obedient* (*i.e.,* that they follow the protocol) or that they are *adversaries* (*i.e.,* that they may deviate from the protocol in arbitrary ways that harm other users, even if the deviant behavior does not bring them any obvious tangible benefits). In contrast, economists design market mechanisms in which it is assumed that agents are neither obedient nor adversarial but rather *strategic*: They respond to well defined incentives and will deviate from the protocol only for tangible gain. Until recently, computer scientists ignored incentive compatibility, and economists ignored computational efficiency.

The emergence of the Internet as a standard, widely used distributed-computing environment and of Internet-enabled commerce (both in traditional, "real-world" goods and in electronic goods and computing services themselves) has drawn computer scientists' attention to incentive-compatibility questions in distributed computation. In particular, there is growing interest in incentive compatibility in both distributed and centralized computation in the theoretical computer science community (see, *e.g.,* Archer and Tardos (2002); Feigenbaum *et al.* (2001); Fiat *et al.* (2002); Hershberger and Suri (2001); Jain and Vazirani (2001); Nisan and Ronen (2001); Roughgarden and Tardos (2002)) and in the "distributed-agents" part of the AI community (see, *e.g.,* Monderer and Tennenholtz (1999); Parkes (1999); Parkes and Ungar (2000); Sandholm (1999); Wellman (1993); Wellman *et al.* (2001)).

A standard economic model for the design and analysis of scenarios in which the participants act according to their own self-interest is as follows: There are n agents. Each agent i, for $i \in \{1, \dots, n\}$, has some private information t^i, called its *type*. For each mechanism-design problem, there is an *output specification* that maps each type vector $t = (t^1, \dots, t^n)$ to a set of allowed outputs. 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. For example, in an instance of the task-allocation problem studied in the original paper of Nisan and Ronen (2001), there are k tasks $z_1, \ldots, z_k$, agent i's type $t^i = (t^i_1, \ldots, t^i_k)$ is the set of minimum times in which it is capable of completing each of the tasks, the space of feasible outputs consists of all partitions $Z = Z^1 \sqcup \ldots \sqcup Z^n$, in which Z^i is the set of tasks assigned to agent i, and the valuation functions are $v^i(t^i, Z) = -\sum_{z_j \in Z^i} t^i_j$. Except for the private-type information, everything else in the scenario is public knowledge.

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 $p^i = p^i(a^1, \ldots, a^n)$. Agent i's *welfare* is $w_i = v^i(t^i, o) + p^i$, and it is this quantity that the agent seeks to maximize. A *strategyproof* mechanism is one in which the set of allowable types t^i is a subset of the strategy space A^i, and each agent maximizes his welfare by giving his type t^i as input regardless of what other agents do. In other words, the relation

$$v^i(t^i, o(a^{-i}, t^i)) + p^i(a^{-i}, t^i) \geq v^i(t^i, o(a^{-i}, a^i)) + p^i(a^{-i}, a^i)$$

(where a^{-i} denotes the vector of strategies of all players except player i) must hold for all i and all possible values of t^i, a^{-i} and a^i.

Thus, the mechanism wants each agent to report his private type truthfully, and it is allowed to pay agents in order to provide incentives for them to do so. In the task-allocation problem described above, an agent may be tempted to lie about the times he requires to complete each task, in the hope that his resulting allocation will have a higher valuation. If tasks were allocated by a strategyproof mechanism, he would have no incentive to do this, because his resulting payment would be lower; indeed it would be sufficiently lower that his overall welfare would be no greater than it would have been if he had told the truth.

For a thorough introduction to economic mechanism design, see Chapter 23 of the book by Mas-Colell, Whinston, and Green (Mas-Colell *et al.*, 1995).

In their seminal paper on *algorithmic mechanism design*, Nisan and Ronen (2001) add computational efficiency to the set of concerns that must be addressed in the study of how privately known preferences of a large group of selfish entities can be aggregated into a "social choice" that results in optimal allocation of resources. 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 $o()$ and the $p^i()$'s are polynomial-time computable. They also provide strategyproof, polynomial-time mechanisms for some concrete problems of interest, including LCPs and task allocation.

To achieve feasible algorithmic mechanisms within an Internet infrastructure, the mechanism-design framework must be enhanced with more than computational efficiency; it also requires a distributed computational model. 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" receives all of the inputs and doles out all of the outputs and payments. The first work to address this issue is the multicast cost-sharing paper of Feigenbaum, Papadimitriou, and Shenker. This work does not attempt to provide a general decentralized-mechanism computational model. Rather, it achieves the more modest goal of using the same network-algorithmic infrastructure that is needed for multicast to compute two natural mechanisms for assigning cost shares to the recipients of the multicast. It puts forth a general concept of "network complexity" that requires the distributed algorithm executed over an interconnection network T to be modest in four different respects: the total number of messages that agents send over T, the maximum number of messages sent over any one link in T, the maximum size of a message, and the local computational burden on agents.

Clearly, "network complexity" is not (yet) a well defined notion; indeed, there is not (yet) in general a full-fledged "complexity theory of Internet computation." We expect the development of more *prima facie* good (and bad) distributed algorithmic mechanisms to lead eventually to a satisfactory formalization of network complexity. In these early stages of the field, it suffices to note that the four measures of complexity identified in Feigenbaum *et al.* (2001) are all of practical importance and that, in particular, because the SH mechanism *cannot* be computed exactly when the maximum number of bits sent over a link is $o(|P|)$, where P is the set of potential receivers, it makes sense to attempt to approximate it.

2.2. Multicast Cost Sharing

The *multicast cost-sharing mechanism-design* problem involves an agent population P residing at a set of network nodes N that are connected by bidirectional network links L. The multicast flow emanates 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 spanning the nodes at which agents 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 smallest subtree of $T(P)$ required to reach the elements in R.[7] Since we usually draw the tree with the root α_s at the top, if node α lies along the path from node β to α_s (and $\alpha \neq \beta$) then we say α lies *above* β and is an *ancestor* of β. Symmetrically, β lies *below* α and is a *descendent* of α. If these two nodes are directly connected by a link, then α is β's *parent*, and β is a *child* of α. Each link $l \in L$ has an associated cost $c(l) \geq 0$ that is known by the nodes on each end, and each agent i assigns a *utility* value $u_i \geq 0$ to receiving the transmission. Let $u = (u_1, u_2, \ldots, u_{|P|})$ denote the vector of utilities. Only player i knows her true utility u_i.

A cost-sharing mechanism determines which agents receive the multicast transmission and how much each receiver is charged. Since the players' utilities are private information, the mechanism will ask each player to report some utility μ_i and base its decisions on the input vector μ of these reported utilities. We let $x_i(\mu)$ denote how much agent i is charged and $\sigma_i(\mu)$ denote whether agent i receives the transmission; $\sigma_i(\mu) = 1$ if the agent receives the multicast transmission, and $\sigma_i(\mu) = 0$ otherwise. The mechanism M is then a pair of functions $M(\mu) = (x(\mu), \sigma(\mu))$. The *receiver set* for a given input vector is $R(\mu) = \{i \mid \sigma_i(\mu) = 1\}$. An agent's individual welfare is given by the quasilinear form $w_i(\mu) = \sigma_i(\mu)u_i - x_i(\mu)$. Notice that $w_i(\mu)$ does depend on i's true utility u_i, but we suppress this in the notation. The cost of the tree $T(R)$ reaching a set of receivers R is $c(T(R))$, and the overall welfare, also known as *efficiency* 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 total transmission cost). Of course, the mechanism does not have direct access to u; so it can only compute $NW_\mu(R) = \mu_R - c(T(R))$, the net worth with respect to the reported utilities.

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

Let μ_{-i} denote the vector of all reported utilities besides player i's, so we can write μ as (μ_{-i}, μ_i). A *strategyproof* cost-sharing mechanism is one that satisfies the property $w_i(\mu_{-i}, u_i) \geq w_i(\mu_{-i}, \mu_i)$, for all i, u_i, μ_{-i}, and μ_i. In other words, no matter what utilities the other players report, i's best strategy is to report her true utility u_i (although i may have other strategies that are equally good).

[7] This approach is consistent with the design philosophy embedded in essentially all current multicast-routing proposals (see, *e.g.*, Ballardie *et al.* (1993); Deering and Cheriton (1990); Deering *et al.* (1996); Holbrook and Cheriton (1999); Perlman *et al.* (1999)).

Strategyproofness does not preclude the possibility of a group of users colluding to improve their individual welfares.

Any reported utility profile μ can be considered a group strategy for any group $S \supseteq \{i \mid \mu_i \neq u_i\}$. It will be handy to have a notation for perturbing reported utilities. If μ is one utility profile, and $\hat{\mu}_S$ is a vector of utilities for players in the set S, then let $\mu|^S\hat{\mu}_S$ denote the vector whose i^{th} component is μ_i if $i \notin S$ and $\hat{\mu}_i$ if $i \in S$. Thus, if S is the strategizing set, we can write the reported utility profile as $u|^S\mu_S$. 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 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 u, μ, and $S \supseteq \{i|u_i \neq \mu_i\}$:

$$\text{either} \quad w_i(\mu) = w_i(u) \ \forall i \in S$$

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

Economic considerations (Moulin and Shenker, 2001) point to two strategyproof mechanisms that are worthy of algorithmic consideration: marginal-cost (MC) and Shapley-value (SH). The MC mechanism, a member of the Vickrey-Clarke-Groves (VCG) family (Vickrey, 1961; Clarke, 1971; Groves, 1973), is *efficient*, which means that it chooses the receiver set R that maximizes $NW_\mu(R)$. Let W_μ be the net worth of this welfare-maximizing R. For each $i \in R$, let W_μ^{-i} be the net worth of the receiver set that the MC mechanism would have computed if i had not participated (*i.e.,* if μ_i had been set to 0). Then $W_\mu - W_\mu^{-i}$ measures the gain in overall welfare that results from i's participation. The cost share that MC assigns to i is $x_i(\mu) \equiv \mu_i - (W_\mu - W_\mu^{-i})$. MC is the only strategyproof and efficient mechanism that also has the following two properties:

NPT No Positive Transfers: $x_i(\mu) \geq 0$, or, in other words, the mechanism cannot *pay* receivers to receive the transmission.

VP Voluntary Participation: $w_i(\mu) \geq 0$, provided agent i reports truthfully (*i.e.* $\mu_i = u_i$); this implies that $x_i = 0$ whenever $\sigma_i = 0$ and that agents are always free to not receive the transmission and not be charged (by setting $\mu_i = 0$).

However, MC is not GSP and does not guarantee budget-balance.

By contrast, the SH mechanism is GSP and *budget-balanced*, where the latter means simply that $\sum_{i \in R} x_i = c(T(R))$, where R is the receiver set chosen by the mechanism. SH assigns cost shares x_i by dividing the cost $c(l)$ of each link l in $T(R)$ equally among all members $i \in R$ that are downstream of l. The SH receiver set is the largest $R \subseteq P$ such that $\mu_i \geq x_i$, for all $i \in R$. As mentioned in Section 1 above, there is no strategyproof mechanism that is both efficient and budget-balanced (Green and Laffont, 1979).

The MC mechanism has good network complexity: In Feigenbaum *et al.* (2001), a distributed algorithm is given that computes the MC receiver set and cost shares by sending just two modest-sized messages over each $l \in L$ and doing two very simple calculations at each node. We review this algorithm in Section 4 below. On the other hand, the SH mechanism has bad network complexity: In Feigenbaum *et al.* (2003), it is shown that any algorithm, deterministic or randomized, that computes SH must, in the worst case, send $\Omega(|P|)$ bits over $\Omega(|N|)$ links.

