

Mechanisms for Multi-Level Marketing

Yuval Emek* Ron Karidi† Moshe Tennenholtz‡ Aviv Zohar§

Abstract

Multi-level marketing is a marketing approach that motivates its participants to promote a certain product among their friends. The popularity of this approach increases due to the accessibility of modern social networks, however, it existed in one form or the other long before the Internet age began (the infamous Pyramid scheme that dates back at least a century is in fact a special case of multi-level marketing). This paper lays foundations for the study of reward mechanisms in multi-level marketing within social networks. We provide a set of desired properties for such mechanisms and show that they are uniquely satisfied by geometric reward mechanisms. The resilience of mechanisms to false-name manipulations is also considered; while geometric reward mechanisms fail against such manipulations, we exhibit other mechanisms which are false-name-proof.

1 Introduction

Social networks are everywhere: our e-mail and phone address books, our family relatives, and our business connections, all define either explicit or implicit social networks. Social networks have existed long before the Internet, but their recent web-based form, as exhibited by companies like Facebook, Twitter, or LinkedIn, made them more tangible. In their new manifestation, social networks have become an attractive playground for viral marketing: the dream of any marketer is that her products will be promoted via “word of mouth” (which relies on social networks). In order to make that dream a reality, various forms of marketing have been advocated. The so-called affiliate marketing, direct marketing, and multi-level marketing all refer to (overlapping) approaches that facilitate viral marketing. In this paper, we shall adhere to the term multi-level marketing, as it seems to be the least restrictive one.

The fundamental idea behind multi-level marketing is that Alice, who already purchased the product, is rewarded for *referrals*, i.e., for purchases made by Bob as a result of Alice’s promotion. The *reward mechanism* associated with multi-level marketing may take various forms. In particular,

*ETH Zurich. yuval.emek@tik.ee.ethz.ch

†Microsoft Israel R&D Center. ronkar@microsoft.com

‡Microsoft Israel R&D Center and Technion–Israel Institute of Technology. moshet@microsoft.com

§Microsoft Research Silicon Valley, Mountain View, CA. avivz@microsoft.com

Alice may be rewarded for both purchases made by Bob and for Bob’s own referrals in a recursive manner.

The potential to accumulate small rewards from each person to a sizable sum is important as it allows advertisers to attract early adopters and trendsetters that are of great value to them. On the downside, the possibility of gathering a large sum has also inspired more illicit versions of multi-level marketing, namely *pyramid schemes*. These illegal¹ mechanisms, essentially based on the notion of indirect referrals, are not intended to promote a real product, but rather to collect money from the social network, although sometimes a product is used in an attempt to cover the nature of the pyramid scheme and bypass legal restrictions. In these cases, customers seldom enjoy the actual product being promoted (when a product is being promoted), but are only participating in the (usually false) hope of getting rewards from recruiting others.

Needless to say that selecting an appropriate reward mechanism is inherent to the design of a successful multi-level marketing scheme. Interestingly, despite the popularity of work on information spreading and influence in social networks (see, e.g., the survey in [11]) the study of reward mechanism design in that context has been almost completely neglected. Such study is the main subject of the current paper.

Consider for example the following basic *coupon driven* scheme. Upon purchase of the product, Alice is given coupons that she can distribute among her friends. Then, for any purchase made by Bob in which Alice’s coupon is used, Alice is rewarded with appropriate rebates on future purchases. This scheme and similar ones, are easy to implement and have become quite standard in our daily life. Note that the coupon driven scheme does not exploit indirect referrals: Alice is not rewarded by purchases made with Bob’s coupons or with the coupons of Bob’s referrals.

Reward mechanisms that exploit indirect referrals used to be difficult to implement as they require some central authority that keeps track of the referral structure. Information technology has made this task much easier. Consider for example the setting in which Alice promotes a product by publishing a link to the seller’s web-site in her blog or Facebook page. Bob can buy the product by clicking on that link; together with the actual product, Bob receives a link to the seller’s web-site that he can also publish in his blog or Facebook page. The seller’s web-site can easily identify Bob as a buyer that followed Alice’s link.² This way a complete record of direct and indirect referrals can clearly be maintained. This ease of implementation makes reward mechanisms that take indirect referrals into account even more appealing than they have been before.

Are indirect referrals really that important? We believe so. To demonstrate their significance, suppose that Bob is a rock music authority and that following Alice’s promotion, he downloads

¹ The current paper does not take legal issues into consideration. In particular, our analysis will not make the distinction between legitimate multi-level marketing and pyramid schemes.

² This can be implemented by associating Alice’s link with a unique identifier. In the current paper we abstract away this technical issue.

a new rock song. If Bob recommends this song in his blog, and consequently many other users download this song, then Alice certainly played a major role in the promotion process, even if she only had a few direct referrals. A reward mechanism that depends only on direct referrals is therefore bound to miss the bigger picture.