2.3. Strategically Faithful Approximate Mechanisms

In view of the proof given in Feigenbaum *et al.* (2003) that exact computation of the SH mechanism has unacceptably high communication cost, it is natural to ask the following question: Can one compute an *approximation* to the SH mechanism using an algorithm that is significantly more communication-efficient? To approach this question, we must first say what it means to "approximate the SH mechanism."

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, for some p, the L^p-difference of $x(u)$ and $x'(u)$ is small, for each u). 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.*, $(\sigma'(\mu), x'(\mu))$ might be very far from $(\sigma(u), x(u))$, where μ 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. 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.

The SH mechanism is GSP, budget-balanced, and, among all mechanisms with these two properties, the unique one that minimizes the worst-case welfare loss. We should therefore strive for a GSP mechanism that has low network complexity and is approximately budget-balanced and approximately welfare-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:

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

Here $T(P)$ is used to denote the non-numerical, "universal-tree" part of a multicast cost-sharing problem instance, the four components of which are the node-set N, the link-set L, the locations of the agents, and the multicast source location α_s.[8]

The *efficiency loss* of a mechanism M on an instance $I = (T(P), c(\cdot), u)$ is the difference between the optimal net worth of I (*i.e.*, that realized by the MC mechanism) and the net worth realized by M. The SH mechanism minimizes the worst-case loss in the following sense: For any given cost structure $(T(P), c(\cdot))$, the worst-case efficiency loss $L(M, T(P), c(\cdot))$ of a mechanism M on this cost structure in the maximum, over all possible utility profiles u, of the efficiency loss of M on the instance $(c(\cdot), T(P), u)$. Among all GSP, budget-balanced mechanisms, the SH mechanism achieves the minimum $L(M, T(P), c(\cdot))$; further, SH is the only mechanism to achieve this minimum for *all* cost structures $(T(P), c(\cdot))$. A mechanism M is "approximately efficiency-loss minimizing in the worst case" if there is a constant $\gamma > 1$ such that, for all cost structures $(T(P), c(\cdot))$, the worst-case efficiency loss of M on this cost structure is at most γ times the worst-case efficiency loss of SH on the same cost structure.

We do not obtain an approximate SH mechanism here, but we do make some progress toward the goal; our mechanism is GSP and fails to achieve exact budget balance and exact minimum-welfare loss by bounded amounts, but the bounds are not constant factors. Furthermore, there is a distributed algorithm that computes this mechanism using far less communication over the links of $T(P)$ than is needed by SH.

This notion of approximating a mechanism M that we use in this paper – roughly, "retain the strategic properties of M but approximate the other mechanism design goals" – is called *strategically*

[8] An alternative definition of approximate budget balance could allow for only a one-sided error, *e.g.*, a surplus but not a deficit, as in Jain and Vazirani (2001).

faithful approximation. Approximation is an increasingly active area of algorithmic mechanism design, and several other interesting notions of approximation have been put forth – see Section 5 of Feigenbaum and Shenker (2002) for an overview. Here we mention only the work that is most closely related to the results in this paper.

Nisan and Ronen (2000) were the first to address the question of approximate computation in algorithmic mechanism design. They considered VCG mechanisms in which optimal outcomes are NP-hard to compute (as they are in 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. Nisan and Ronen (2000) approach 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 a communication-inefficient, but polynomial-time computable, function; we are interested in network complexity in a distributed computational model, and Nisan and Ronen (2000) were interested in time complexity in a centralized computational model. Approximate multicast cost sharing was first addressed by Jain and Vazirani (2001). They exhibited a GSP, approximately budget-balanced, 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 welfare loss. Finally, "competitive-ratio" analysis (a form of approximation) has been studied for a variety of strategyproof auctions (see, *e.g.*, Fiat *et al.* (2002), Goldberg *et al.* (2001), and Lavi and Nisan (2000)).

2.4. Comments on the problem formulation

Our goal is to explore the relationships between incentives and computation in multicast cost sharing, but, before we do so, we first comment on several aspects of the model. The cost model we employ is a poor reflection of reality, in that transmission costs are not per-link; current network-pricing schemes typically only involve usage-based or flat-rate access fees, and the true underlying costs of network usage, though hard to determine, involve small incremental costs (*i.e.*, sending additional packets is essentially free) and large fixed costs (*i.e.*, installing a link is expensive). However, we are not aware of a well validated alternative cost model, and the per-link cost structure is intuitively appealing, relatively tractable, and widely used.

We assume that the total transmission costs are shared among the receivers. There are certainly cases in which the costs would more naturally be borne by the source (*e.g.*, broadcasting an infomercial) or the sharing of costs is not relevant (*e.g.*, a teleconference among participants from the same organization); in such cases, our model would not apply. However, we think that there will be many cases, particularly those involving the widespread dissemination of popular content, in which the costs would be borne by the receivers.

In some situations, such as the high-bandwidth broadcast of a long-lived event such as a concert or movie, the bandwidth required by the transmission is much greater than that required by a centralized cost-sharing mechanism (*i.e.*, sending all the link costs and utility values to a central site at which the receiver set and cost shares could be computed). For these cases, our feasibility concerns would be moot. However, Internet protocols are designed to be general-purpose; what we address here is the design of a protocol that would share multicast costs for a wide variety of uses, not just long-lived and high-bandwidth events. Thus, we need only claim that there are many scenarios in which our feasibility concerns would be relevant, not that our concerns are relevant in all scenarios.

In comparing the bandwidth required for transmission to the bandwidth required for the cost-sharing mechanism, one much consider several factors. First, and most obvious, is the transmission rate b of the application. For large multicast groups, it will be quite likely that there will be at least one user connected to the Internet by a slow modem. Because the multicast rate must be chosen to accommodate the slowest user, one can't assume that b will be large. Second, the bandwidth consumed on any particular link by the centralized cost-sharing mechanisms scales linearly with the number of users $p = |P|$, but the multicast's usage of the link is independent of the number of users. Third, one must consider the time increment Δ over which the cost accounting is done. For some events, such as a movie, it would be appropriate to calculate the cost shares once (at the beginning of the transmission) and not allow users to join after the transmission has started. For other events, such as the transmission of a shuttle mission, users would come and go during the course of the transmission. To share costs accurately in such cases, the time increment Δ must be fairly short. In centralized cost sharing, the accounting bandwidth on a single link scales roughly as p, which must be compared to the bandwidth Δb used over a single accounting interval. Although small multicast groups with large Δ and b could easily use a centralized mechanism, large multicast groups with small Δ and b could not.

We have assumed that budget-balanced cost sharing, where the sum of the charges exactly covers the total incurred cost, is a natural goal of a charging mechanism. If the charging mechanism were being designed by a monopoly network operator, then one might expect the goal to be maximizing revenue. There have been some recent investigations of revenue-maximizing charging schemes for multicast (see, *e.g.*, Fiat *et al.* (2002)), but here we assume, as in Herzog *et al.* (1997); Moulin and Shenker (2001); Feigenbaum *et al.* (2001) and Adler and Rubenstein (2002), that the charging mechanism is decided by society at large (*e.g.*, through standards bodies) or through competition. Competing network providers could not charge more than their real costs (or otherwise their prices would be undercut) nor less than their real costs (or else they would lose money), and so budget balance is a reasonable goal in such a case. For some applications, such as big-budget movies, the bandwidth costs will be insignificant compared to the cost of the content, and then different charging schemes will be needed, but for low-budget or free content (*e.g.*, teleconferences) budget-balanced cost-sharing is appropriate.

Lastly, in our model it is the *users* who are selfish. The routers (represented by tree nodes), links, and other network-infrastructure components are obedient. Thus, the cost-sharing algorithm does not know the individual utilities u_i, and so users could lie about them, but once they report them to the network infrastructure (*e.g.*, by sending them to the nearest router or accounting node), the algorithms for computing $x(u)$ and $\sigma(u)$ can be reliably executed by the network. Ours is the simplest possible strategic model for the distributed algorithmic mechanism-design problem of multicast cost sharing, but, even in this simplest case, determining the inherent network complexity of the problem is non-trivial. Alternative strategic models (*e.g.*, ones in which the routers are selfish and their strategic goals may be aligned or at odds with those of their resident users) may also present interesting distributed algorithmic mechanism-design challenges. Preliminary work along these lines is reported in Mitchell and Teague (2002).

3. TOWARDS APPROXIMATING THE **SH** MECHANISM

In this section, we develop a GSP mechanism that exhibits a tradeoff between the other properties of the Shapley value: It can be computed by an algorithm that is more communication-efficient than the natural SH algorithm (exponentially more so in the worst case), but it might fail to achieve exact budget balance or exact minimum welfare loss (albeit by a bounded amount).

First, in Section 3.1, we review the natural SH algorithm given in Feigenbaum *et al.* (2001). In Section 3.2, we give an alternative SH algorithm that also has unacceptable network complexity but that leads naturally to our approach to approximation. In Sections 3.3, 3.4, and 3.5, we define a new mechanism that has low network complexity, prove that it is GSP, and obtain bounds on the budget deficit and the welfare loss.

3.1. The natural multi-pass SH algorithm

The Shapley-value mechanism divides the cost of a link l equally among all receivers downstream of l. The mechanism can be characterized by its cost-sharing function $f : 2^P \mapsto \Re^P_{\geq 0}$ (Moulin and Shenker, 2001; Moulin, 1999). For a receiver set $R \subseteq P$, player i's cost share is $f_i(R)$. Feigenbaum *et al.* (2001) present a natural, iterative algorithm that computes SH. We restate it here:

The simplest case of the SH cost-share problem is the one in which all u_i are sufficiently large to guarantee that all of P receives the transmission. (For example, $u_i > c(T(P))$, for all i, would suffice.) For this case, the SH cost shares can be computed as follows.[9] Do a bottom-up traversal of the tree that determines, for each node α, the number p_α of users in the subtree rooted at α. Then, do a top-down traversal, which the root initiates by sending the number $md = 0$ to its children. After receiving message md, node α computes $md' \equiv \left(\frac{c(l)}{p_\alpha}\right) + md$, where l is the network link between α and its parent, assigns the cost share md' to each of its resident users, and sends md' to each child. Thus, each user ends up paying a fraction of the cost of each link in its path from the source, where the fraction is determined by the number of users sharing this link.

In the general case, we initially start, as before, with $R = P$ and compute the cost shares as above. However, we cannot assume that $u_i \geq md'$ for all i, and so some users may prefer not to receive the transmission. After each pass up and down the tree, we update R by omitting all users i such that $u_i < md'$ and repeat. The algorithm terminates when no more users need to be omitted.

Unfortunately, this algorithm could make as many as $|P|$ passes up and down the tree and send a total of $\Omega(|N| \cdot |P|)$ messages in the worst case. Moreover, Feigenbaum *et al.* (2003) contains a corresponding lower bound: Any algorithm that computes SH requires $\Omega(|N| \cdot |P|)$ bits of communication in the worst case.

3.2. 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. We do this by communicating, in a single message, a digest of the utility profile of all the players in a subtree. This algorithm still sends more than $|N| \cdot |P|$ communication bits in the worst case, and thus it 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 in a significantly more communication-efficient manner and has other desirable properties.

Let μ be the (reported) utility profile. Then, for every link l in $T(P)$, the digest we compute is: $n_l(p, \mu) \stackrel{def}{=}$ *the number of players in the subtree beneath l who are each willing to pay p for the links above l* (*i.e.*, the number of players in this subtree who will not drop out of the receiver set when their cost share for the links from the root down to but excluding l is p).

(We put the utility profile μ in explicitly as an argument so that it can be used below in the proof of group strategyproofness; however, in any one run of the algorithm, μ is fixed.)

[9] This simple case is essentially a distributed version of the linear-time algorithm given in Megiddo (1978).

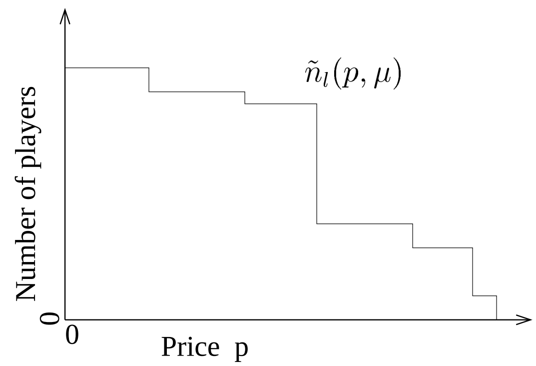

FIG. 2 The function $n_l(p, \mu)$ computed for each link l

Note that this definition requires that the cost from the leaves through l has already been adjusted for. The information conveyed through the function $n_l(p, \mu)$ is a sufficient digest of the costs and utilities in the subtree beneath l, because the SH mechanism does not distinguish between receivers downstream of l when sharing the cost of l or its ancestors; all such receivers pay the same amount for these links. For each link, we compute this function at all prices p. The function $n_l(p, \mu)$ is monotonically decreasing with p, and, for any given utility profile μ, can be represented with at most $|P|$ points with coordinates (p_i, n_i) corresponding to the "corners" in the graph of $n_l(p, \mu)$ in Figure 2. We use this *list-of-points* representation of $n_l(p, \mu)$ in our algorithm.

The Feigenbaum *et al.* (2001) statement of the multicast cost-sharing 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 link-cost zero. Thus, we assume, without loss of generality, that all players are at leaf nodes only.

The function $n_l(p, \mu)$ is computed at the node α_l below l in the tree. The computation is easy if α_l is a leaf node. Let p_{α_l} be the number of agents at α_l, and assume that $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_{p_{\alpha_l}}$. Let $c(l)$ be the cost of link l. For a given price p, compute $n_l(p, \mu)$ as follows. Let $k = 0$. If $p + \frac{c(l)}{p_{\alpha_l} - k} \leq \mu_{(p_{\alpha_l} - k)}$, then stop with $n_l(p, \mu) = p_{\alpha_l} - k$. Otherwise, increment k by 1 and repeat the test. If k reaches $p_{\alpha_l} - 1$, and the test fails (*i.e.*, if $p + c(l) > \mu_1$), then stop with $n_l(p, \mu) = 0$.

If α_l is not a leaf node, we have to include the functions reported by its children in this calculation. Suppose we are at node α_l and have received the functions $n_{l_i}(p, \mu)$ from all the child links $\{l_1, l_2, \ldots l_r\}$ of l. We can compute $n_l(p, \mu)$ in two steps:

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

$$m_l(p, \mu) = \sum_{i=1}^{r} n_{l_i}(p, \mu)$$

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

- **Step 2**: Now, we have to account for the cost $c(l)$ of the link l to compute the function $n_l(p, \mu)$.

For any p such that $p \cdot m_l(p,\mu) \geq c(l)$, we have

$$n_l(p - \frac{c(l)}{m_l(p,\mu)}, \mu) \;\geq\; m_l(p,\mu)\;, \tag{1}$$

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

$$q - \frac{c(l)}{m_l(q,\mu)} \geq p - \frac{c(l)}{m_l(p,\mu)}$$

and so could also support the price $p' = p - (c(l)/m_l(p,\mu))$ each for the links above l. However, the value of $n_l(p,\mu)$ must correspond to $m_l(p',\mu)$ for some $p' \geq p$, because every player beneath l who receives the transmission pays an equal amount for the link l. It follows that

$$n_l(p,\mu) = \max_{\left\{p' - \frac{c(l)}{m_l(p',\mu)} \geq p\right\}} m_l(p',\mu) \tag{2}$$

When the right hand side of Equation 2 is undefined (because there is no p' satisfying the condition), we set $n_l(p,\mu) = 0$. Given a list of points $(p^{(i)}, m^{(i)})$ corresponding to $m_l(\cdot)$, we can compute $n_l(\cdot)$ through the following procedure: For each point $(p^{(i)}, m^{(i)})$, we get the transformed point $(p^{(i)} - (c(l)/m^{(i)}), m^{(i)})$. 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 $n_l(\cdot)$ for all links, until we reach the root. At the root, we can combine the functions received from the root's children to get $m_{root}(\cdot)$. Because there are no further costs to be shared, it follows that there are $m = m_{root}(0,\mu)$ players that 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 largest 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 $n_l(\cdot)$ on the way up the tree, we compute the prices on the way down as follows: For each link l, we let x_l be the cost share of any receiver below l for the path down to (but not including) l. If l is the link from node β to β's parent, then we use x_l and x_β interchangeably. Then, $x_{root} = 0$ and, if l has child links $l_1, l_2, \ldots l_k$,

$$x_{l_j} = x_l + \frac{c(l)}{n_l(x_l,\mu)} \tag{3}$$

We descend the tree in this manner until we get a price x_i for every player $i \in P$: If i is at node β, and l is the link from β to its parent, then $x_i = x_l + \frac{c(l)}{n_l(x_l,\mu)}$. Then, we include i in $R(\mu)$ iff $x_i \leq \mu_i$, 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 $n_l(x_l,\mu)$ players beneath l who can pay x_l for the path down to but excluding l. It follows that

$$\forall j \; n_l(x_l,\mu) = m_l(x_{l_j},\mu) = \textstyle\sum_i n_{l_i}(x_{l_i},\mu)\;.$$

Using this inductively until we reach the leaves, we can show that there are $n_l(x_l,\mu)$ players downstream of l in the receiver set chosen by the algorithm, *i.e.*, with $x_i \leq \mu_i$. Equation 3 then shows that the cost of each link is exactly balanced, and hence the overall mechanism is budget-balanced. ∎

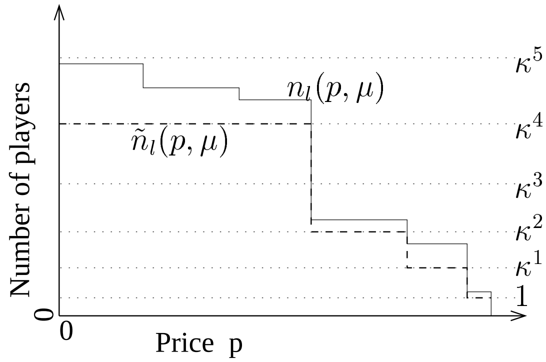

FIG. 3 Approximation to $n_l(p, \mu)$.

LEMMA 2. *The receiver set computed by this algorithm is the same as the receiver set computed by the Shapley-value algorithm given in Section 2.1.*

Proof. By *Lemma 1*, we know that the set $R(\mu)$ constructed can bear the cost of transmitting to $R(\mu)$. Let $\overline{R}(\mu)$ be the receiver set chosen by the iterative Shapley-value algorithm (*i.e.*, the one in Section 2.1). Because $\overline{R}(\mu)$ is the greatest fixed point, $\overline{R}(\mu) \supseteq R(\mu)$.

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

$$n_l(\overline{x}_l(\mu), \mu) \geq \overline{n}_l(\mu)$$

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

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.

3.3. A communication-efficient approximation of $\mathbf{n_l}(\cdot)$

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 $n_l(\cdot)$ passed up link l requires $|P|$ points (p_i, n_i) to represent it, which is undesirable.

Our approach to approximating the SH mechanism is as follows: We replace the function $n_l(\cdot)$ in the one-pass SH mechanism by a small **approximate representation of** $n_l(\cdot)$; only this approximate representation is communicated up the tree, resulting in an exponential saving in the worst-case number of communication bits. What should this approximation look like? To begin with, we would like to **underestimate** $n_l(\cdot)$ at every point, effectively underestimating the players' utilities, so that we can still compute a feasible receiver set in one pass.

For each link l, instead of $n_l(p, \mu)$, the mechanism uses an under-approximation $\tilde{n}_l(p, \mu)$. The approximation we choose is simple and is illustrated in *Figure 3*. For some parameter $\kappa > 1$, we round down all values of $n_l(p, \mu)$ to the closest power of κ. The resulting function $\tilde{n}_l(p, \mu)$ has at most $(\log |P| / \log \kappa)$ "corners," and so it can be represented by a list of $\mathcal{O}(\log |P|)$ points.

At the leaf nodes, we first compute the exact function $n_l(p,\mu)$ as before, and from this we compute the approximation $\tilde{n}_l(p,\mu)$ as illustrated in *Figure 3*. At non-leaf nodes, we compute $\tilde{n}_l(p,\mu)$ by using the following modified versions of Steps 1 and 2 of the one-pass algorithm:

- **Step 1'**: Compute

$$\hat{m}_l(p,\mu) = \sum_{l_i} \tilde{n}_{l_i}(p,\mu)$$

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

- **Step 2'**: First, adjust for cost $c(l)$ as before

$$\hat{n}_l(p,\mu) = \max_{\left\{p' - \frac{c(l)}{\hat{m}_l(p',\mu)} \geq p\right\}} \hat{m}_l(p',\mu)$$

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

$$\tilde{n}_l(p,\mu) = \kappa^{\lfloor \log_\kappa \hat{n}_l(p,\mu) \rfloor}$$

Because $\tilde{n}(\cdot)$ is given in the list-of-points representation, this is easily done by dropping elements of the list that do not change $\tilde{n}_l(p,\mu)$.

The function $\hat{n}_l(\cdot)$ computed on the way up is stored at the node beneath l.[10] On the way down, we compute

$$x_{l_j} = x_l + \frac{c(l)}{\hat{n}_l(x_l,\mu)}$$

Note that Step 2' guarantees that there are at least $\hat{n}_l(x_{l_j},\mu)$ players beneath l who can afford to pay $p' = x_{l_j}$ for the links from the root through l.

We can now define a mechanism (called Mechanism SF, for "step function") by computing x_i for $i \in P$ as in the one-pass algorithm for SH in Section 2.2, including i in the receiver set if $x_i \geq \mu_i$, and assigning cost share x_i to i if i is included. However, we now have a situation in which the number of receivers downstream of link l is potentially greater than $\tilde{n}_l(x_l,\mu)$, because $\tilde{n}_l(\cdot)$ is an under-approximation. Thus, SF does not achieve exact budget balance; there may be a budget surplus.

For example, consider running mechanism SF on the instance shown in *Figure 4* with $\kappa = 2$. Node B computes $\hat{n}_{l_1}(\cdot)$ as follows: If only player 4 is included, he would have to pay the entire cost of link l_1 and hence have only 12 left to pay for link l_3; this gives us a corner at point $(12,1)$. Further computations show that the other corner points are $(11,2)$, $(10,3)$, and $(3,4)$. *Figure 4* also shows the approximate function $\tilde{n}_{l_1}(\cdot)$: the only difference is that the corner at $(10,3)$ is dropped. Similarly, node C computes $\tilde{n}_{l_2}(\cdot)$; in this case, there is a single corner at $(7,2)$.

Now, node A receives the approximate functions $\tilde{n}_{l_1}(\cdot)$ and $\tilde{n}_{l_2}(\cdot)$. It then combines them to compute $\hat{n}_{l_3}(\cdot)$. It turns out that the only way to share the cost of l_3, based on the received $\tilde{n}_{l_1}(\cdot)$ and $\tilde{n}_{l_2}(\cdot)$, is to admit two players from each of A and B; each of these players is willing to pay at least 7 for links l_3 and above, and so can share the cost of link l_3 and be willing to pay up to 1 more. Thus, the function $\hat{n}_{l_3}(\cdot)$ has a single corner at $(1,4)$. Our approximation procedure makes no difference in this case, and so the function $\tilde{n}_{l_3}$ is identical. Finally, the root receives $\tilde{n}_{l_3}$, and as there are no additional costs to share, it computes that transmission is feasible.

On the way down, the payments are computed as follows: At node A, the cost of l_3 is divided by $\hat{n}_{l_3}(0,u) = 4$, and thus $x_{l_1} = x_{l_2} = 6$. Node B then adds on the additional cost of l_1, and divides it

[10] If there are space constraints, it is easy to modify the mechanism to store $\tilde{n}_l(\cdot)$ instead, by rounding $\hat{m}_l$ to a compact approximation $\tilde{m}_l$ and using this function to compute $\tilde{n}_l(\cdot)$ in Step 2.

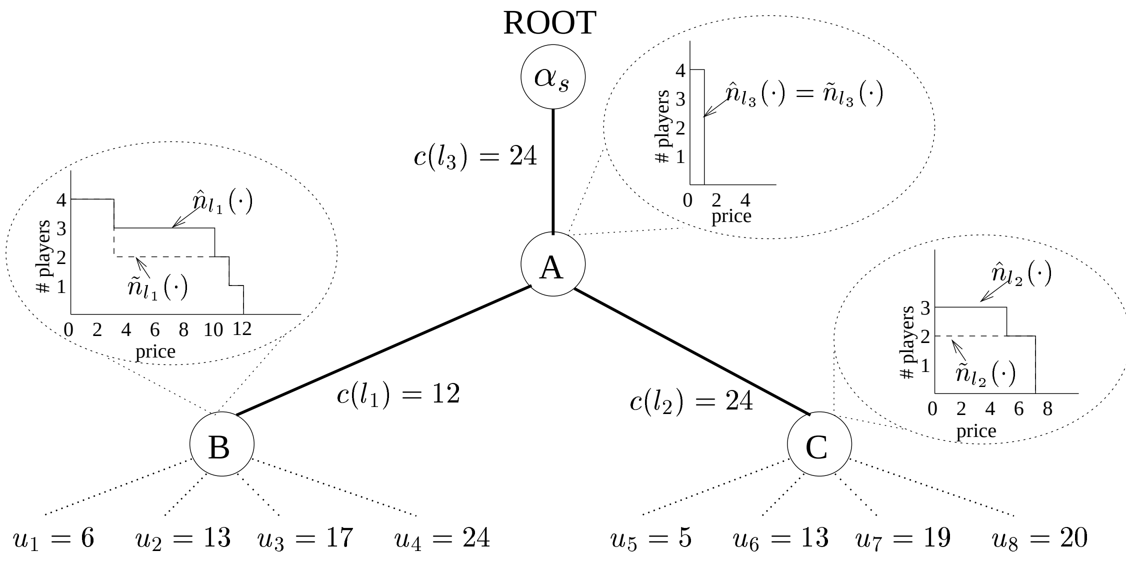

FIG. 4 Example illustrating budget surplus of Mechanism SF with $\kappa = 2$.

among $\hat{n}_{l_1}(6, u) = 3$ players. Thus, the ask price for players at node B is $6 + 4 = 10$; players $2, 3$, and 4 are included in the receiver set and pay 10 each. Similarly, the ask price at node C is computed to be $6 + 12 = 18$; players 7 and 8 are included and pay 18 each. The total amount collected is 66, but the cost of transmission is only 60, resulting in a surplus of 6. This surplus arises because node A counted on having only 4 receivers sharing the cost of l_3, whereas there were actually 5 receivers.

3.4. Group strategyproofness of mechanism SF

Notation

Throughout this section, we use $u = (u_1, u_2, \ldots, u_n)$ to indicate the true utility profile of the players. Recall that $\mu|^i r_i$ denotes the utility profile $(\mu_1, \mu_2 \ldots, \mu_{i-1}, r_i, \mu_{i+1}, \ldots, \mu_n)$, *i.e.*, the utility vector μ perturbed by replacing μ_i by r_i.

Now, let μ be the reported utility profile. Then $S = \{i \mid u_i \neq \mu_i\}$ 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:

$$\forall i \in S \;\; w_i(\mu) \geq w_i(u)$$

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

We prove that mechanism SF is GSP in three steps: First, we prove that, if there is a successful (individual or group) strategy, there is a successful strategy μ in which all colluding players *raise* their utility, *i.e.*, $\mu_i \geq u_i$. 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 $x_i \leq u_i < \mu_i$, then the outcome of the mechanism (both receiver set and cost shares) is unchanged in moving from strategy μ to $\mu|^i u_i$. Finally, we combine these two results to show that a successful strategy against mechanism SF cannot exist.

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:

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

(i). $\forall l, x \;\; \tilde{n}_l(x,\mu) \geq \tilde{n}_l(x,u)$

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

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

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

Proof. Note that our approximation technique has the property that, if $\hat{n}_l(x,\mu) \geq \hat{n}_l(x,u)$, then $\tilde{n}_l(x,\mu) \geq \tilde{n}_l(x,u)$. Statement *(i)* is then immediately true at the leaves and follows by induction at non-leaf nodes. Because the cost of any link l is divided among $\hat{n}_l(x_l,\mu)$ 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 μ, we can get a successful strategy μ' by raising μ_i to u_i whenever $\mu_i < u_i$. However, we first have the technical detail of eliminating non-receivers from the strategizing group:

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

Proof. Because $i \notin \tilde{R}(\mu)$, $x_i(\mu) > \mu_i$. When $\mu_i \leq u_i$, the statement follows directly from Lemma 3. When $\mu_i > u_i$, we can show that $\tilde{n}_l(x_l(\mu),\mu') = \tilde{n}_l(x_l(\mu),\mu)$ by induction on the height of l (where l is the link from the location of i to its parent), and the statement follows. ∎

Combining the last two results, we get:

LEMMA 5. *Suppose a group S has a successful strategy. Then, S has a successful strategy μ' where $\mu'_i \geq u_i$.*

Proof. By lemma 4, we can assume, without loss of generality, that S has a successful strategy μ such that $S \subseteq R(\mu)$. Define a sequence of strategies

$$\mu = \mu^{(0)}, \mu^{(1)}, \ldots \mu^{(n-1)}, \mu^{(n)} = \mu'$$

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

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 μ be the perturbed profile obtained by increasing one element of u ($\mu = u|^i\mu_i$, where $\mu_i > u_i$). If $u_i \geq x_i(u)$, then*

$$\forall l, \forall x < x_l(u) \;\; \tilde{n}_l(x,\mu) = \tilde{n}_l(x,u)$$

Proof. It is obviously true if i is at a leaf and l is the link from the leaf to its parent, because the utility μ_i only affects the value of $\tilde{n}_l(\cdot)$ 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 $\tilde{n}_l(\cdot)$, 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 μ. This follows from the fact that $\tilde{n}_l(\cdot)$ is not evaluated at prices above $x_l(u)$ on the way down, and so inductively $x_l(\mu) = x_l(u)$ for all links l. Hence, each player gets the same ask price $x_i(\mu) = x_i(u)$.

We can now prove the main result:

THEOREM 1. *Mechanism SF is GSP.*

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 μ for some set S, where every member of S receives the transmission after the strategy. Define the sequence of strategies:

$$\mu = \mu^{(0)}, \mu^{(1)}, \ldots \mu^{(n-1)}, \mu^{(n)} = u$$

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

We have an alternative proof that mechanism SF is GSP that uses Moulin's characterization of budget-balanced mechanisms based on cross-monotonic cost-sharing functions (Moulin, 1999). We give this alternative proof in the appendix below.

3.5. Mechanism SSF: bounded budget deficit and welfare loss

While mechanism SF is group strategyproof and has a bounded budget deficit, it has a potentially fatal flaw: it may output an empty receiver set in situations in which the SH mechanism would give a large receiver set. As a result, it may incur a very large welfare loss with respect to the SH mechanism. 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. The goal of the modification is to ensure that, for every utility profile, the mechanism has a receiver set at least as large as the SH receiver set. We do this by discounting the cost of each link by a bounded fraction; this converts the budget surplus of mechanism SF to a budget deficit, but improves the worst-case welfare loss.

This mechanism works as follows:

Mechanism SSF:
Let h_l be the height of link l in the tree. (If one of the endpoints of link l is a leaf, then $h_l = 1$.) Then, define the *scaled cost* $c^\kappa(l)$ of the link l to be $c(l)/(\kappa^{h_l})$. Run mechanism SF assuming link costs $c^\kappa(l)$ instead of $c(l)$, to compute a receiver set $R^\kappa(u)$ and cost shares $x_i^\kappa(u)$.

LEMMA 7. *Mechanism SSF is GSP.*

Proof. The player's utility does not affect the scaled costs, and mechanism SF is GSP for any tree costs. ∎

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

LEMMA 8. *For any link l, let $\tilde{n}_l^\kappa(x,u)$ be the approximation computed by mechanism SSF. Let $n_l(x,u)$ and x_l be defined as in the one-pass exact Shapley-value algorithm given in Section 2.2. Then,*

$$\forall l \quad \tilde{n}_l^\kappa(x_l,u) \geq \frac{n_l(x_l,u)}{\kappa^{h_l}}$$

Proof. We prove the statement by induction on h_l. For $h_l = 1$, it is true because of our approximation method. Suppose the statement is true for all links of height no more than r, and $h_l = r+1$. Let $\{l_1, l_2, \ldots, l_k\}$ be the child links of l. By the inductive assumption, $\tilde{n}_{l_i}^\kappa(x,u) \geq (n_{l_i}(x_{l_i},u))/\kappa^r$. It follows that

$$\hat{m}_l^\kappa(x_{l_i},u) = \sum_{i=1}^{k} \tilde{n}_{l_i}^\kappa(x_{l_i},u) \tag{4}$$

$$\geq \frac{n_l(x_l,u)}{\kappa^r} \tag{5}$$

From the computation of the ask prices x_l and x_{l_i} in the exact Shapley value mechanism, we know

$$x_l = x_{l_i} - \frac{c(l)}{n_l(x_l,u)} .$$

Let

$$x' = x_{l_i} - \frac{c^\kappa(l)}{\hat{m}_l^\kappa(x_{l_i},u)} . \tag{6}$$

Then, $x' \geq x_l$ follows from Equation (5).

Now, in *Step 2'* of mechanism SSF, the function $\hat{m}_l^\kappa(\cdot)$ is adjusted for the scaled cost $c^\kappa(l)$ to compute the function $\hat{n}_l^\kappa(\cdot)$. Equation (6) guarantees that $\hat{m}_l^\kappa(x_{l_i},u)$ players in the subtree below l can share the additional cost $c^\kappa(l)$ and still be willing to pay x' each for links above l. Thus, we have

$$\hat{n}_l^\kappa(x',u) \geq \hat{m}_l^\kappa(x_{l_i},u) ,$$

and, because $x' \geq x_l$, $\hat{n}_l^\kappa(x_l,u) \geq \hat{n}_l^\kappa(x',u)$. Finally, in passing from $\hat{n}^\kappa(\cdot)$ to $\tilde{n}^\kappa(\cdot)$, we get

$$\tilde{n}_l^\kappa(x_l,u) \geq \frac{\hat{n}_l^\kappa(x_l,u)}{\kappa}$$

$$\tilde{n}_l^\kappa(x_l,u) \geq \frac{n_l(x_l,u)}{\kappa^{r+1}}$$

And thus the statement is proved by induction. ∎

LEMMA 9. *$R^\kappa(u) \supseteq R(u)$.*

Proof. Using Lemma 8,

$$\frac{c^\kappa(l)}{\tilde{n}_l^\kappa(x_l,u)} \leq \frac{c(l)}{n_l(x_l,u)} ,$$

and we can show inductively that $x_l^\kappa \leq x_l$ for all links l. Because this is true at the leaves, it follows that $R^\kappa(u) \supseteq R(u)$. ∎

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 κ and the height h of the tree:

THEOREM 2.

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

Proof. Let $X = \sum_{i \in R^\kappa(u)} x_i^\kappa(u)$. Because mechanism SF never runs a deficit,

$$X \geq c^\kappa(T(R^\kappa(u))) \geq \frac{c(T(R^\kappa(u)))}{\kappa^h}.$$

We now show that mechanism SSF never runs a budget surplus. For each link l, let x_l denote the offer price computed by mechanism SSF. Consider a link l, and let $l_1, l_2, \cdots, l_k$ be its child links. Note that the cost of link l is factored into x_{l_i} by assuming that there are $\hat{n}_l^\kappa(x_l, u)$ receivers downstream of l. It is sufficient to prove that, for any link l, the number of receivers downstream of l (in R^k) is at most $\kappa^{h_l} \cdot \hat{n}_l^\kappa(x_l, u)$; as the cost of link l has been scaled down by κ^{h_l}, it follows that we never collect a surplus with respect to the true cost.

We prove this by induction on the height h_l of l. When $h_l = 1$, this is clearly true: there are exactly $\hat{n}_l^\kappa(x_l, u)$ receivers downstream of l. Assume it is true for all links of height at most r, and consider a link l of height $r+1$. By the inductive assumption, for each child link l_i, we have

$$\hat{n}_{l_i}^\kappa(x_{l_i}, u) \geq \frac{1}{\kappa^r} \times \text{ number of receivers downstream of } l_i \text{ in } R^k \text{ .}$$

Thus, we have

$$\tilde{n}_{l_i}^\kappa(x_{l_i}, u) \geq \frac{1}{\kappa^{r+1}} \times \text{ number of receivers downstream of } l_i \text{ in } R^k \text{ ,}$$

and so

$$\hat{m}_l^\kappa(x_{l_i}, u) \geq \frac{1}{\kappa^{r+1}} \times \text{ number of receivers downstream of } l \text{ in } R^k \text{ .}$$

Finally, the computation of the price x_{l_i} from x_l satisfies $\hat{n}_l^\kappa(x_l, u) = \hat{m}_l^\kappa(x_{l_i}, u)$, which gives us

$$\hat{n}_l^\kappa(x_{l_i}, u) \geq \frac{1}{\kappa^{r+1}} \times \text{ number of receivers downstream of } l \text{ in } R^k \text{ .}$$

Thus, by induction this is true for every link l. The total payment collected for any link l is at most $\kappa^{h_l} c^\kappa(l) \leq c(l)$, and so mechanism SSF never runs a budget surplus. ∎

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

THEOREM 3.

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

Proof. The welfare of the receiver set $R^\kappa(u)$ is

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

Now, for any subtree T_j of T^κ,

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

and hence

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

∎

To summarize, Mechanism SSF sends $O(\log_\kappa n)$ points (p_i, n_i) over each link, incurs a cost of at most κ^h times the revenue collected, and has an welfare loss of at most $(\kappa^h - 1)U$ with respect to the SH mechanism.

For example, when $|P| = 100,000$ and $h = 5$, the natural algorithm for the SH mechanism given in Feigenbaum *et al.* (2001) would require about $100,000$ messages to be sent across a link in the worst case. Our algorithm for SSF requires one bottom-up pass and one top-down pass, *i.e.*, exactly two messages over each link. The maximum size of each point (p_i, n_i) in a message in the bottom-up pass is always bounded by $O(\log|P| + \max_{i \in P} \log u_i)$ bits, and the maximum size of a message sent in the top-down pass is always bounded by $O(\log(\sum_{l \in L} c(l)))$ bits. For $|P| = 100,000$, $h = 5$, and $\kappa = 1.03$, SSF has a budget deficit of at most 14% of the tree cost and a worst-case welfare loss with respect to SH of at most 16% of the total utility, and the largest message sent in the bottom-up pass contains at most 400 points (p_i, n_i). As another example, when $|P| = 10^6$ and $h = 10$, we can use $\kappa = 1.02$ to achieve a worst-case deficit of 18% and worst-case welfare loss of 22% of the total utility, with maximum bottom-up message size of 700 points, or use $\kappa = 1.04$ to achieve corresponding bounds 33%, 48%, and 350 points.

4. GROUP STRATEGIES THAT SUCCEED AGAINST THE MC MECHANISM

The *marginal cost* (MC) mechanism for multicast cost sharing is an instance of the Vickrey-Clarke-Groves (VCG) family of mechanisms (Vickrey, 1961; Clarke, 1971; Groves, 1973). As described in Section 2.2, the MC mechanism selects the receiver set that maximizes total net worth, and charges each receiver her reported utility minus a "bonus" equal to the marginal value she added to the system. VCG mechanisms are always strategyproof, but in general not group strategyproof. In this section, we characterize exactly how the MC mechanism fails to be group strategyproof.

We say that a strategy μ for a group S is a *successful group strategy at utility profile u* if

- $S \supseteq \{i \in P | \mu_i \neq u_i\}$,
- $\forall i \in S,\ w_i(\mu) \geq w_i(u)$, and
- $\exists j \in S$ such that $w_j(\mu) > w_j(u)$.

In other words, a successful group strategy at u is one that (compared to truthtelling) harms none of the members of the group and benefits at least one. Note that the member who benefits may not be one of the members who misrepresented her utility. If the group S has only two members, we call the strategy a *successful pair strategy.* If there is no group that has a successful strategy at u, then we say that the mechanism is *GSP at u.* A GSP mechanism is one that is GSP at all u.

It is well known that the MC mechanism is not GSP. However, it is not obvious in general which forms of collusion would result in successful manipulation. In this section, we examine this issue in detail by asking two questions. First, at which utility profiles is MC GSP? Second, for a utility profile u at which MC is not GSP, exactly which groups can strategize successfully? We will show that MC fails to be GSP at u if and only if there exists a successful pair strategy or a specific simple kind of three-player strategy, and show exactly when these strategic opportunities arise.

Feigenbaum *et al.* (2001) give a low network complexity algorithm for the MC mechanism. The algorithm itself gives insights into the workings of the mechanism, so we describe it here.

Given a reported utility profile μ, the receiver set is the unique *maximal efficient set* of players. To find it, we recursively compute the *worth* $W_\mu(\beta)$ of each node $\beta \in N$ as

$$W_\mu(\beta) = \left(\sum_{\substack{\gamma \in \mathrm{Ch}(\beta) \\ W_\mu(\gamma) \geq 0}} W_\mu(\gamma) \right) - c(l) + \mu(\beta)$$

where $\mathrm{Ch}(\beta)$ is the set of children of β in the tree, $c(l)$ is the cost of the link connecting β to its parent node, and $\mu(\beta)$ denotes the total reported utility of the players residing at β. The worth of a node β measures the marginal amount that this node and the optimal subtree below it would contribute to the net worth of the chosen multicast tree, assuming that all the nodes above β had already been included. We can easily compute the worth at the leaves of the tree, then work our way up the tree to compute $W_\mu(\cdot)$ for the remaining nodes recursively. The maximal efficient set $R(\mu)$ is the set of all players i such that every node on the path from i to the root has nonnegative worth.

Another way to view this is as follows: The algorithm partitions the universal tree $T(P)$ into a forest $F(\mu) = \{T_1(\mu), T_2(\mu), \ldots, T_k(\mu)\}$. A link from $T(P)$ is included in the forest if and only if the child node has nonnegative worth. This is illustrated in Figure 5. $R(\mu)$ is then the set of players at nodes in the subtree $T_1(\mu)$ containing the root.

Once $F(\mu)$ has been computed, for each player $i \in R(\mu)$, define $Y(i, \mu)$ to be the node at or above i with minimum worth. The payment $x_i(\mu)$ of each player i is then defined as

$$\begin{aligned} x_i(\mu) = &\quad \max(0, \mu_i - W_\mu(Y(i,\mu))) \\ = &\quad \mu_i - \min(\mu_i, W_\mu(Y(i,\mu))) \qquad \forall i \in R(\mu) \\ x_i(\mu) = &\quad 0 \qquad \forall i \notin R(u) \end{aligned} \tag{7}$$

This completes our description of the algorithm. If there are multiple nodes at or above i with the same worth, we choose $Y(i, u)$ 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. For the same reason, we define $Y(i, \mu)$ for $i \notin R(\mu)$ to be the closest node at or above i that has zero or negative

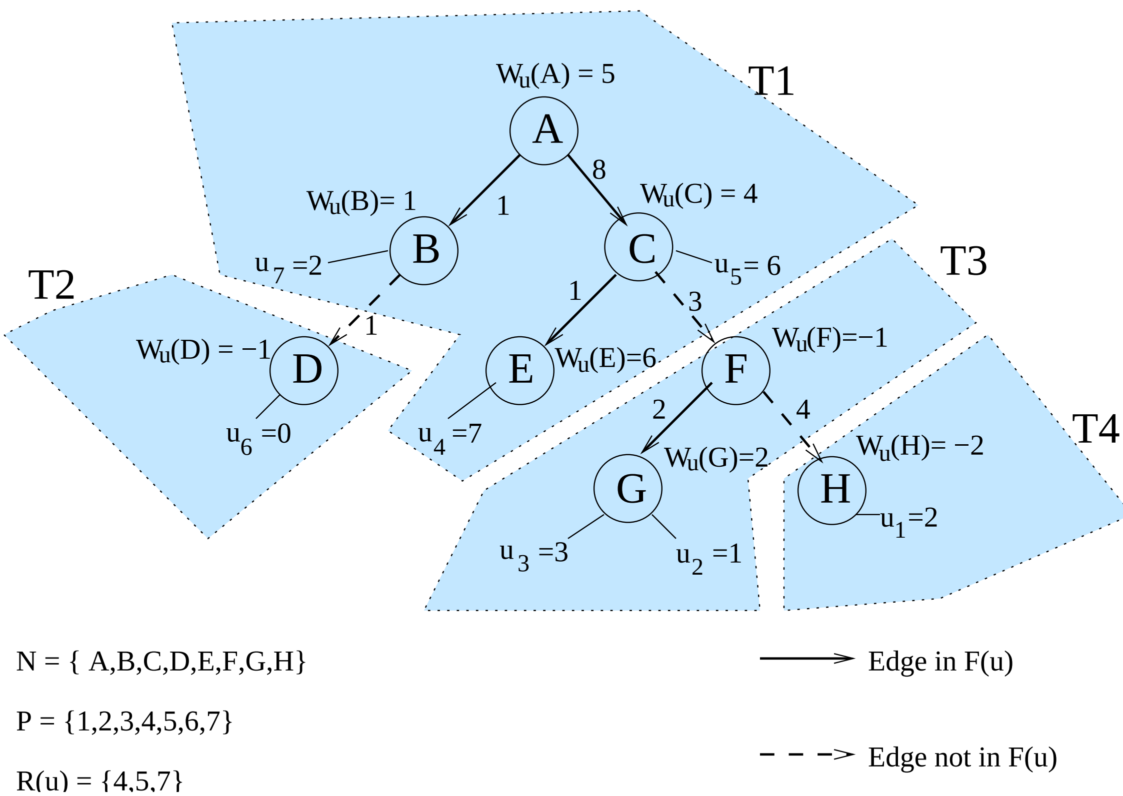

FIG. 5 Forest induced by MC mechanism

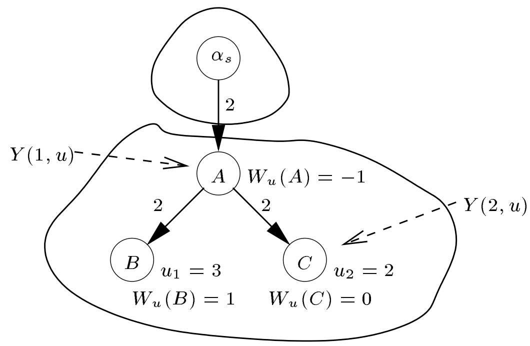

FIG. 6 An opportunity for a successful pair strategy. The circled groups of nodes represent the two components of $F(u)$.

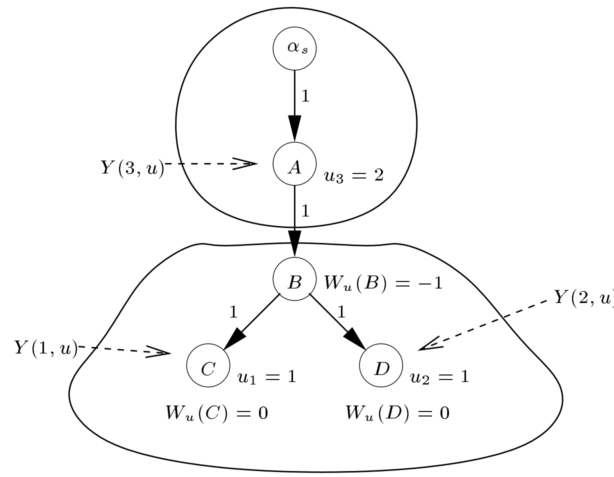

FIG. 7 An opportunity for a basic triple strategy. The circled groups of nodes represent the two components of $F(u)$.

worth. We will use this characterization of the receiver set and payments in terms of $F(u)$ and $Y(i,u)$ in our analysis of group strategies against the MC mechanism.

Before launching into the analysis of successful group strategies, we now give two specific instances which will serve as canonical examples of such strategies.

EXAMPLE 1. *Consider Figure 6. Here $R(u) = \emptyset$, so when both players report truthfully, both attain a welfare of zero. But if both lie by reporting $\mu_1 = \mu_2 = 4$, then $R(\mu) = \{1,2\}$ and each pays* 2*, so player* 2*'s welfare remains at zero while player* 1*'s rises to* 1*. Notice that $\mu' = (3,4)$ would not be a successful group strategy, because in that case player* 2 *would have to pay* 3.

EXAMPLE 2. *Consider Figure 7. In this case, there is no successful pair strategy. When all players report truthfully, only player* 3 *receives the transmission, paying $x_3(u) = 1$. Players* 1 *and* 2 *can never attain positive welfare, because whenever either one appears in the receiver set, she must at least pay for the link directly above her. If only one of these two players lies and she manages to join the receiver set, then she will effectively pay* 2 *for the two links above her. However, if players* 1 *and* 2 *both lie so that the reported utilities are $\mu = (2,2,2)$, then $R(\mu) = \{1,2,3\}$, and $x(\mu) = (1,1,0)$, so player* 3*'s welfare rises from* 1 *to* 2 *and the other two players remain at zero welfare.*

To better understand these examples, it helps to interpret the payment formula (7) in a different way, in terms of *cutoff utilities.*

DEFINITION 1. *Fixing μ_{-i}, the vector of reported utilities of all players aside from i, suppose there is some number $C_i(\mu_{-i})$ such that if $\mu_i < C_i(\mu_{-i})$ then $i \notin R(\mu)$, and if $\mu_i \geq C_i(\mu_{-i})$ then $i \in R(\mu)$ and i pays $x_i(\mu) = C_i(\mu_{-i})$. Then we call $C_i(\mu_{-i})$ the* cutoff utility *for i.*

LEMMA 10. *The cutoff utility always exists, and can be computed from $W_\mu(\cdot)$ as follows. For $i \in R(\mu)$, $C_i(\mu_{-i}) = \max\{0, \mu_i - W_\mu(Y(i,\mu))\}$. For $i \notin R(\mu)$, $C_i(\mu_{-i}) = \mu_i + L$, where*

$$L = \sum_{\alpha \in N_\mu(i)} |W_\mu(\alpha)|$$

and $N_\mu(i)$ denotes the set of nodes α at or above i such that $W_\mu(\alpha) < 0$.

Proof. We consider what happens if i raises or lowers her reported utility from μ_i, while the other players hold theirs fixed at μ_{-i}.

If $i \in R(\mu)$, then all of the nodes at or above i have non-negative worth, and

$$x_i(\mu) = \mu_i - W_\mu(Y(i,\mu)), \tag{8}$$

provided $W_\mu(Y(i,\mu)) \leq \mu_i$. Since $Y(i,\mu)$ is the node of minimum worth at or above i, player i can lower her reported utility by $W_\mu(Y(i,\mu))$ before the worth of this node drops below zero, causing i to leave the receiver set. As μ_i varies anywhere above this threshold, both terms on the right side of (8) change by the same amount, so i's payment remains unchanged. Thus, $C_i(\mu_{-i}) = \mu_i - W_\mu(Y(i,u))$. If $W_\mu(Y(i,\mu)) > \mu_i$, then i is in the receiver set and pays zero no matter what her reported utility, so $C_i(\mu_{-i}) = 0$.

If $i \notin R(\mu)$ then there is some sequence of nodes $\alpha_1, \ldots, \alpha_k$ $(k \geq 1)$ in this order along the path from i to α_s such that $W_\mu(\alpha_j) < 0$ for $j = 1, \ldots, k$. As i increases her reported utility from μ_i to $\mu_i + |W_\mu(\alpha_1)|$, the worth of each node from i to α_1 also increases by $|W_\mu(\alpha_1)|$. As i's reported utility rises another $|W_\mu(\alpha_2)|$, all the nodes from i to α_2 increase their worth by the same amount. Inductively, we see that i first joins the receiver set when her reported utility reaches $\mu_i + L$. As the reported utility increases further, all nodes at or above i increase in worth. Since $W_{(\mu_{-i},\mu_i+L)}(\alpha_k) = 0$, i's payment is $\mu_i + L$ when she first joins the receiver set. Further raising her reported utility does not change her payment. ∎

Armed with this new understanding of the payments, it is easy to detect when player j has the opportunity to lie to increase the welfare of another player i. If $i \in R(u)$, then i's cutoff utility is already at most u_i, so j needs to somehow lower it further. If $i \notin R(u)$, then j needs to somehow lower i's cutoff utility below u_i, so that i joins the receiver set and attains positive welfare. In both cases, this is possible if and only if $C_i(u_{-i}) > 0$ and j resides at or below node $Y(i,u)$, by Lemma 10. The tricky part is to figure out how j can do this without hurting herself – if j raises μ_j enough to help i, then j will necessarily join the receiver set. (This is because μ_j affects the worth of $Y(i,u)$ only if all nodes on the path from j to $Y(i,u)$ have strictly positive worth, and since i must be in the receiver set to have positive welfare, all nodes above $Y(i,u)$ must have non-negative welfare.) If $j \notin R(u)$, then $C_j(u_{-j}) > u_j$ so if the other players continue to report truthfully, then j will be charged more than u_j, incurring negative welfare. The pair strategy in Example 1 succeeds because A is the first negative worth node above player 2, and player 1 resides below A, so player 1 can "protect" player 2. The following definition and theorem formalize this observation.

Definition 2. *Define $P'(u) = \{i \in P | w_i(u) < u_i\}$. Then u is said to admit a* pair opportunity *if there are players i and j in the same component of $F(u)$ such that $i \in P'(u)$ and j resides at or below $Y(i,u)$.*

Theorem 4. *There exists a successful pair strategy for players i and j at utility profile u if and only if u admits a pair opportunity for i and j. In this case, let*

$$L = \sum_{\alpha \in N_u(i)} |W_u(\alpha)|,$$

where $N_u(i)$ is defined as in Lemma 10, and $L' > L$. Then $u|^{\{i,j\}}(u_i + L, u_j + L')$ is a successful pair strategy for i and j.

Proof. If direction: Suppose i and j are both in $R(u)$. Then j can raise her reported utility μ_j without hurting herself, since u_j is already at least as large as j's cutoff utility. Doing so will increase the worth of all nodes above j, including the entire path from $Y(i,u)$ to α_s. This will decrease the price that i pays, since $Y(i,u)$ is the bottleneck node determining that price.

Now suppose that i and j are in some other component of $F(u)$. We have $L > 0$, since otherwise $i \in R(u)$. If j raises her reported utility μ_j to $u_j + L$, then i's cutoff utility will be exactly u_i. Since j resides at or below $Y(i,u)$, further raising j's reported utility to $u_j + L'$ will make the worth of all nodes at or above i strictly positive, driving i's cutoff strictly below u_i. Since i and j are in the same component of $F(u)$, the root of this component is the first negative-worth node above each one (with respect to u). Therefore, if i also raises her reported utility to $u_i + L$, then j's cutoff will be exactly u_j. Thus, by colluding in this way, i and j will both be included in the receiver set, j's welfare will remain zero, and i's welfare will increase. (Note: if i does not reside below $Y(j,u)$, then it is impossible for i to lower j's cutoff utility any lower than u_j.)

Only if direction: If i and j have a successful pair strategy, then the collusion must cause one of them to improve her welfare, so without loss of generality we can assume it is i, hence $i \in P'(u)$. We will argue using i and j's cutoff utilities. Let $\hat{\mu}$ denote the successful pair strategy. Since all players k aside from i and j have $\hat{\mu}_k = u_k$, we can consider i's cutoff $C_i(\hat{\mu}_{-i})$ to depend only on $\hat{\mu}_j$, and similarly j's cutoff utility to depend only on $\hat{\mu}_i$.

If $i \in R(u)$, then the only way to improve i's welfare is to lower her cutoff utility, which j can do only if she resides at or below $Y(i,u)$. If $i \notin R(u)$, then i's cutoff is initially above u_i and must be lowered strictly below u_i.

In the latter case, we know that $W_u(Y(i,u)) \le 0$, and there is some node of strictly negative worth above i. Since the strategy $\hat{\mu}$ improves i's welfare, we know by Lemma 10 that $W_{u|^j\hat{\mu}_j}(\alpha) > 0$ for all nodes α at or above i. In order for j to cause i to connect and pay less than u_i, $\hat{\mu}_j$ must be high enough to make all of the nodes at and above $Y(i,u)$ have strictly positive worth with respect to $u|^j\hat{\mu}_j$. In particular, j must reside at or below $Y(i,u)$, since otherwise μ_j does not affect the worth of $Y(i,u)$. Moreover, all nodes at or above j must have strictly positive worth with respect to $u|^j\hat{\mu}_j$. If $\hat{\mu}_i \ge u_i$, then these nodes also have strictly positive worth with respect to $\hat{\mu}$. If $\hat{\mu}_i < u_i$, then $W_{\hat{\mu}}(\alpha)$ may be lower than $W_{u|^j\hat{\mu}_j}(\alpha)$ for some of the nodes α at or above i, but all of these still must have non-negative worth, since otherwise i would not be in the receiver set, so would not attain positive welfare. Therefore, j is also in the receiver set $R(\hat{\mu})$.

Suppose i and j reside in different components of $F(u)$. Since $Y(i,u)$ is in i's component and j resides below this node, we know j's component lies below i's. Let α denote the root of j's component in $F(u)$. We just argued that $j \in R(\hat{\mu})$. But i's reported utility has no effect on $W_\mu(\alpha)$, which is negative when $\mu = u$ (i.e. when all utilities are reported truthfully). Thus, j's cutoff $C_j(u_{-j}|^i\hat{\mu}_i)$ is at least $u_j + |W_u(\alpha)|$, so $w_j(\hat{\mu}) < 0$, which contradicts the assumption that $\hat{\mu}$ is a successful strategy. Thus, i and j must reside in the same component of $F(u)$. ∎

In Example 2, u admits no pair opportunity, so by Theorem 4, there is no successful pair strategy. Yet, the three players can still collude successfully. This is because players 1 and 2 are in the same component of $F(u)$, hence can protect each other from incurring negative welfare, and reside below $Y(3,u)$, hence can help player 3. It turns out that this situation is the only other way in which a succussful group strategy can arise. We now formalize this result.

DEFINITION 3. *Suppose $i \in P'(u)$ and there is a component of $F(u)$, distinct from i's, that lies below $Y(i,u)$ and contains players j and k. Then we say that u presents a* triple opportunity *for $\{i,j,k\}$.*

DEFINITION 4. *Suppose u presents a triple opportunity for $\{i,j,k\}$. Let $L = \sum_{\alpha \in N_u(j)} |W_u(\alpha)|$ and $L' > L$. Then the strategy $\mu_{\{i,j,k\}} = (u_i, u_j + L', u_k + L')$ is called a* basic triple strategy *for $\{i,j,k\}$.*

Note that if either j resides at or below $Y(k,u)$ and $k \in P'(u)$ or vice versa, then j and k have a successful pair strategy, by Theorem 4. But even if neither of those conditions holds, the next claim shows that the basic triple strategy still succeeds.

CLAIM 1. *Suppose u presents a triple opportunity for $\{i, j, k\}$. Then the corresponding basic triple strategies are successful group strategies for $\{i, j, k\}$.*

Proof. By assumption, $j, k \notin R(u)$, so these players intially have zero welfare. Since j and k are in the same component of $F(u)$, we have $N_u(j) = N_u(k)$, the lowest node in this set being the root of this component. Thus, $W_{u|^j(u_j+L')}(\alpha) > 0$ for all $\alpha \in N_u(k)$, so $C_k(u_{-k}|^j(u_j + L')) \leq u_k$. Similarly, $C_j(u_{-j}|^k(u_k + L')) \leq u_j$. Thus, under the group strategy $\hat{\mu} = u|^{\{j,k\}}(u_j + L', u_k + L')$, j and k at least maintain their zero welfare.

Since $N_u(i) \subset N_u(j)$, $L' > L$, and j resides below $Y(i, u)$, we have $W_{u|^j(u_j+L')}(\alpha) > 0$ for all α at or above i. Increasing μ_k as well only increases the worths, so $i \in R(\hat{\mu})$ and $x_i(\hat{\mu}) < u_i$, so $w_i(\hat{\mu}) > 0$. Thus, if $i \notin R(u)$, the strategy improved i's welfare. If $i \in R(u)$, the strategy improves i's welfare because $L' > L$ guarantees $W_{\hat{\mu}}(Y(i, u)) > W_u(Y(i, u))$, so j and k's increased reported utilities lowered i's cutoff utility. ∎

DEFINITION 5. *If I is a set of players, we say that $i \in I$ is* minimal with respect to $F(u)$ *if there is no other player $j \in I$ located in a different component of $F(u)$ such that i's component of $F(u)$ contains any nodes above j. That is, if we contract components of $F(u)$, there is no player $j \in I$ who resides strictly below i.*

THEOREM 5. *If some coalition S has a successful group strategy at u, then either there exist players $i, j \in S$ with a successful pair strategy, or there exist players $i, j, k \in S$ with a successful basic triple strategy. Conversely, if there is a pair or triple of players with a successful pair or basic triple strategy, then every set of players S containing that pair or triple has a successful group strategy.*

Proof. Denote the group strategy by $\hat{\mu}$. Since the strategy succeeds, the set $I = \{i \in S : w_i(\hat{\mu}) > w_i(u)\}$ is non-empty. Select some player $i \in I$ that is minimal with respect to $F(u)$. Since i benefitted from the strategy $\hat{\mu}$ as compared to u, the manipulations of the other players in S must have reduced i's cutoff utility, and $i \in R(\hat{\mu})$. Thus, S must include some other player h residing at or below $Y(i, u)$, such that $\hat{\mu}_h > u_h$ and the worth $W_{\hat{\mu}}(\alpha)$ of each node α between h and $Y(i, u)$ is strictly positive. Let J denote the set of all such players in S, and select some player $j \in J$ that is minimal with respect to $F(u)$. Note that $j \in R(\hat{\mu})$, since (under $\hat{\mu}$) it has a positive welfare path to $Y(i, u)$, and $i \in R(\hat{\mu})$. If j lies in the same component of $F(u)$ as i, then i and j have a successful pair strategy, by Theorem 4. Otherwise, j's component lies below i's.

Suppose j's component contains no other players in S, and let β denote the root of that component. Because j is minimal in J, even if there is some player $k \in S$ who resides in a component strictly below j's, then her misreported utility has no positive effect on the welfares computed at the nodes between j and β. This is because $j \in J$ but $k \notin J$, so there is some node α at or above k but not at or above j such that $W_{\hat{\mu}}(k) \leq 0$. Thus, j's cutoff utility is at least $u_j + |W_u(\beta)| > u_j$. So $w_j(\hat{\mu}) < 0$, which contradicts the assumption that $\hat{\mu}$ is a successful strategy. Thus, j's component of $F(u)$ contains some other player $k \in S$. Thus, by Claim 1, there is a successful basic triple strategy available for $\{i, j, k\}$.

The converse holds because the successful pair and basic triple strategies involve raising reported utilities, which can never increase the cutoffs for the other players. So if each other player h reports her true utility u_h, then her welfare does not decrease, so she can be considered to be part of the strategizing set. ∎

Summarizing the previous results, we have the following characterization of utility profiles for which the MC mechanism is GSP.

THEOREM 6. *The MC mechanism is GSP at u if and only if the following condition holds for each $i \in P'(u)$: there is no player j in the same component of $F(u)$ as i such that j resides at or* *below* $Y(i,u)$*, and there is no pair of players* j *and* k *residing in the same component as each other and below* $Y(i,u)$.

We have now completely characterized the utility profiles u at which the MC mechanism is GSP, shown that the minimal sets of players who can successfully manipulate the mechanism are pairs and triples, and shown that every set of players containing such a pair or triple has a successful group strategy. The question remains, what do successful group strategies look like in general? The following theorem shows that every group strategy can be converted into a "canonical" one.

THEOREM 7. *If* S *has a successful strategy* $\hat{\mu}$ *at true utility profile* u*, then* $S' = S \cap R(\hat{\mu})$ *has a successful strategy* $\hat{\mu}'$ *such that*

- $S' \subseteq R(\hat{\mu}')$
- $w_i(\hat{\mu}') \geq w_i(\hat{\mu})$ *for all* $i \in P$
- $\hat{\mu}'_i \geq u_i$ *for all* $i \in S'$

Proof. For each $i \in S - R(\hat{\mu})$ such that $\hat{\mu}_i > u_i$, set $\hat{\mu}'_i = u_i$. This may raise the cutoff utilities for some players outside $R(\hat{\mu})$, but these had zero welfare anyway. It has no effect on the computed welfares for nodes in the root component of $F(\hat{\mu})$, hence no effect on the receiver set or on the price charged to any node in $R(\hat{\mu})$. Now set $\hat{\mu}'_i = u_i$ for each remaining $i \in S - R(\hat{\mu})$. This can only decrease the prices paid by players in $R(\hat{\mu})$. It may also expand the receiver set, but that will not cause any of the new recipients to get negative welfare because we have already gotten rid of all the players in $S - R(\hat{\mu})$ who exaggerated their utilities. Now set $\hat{\mu}'_i = u_i$ for each $i \in S \cap R(\hat{\mu})$ such that $\hat{\mu}_i < u_i$. This does not change the receiver set, and can only decrease the prices paid by the receivers. Since S had some player who benefitted from the strategy and this player is in $R(\hat{\mu})$, we can now throw all players in $S - R(\hat{\mu})$ out of the strategizing set, leaving us with S'. ∎

Thus, in some sense, the "interesting" group strategies are the ones in which the players who misreport their utilities only exaggerate them, and all of them end up in the receiver set.

5. OPEN PROBLEMS

The results in Section 3 above lead naturally to the following question about the SH mechanism: Is there an approximation to the SH mechanism with the same worst-case network complexity as SSF? That is, is there a mechanism with the same worst-case network complexity that also achieves constant-factor bounds on the budget deficit (or surplus) and on the worst-case welfare loss?

The results in Section 4 above 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 GSP. We note, however, that some form of *tolerable manipulability* might be acceptable. That is, one may be quite willing to deploy a mechanism that is known not to be GSP 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.

Additional open problems about multicast cost sharing in particular and distributed algorithmic mechanism design in general can be found in Feigenbaum and Shenker (2002).

REFERENCES

Adler, M. and Rubenstein, D. (2002). "Pricing Multicast in More Practical Network Models," in *Proceedings of the 13th Symposium on Discrete Algorithms*, ACM Press/SIAM, New York/Philadelphia, pp. 981–990.

Archer, A. and Tardos, E. (2002). "Frugal path mechanisms," in *Proceedings of 13th Symposium on Discrete Algorithms*, ACM Press/SIAM, New York/Philadelphia, pp. 991–999.

Ballardie, A., Francis, P., and Crowcroft, J. (1993). "Core Based Trees (CBT)," in *Proceedings of Sigcomm '93*, ACM Press, New York, pp. 85–95.

Clarke, E. H. (1971). "Multipart Pricing of Public Goods," *Public Choice* **1971**, pp. 17–33.

Deering, S. and Cheriton, D. (1990). "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems* **8**, pp. 85–110.

Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, C., and Wei, L. (1996). "The PIM architecture for wide-area multicast routing," *ACM/IEEE Transactions on Networking* **4**, pp. 153–162.

Feigenbaum, J., Krishnamurthy, A., Sami, R., and Shenker, S. (2003). "Hardness Results for Multicast Cost Sharing," *Theoretical Computer Science* **304**, pp. 215–236.

Feigenbaum, J., Papadimitriou, C., and Shenker, S. (2001). "Sharing the cost of multicast transmissions," *Journal of Computer and System Sciences* **63**, pp. 21–41. (Special issue on Internet Algorithms.)

Feigenbaum, J. and Shenker, S. (2002). "Distributed Algorithmic Mechanism Design: Recent Results and Future Directions," in *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM Press, New York, pp. 1–13.

Fiat, A., Goldberg, A., Hartline, J., and Karlin, A. (2002). "Generalized Competitive Auctions," in *Proceedings of the 34th Symposium on Theory of Computing*, ACM Press, New York, pp. 72–81.

Goldberg, A., Hartline, J., and Wright, A. (2001). "Competitive auctions and digital goods," in *Proceedings of 12th Symp. on Discrete Algorithms*, ACM Press/SIAM, New York/Philadelphia, pp. 735–744,

Green, J. and Laffont, J-J. (1979). **Incentives in Public Decision Making**, North Holland, Amsterdam.

Groves, T. (1973). "Incentives in Teams," *Econometrica* **41**, pp. 617–631.

Hershberger, J. and Suri, S. (2001). "Vickrey prices and shortest paths: What is an edge worth?" in *Proceedings of the 42nd Symposium on the Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, pp. 129–140.

Holbrook, H. and Cheriton, D. (1999). "IP multicast channels: EXPRESS support for large-scale single-source applications," in *Proceedings of Sigcomm '99*, ACM Press, New York, pp. 65–78.

Herzog, S., Shenker, S., and Estrin, D. (1997). "Sharing the 'cost' of multicast trees: An axiomatic analysis," *ACM/IEEE Transactions on Networking* **5**, pp. 847–860.

Jain, K. and Vazirani, V. (2001). "Applications of approximation to cooperative games," in *Proceedings of 33rd Symposium on the Theory of Computing*, ACM Press, New York, pp. 364–372.

Lavi, R. and Nisan, N. (2000). "Competitive analysis of incentive compatible on-line auctions," in *Proceedings of 2nd Conference on Electronic Commerce (EC-00)*, ACM Press, New York, pp. 233–241.

Mas-Colell, A., Whinston, M., and Green, J. (1995). **Microeconomic Theory**, Oxford University Press, New York, 1995.

Megiddo, N. (1978). "Computational complexity of the game theory approach to cost allocation for a tree," *Mathematics of Operations Research* **3**, pp. 189–196.

Mitchell, J. and Teague, V. (2002). Private communication.

Monderer, D. and Tennenholtz, M. (1999). "Distributed Games: From Mechanisms to Protocols," in *Proceedings of 16-th National Conf. on Artificial Intelligence (AAAI 1999)*, AAAI Press, Menlo Park, pp. 32–37.

Moulin, H. (1999). "Incremental cost sharing: characterization by strategyproofness," *Social Choice and Welfare* **16**, pp. 279–320.

Moulin, H. and Shenker, S. (2001). "Strategyproof Sharing of Submodular Costs: Budget Balance Versus Efficiency," *Economic Theory* **18**, pp. 511–533.

Nisan, N. and Ronen, A. (2000). "Computationally feasible VCG mechanisms," in *Proceedings of 2nd Conference on Electronic Commerce (EC-00)*, ACM Press, New York, pp. 242–252.

Nisan, N. and Ronen, A. (2001). "Algorithmic mechanism design," *Games and Economic Behavior* **35**, pp. 166–196.

Parkes, D. (1999). "iBundle: An efficient ascending price bundle auction," in *Proceedings of 1st Conference on Electronic Commerce (EC-99)*, ACM Press, New York, pp. 148–157.

Parkes, D. and Ungar, L. (2000). "Iterative combinatorial auctions: Theory and practice," in *Proceedings of 17-th National Conf. on Artificial Intelligence (AAAI 2000)*, AAAI Press, Menlo Park, pp. 74–81.

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

Roughgarden, T. and Tardos, E. (2002). "How Bad is Selfish Routing?" *Journal of the ACM* **49**, pp. 236–259.

Sandholm, T. (1999). "Distributed rational decision making," in G. Weiss, editor, *Multiagent systems: A Modern Introduction to Distributed Artificial Intelligence*, MIT Press, Cambridge, MA, pp. 201–258.

Vickrey, W. (1961). "Counterspeculation, Auctions and Competitive Sealed Tenders," *Journal of Finance* **1961**, pp. 8–37.

Wellman, M. (1993). "A market-oriented programming environment and its applications to distributed multicommodity flow problems," *Journal of AI Research* **1**, pp. 1–23.

Wellman, M., Walsh, W., Wurman, P., and Mackie-Mason, J. (2001). "Auction protocols for decentralized scheduling," *Games and Economic Behavior* **35**, pp. 271–303.

APPENDIX: ALTERNATIVE PROOF THAT **SF** IS GSP

We now present a proof of Theorem 1 that builds on previous results in the mechanism-design literature. Moulin and Shenker (Moulin and Shenker, 2001; Moulin, 1999) discussed a family of budget balanced mechanisms based on *cross-monotonic cost-sharing* functions and proved them to be group strategyproof.

Consider a function $f : 2^P \mapsto \Re^P_{\geq 0}$. This function is **cross-monotonic** if, $\forall i \in S, \forall T \subseteq P$, $f_i(S \cup T) \leq f_i(S)$. In addition, we require that $f_i(S) \geq 0$ and that, $\forall j \notin S$, $f_j(S) = 0$.

Then, the corresponding cross-monotonic mechanism $M_f = (\sigma(\mu), x(\mu))$ is defined as follows: The receiver set $R(\mu)$ is the unique largest set S for which $f_i(S) \leq \mu_i$. This is well defined, because, if sets S and T each satisfy this property, then cross-monotonicity implies that $S \cup T$ satisfies it. Let $\sigma_i(\mu)$ be the indicator vector for $R(\mu)$ and $x_i(\mu) = f_i(R(\mu))$.

LEMMA 11. *M_f is group strategyproof.*

This was proved by Moulin (1999) in the context of budget-balanced mechanisms, but his proof extends directly to all cross-monotonic mechanisms.

Let $\mathcal{M} = \{M_f \mid \text{f is cross-monotonic}\}$ be the set of cross-monotonic mechanisms. We give an alternate characterization of the mechanisms in this set that does not explicitly use the cost-sharing function in the construction of the receiver set.

THEOREM 8. *Fix the tree and the costs $c(l)$, and let $U = \Re^P_{\geq 0}$ be the space of possible utility profiles. A mechanism $M = (\sigma(\mu), x(\mu))$ is in $\mathcal{M}$ iff it satisfies the following properties:*

1. *Consumer sovereignty: $\exists B$ such that, for all i, for all $u \in U$ such that $u_i \geq B$, $i \in R(u)$.*
2. *Monotonicity of receiver set: if u,u' are utility profiles such that, for all i, $u_i \leq u'_i$, then $R(u) \subseteq R(u')$.*
3. *Let $u, u' \in U$ be utility profiles such that $R(u) = R(u')$. Then, $x_i(u) = x_i(u')$, for all i. In other words, the cost shares are a function of the receiver set alone, and we can use the notation $x_i(S)$ to indicate the payment of player i when the receiver set is S.*
4. *$x_i(.)$ is cross-monotonic on the space of receiver sets, i.e., if $S \subseteq S'$, then $x_i(S') \leq x_i(S)$, for all $i \in S$.*
5. *For any $S \subseteq P$, let $U(S) = \{u \in U \mid R(u) = S\}$. Then, $U(S)$ is closed under the pointwise minimum operation: If $u, u' \in U(S)$, and u'' is defined by $u''_i = \min(u_i, u'_i)$, then $u'' \in U(S)$.*
6. *$x_i(S) = \min_{u \in U(S)} u_i$*

Proof.

If direction: Consider a mechanism M $= (\sigma(\mu), x(\mu))$, and let $R(\mu)$ be the receiver set corresponding to $\sigma(\mu)$. Assume M satisfies properties 1-6.

Properties 5 and 6 say that this mechanism partitions the utility space into "regions" $U(S)$ corresponding to every receiver set S. Every region has a unique minimum point $\overline{u}(S)$ defined by

$$\begin{aligned} \overline{u}_i(S) &= x_i(S) \quad \text{if } i \in S \\ \overline{u}_i(S) &= 0 \quad \text{if } i \notin S \end{aligned}$$

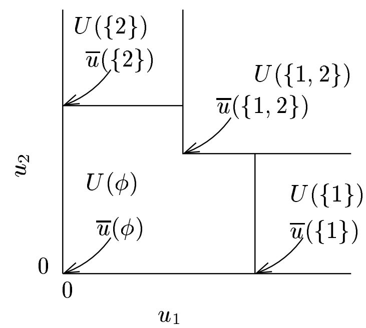

FIG. 8 Partition of utility space for a two-player mechanism

Consider the utility profile u^S given by

$$\begin{aligned} u_i^S &= B \quad \text{if } i \in S \\ u_i^S &= 0 \quad \text{if } i \notin S \end{aligned}$$

Then, by property 1, $R(u^S) \supseteq S$.

Now, consider the cost-sharing function f defined by

$$f_i(S) \stackrel{def}{=} \overline{u}_i(R(u^S)) \tag{9}$$

For any $S' \supseteq S$, we know by property 2 that $R(u^{S'}) \supseteq R(u^S)$. Then, by property 4, it follows that, for all $i \in S$, $f_i(S') \leq f_i(S)$, and hence f is a cross-monotonic cost-sharing function. It only remains to be shown that, for all μ, $R(\mu)$ is the unique largest set S for which

$$\forall i \in S \; f_i(S) \leq \mu_i \tag{10}$$

$R(\mu)$ satisfies equation (10), because $\mu_i \geq \overline{u}_i(R(\mu)) = x_i(R(\mu))$. Consider any set T that also satisfies this condition. Then, by assumption,

$$\begin{aligned} \forall i \;\; f_i(T) &\leq \mu_i \\ \Longrightarrow \overline{u}_i(R(u^T)) &\leq \mu_i \\ \Longrightarrow R(\overline{u}(R(u^T))) &\subseteq R(\mu) \quad \text{by property 2} \end{aligned}$$

We note that $R(\overline{u}(R(u^T))) = R(u^T) \supseteq T$, and so it follows that $T \subseteq R(\mu)$. Because this is true for all such T, $R(\mu)$ must be the largest set for which equation (10) is satisfied.

Only If direction: Consider any cross-monotonic mechanism M_f. Let $B = \max_i f_i(\{i\})$. Then, it is easy to verify that each of the properties above is satisfied. ∎

Figure 8 illustrates one possible partition for two players. The mechanisms in $\mathcal{M}$ are completely characterized by the points $\overline{u}(S)$, over all $S \subseteq P$.

THEOREM 9. *Mechanism SF* $\in \mathcal{M}$.

Proof. We show that mechanism SF has all the properties listed in Theorem 8.

Property 1: Let B be the maximum cost of a path from any player to the root. Then, if $u_i \geq B$, $i \in \tilde{R}(u)$.
Property 2: The monotonicity of mechanism SF was proved in Lemma 3.
Property 3: Suppose $R(u) = R(u') = S$. Then, using Lemma 6 repeatedly, we can show that $x_i(u) = x_i(u^S)$, where u^S is defined as in the proof of theorem 8. Similarly, it also follows that $x_i(u') = x_i(u^S)$, and so $x_i(u) = x_i(u')$. Hence this property is valid, and we can refer to the payment function as $x_i(S)$.
Property 4: For receiver sets S and S' such that $S \subseteq S'$, consider the utility profiles u^S and $u^{S'}$. The conditions of Lemma 3 apply, and so $x_i(S) = x_i(u^S) < x_i(u^{S'}) = x_i(S')$.
Properties 5 and 6: For any utility profile u, with receiver set $R(u) = S$, consider the utility profile $\overline{u}$ defined by

$$\begin{aligned} \overline{u}_i &= x_i(S) \quad \text{if } i \in S \\ \overline{u}_i &= 0 \quad \text{if } i \notin S \end{aligned}$$

Note that it is sufficient to show that $R(\overline{u}) = S$ to prove both properties 5 and 6.

We can prove that $R(\overline{u}) = R(u) = S$ by increasing the elements of $\overline{u}_i$ to u_i one at time and showing that the receiver set remains the same at each step. For $i \notin S$, we can show this by induction on $\tilde{n}_l(.)$, as in Lemma 4. For $i \in S$, this follows directly from Lemma 6. ∎