The referrals tree model. There are many possible ways to take the social network that forms the basis of the referral process into account. In principle, one may wish to consider the times at which promoting messages were sent from one user to another, to consider referrals that were not followed up by a purchase of the product being promoted, or even to consider the social links along which a referral was *not* made. However, all of this information may not be available to the original seller.³ We therefore take the straightforward approach of looking only at the structure of successful referrals. For each buyer, we mark only a single referrer for introducing the product to her (in reality, this would typically be specified at the time of purchase). The induced structure of referrals forms a collection of directed trees, each rooted at a node that corresponds to some buyer that has purchased the product directly from the seller. We shall refer to this tree collection as the *referrals forest*, denoted \mathcal{T} , and to the rooted trees in \mathcal{T} as the *referrals trees*. We find the assumption that \mathcal{T} can be maintained by the seller sufficiently weak.

It should be clarified that the referrals forest corresponds to a single multi-level marketing campaign (typically associated with a single product). Moreover, social network users that did not purchase the product are not represented in \mathcal{T} even if some of their friends attempted at promoting the product to them. For ease of presentation, we assume that \mathcal{T} is fully known when the rewards are to be distributed, although all the mechanisms explored in this paper are also suited for incremental payments performed online. It will also be convenient to identify the buyers with their corresponding nodes in \mathcal{T} , denoting the reward of (the buyer corresponding to) node u under the referrals forest \mathcal{T} by $R_{\mathcal{T}}(u)$.

Constraints on the reward mechanism. The reward mechanism is essentially a function that maps the referrals forest \mathcal{T} to the non-negative real rewards of its nodes. However, not every such function should be considered; specifically, we impose three constraints on the reward mechanisms. The first one is the *subtree constraint*: $R_{\mathcal{T}}(u)$ is uniquely determined by \mathcal{T}_u , namely, by the subtree of \mathcal{T} rooted at u . This is sensible, as each user u can really be credited only for bringing in users she promoted the product to, either directly (the children of u in \mathcal{T}) or indirectly (lower level descendants of u). Moreover, a dependence of $R_{\mathcal{T}}(u)$ on the position of u within \mathcal{T} (rather than on \mathcal{T}_u only) may result in an undesirable behavior on behalf of u : in some cases u is better off delaying the purchase of the product after receiving a referral in hope for a “better” offer, i.e., for a referral

³ In some social networks such as Facebook there is often more explicit knowledge of social connections, but general referral systems do not necessarily have all the information about the underlying social structure and may not be able to track messages in the network.

that would place u in a better position within \mathcal{T} .

One of the consequences of the subtree constraint is that there is no point in dealing with the referrals forest \mathcal{T} in full, but rather focus on trees which are rooted at the nodes whose reward we are trying to calculate. In other words, the reward mechanism is completely specified by the function $R(T)$ that maps the rooted tree T to the non-negative real reward of its root (which may be an internal node within the whole referrals forest).

The second constraint that we impose on the reward mechanism is the *budget constraint*: the seller is willing to spend at most a certain fraction $\phi \leq 1$ of her total income on rewarding her buyers for referrals. Given that the price of the product is π , this means that the total sum of rewards given to all nodes is at most $\phi \cdot \pi |\mathcal{T}|$. We assume without loss of generality that π and ϕ are scaled so that $\phi \cdot \pi = 1$. Thus,

$$\sum_{u \in \mathcal{T}} R(T_u) \leq |\mathcal{T}| .$$

The third constraint is the *unbounded reward constraint*: there is no limit to the rewards one can potentially receive even under the assumption that each user has a limited circle of friends in the underlying social network (imposing a limited number of direct referrals). Formally, the unbounded reward constraint dictates that there exists some positive integer d (a property of the reward mechanism) such that for every real R , there exists some tree T of maximum degree d (i.e., every node has at most d children) such that $R(T) \geq R$. In particular, this constraint implies that the reward mechanisms we consider must take indirect referrals into account.

Our results. We begin our exploration of reward mechanisms with a well known family of mechanisms, namely, *geometric reward mechanisms*. Under these mechanisms, the contribution of a node to the rewards of its ancestors in the referrals tree decreases exponentially (by a fixed factor) with the distance from these ancestors. Three desired properties of geometric reward mechanisms are listed: additivity, child-dependence, and depth-level-dependence. We show that these three properties fully characterize the family of geometric mechanisms in the sense that any mechanism that satisfies all three properties must be a geometric mechanism. (This may explain why pyramid schemes typically rely on geometric reward mechanisms or some close variant of them.) We go on to show that none of the properties is redundant: if any one of the three is left out, alternative reward mechanisms can be found that possess the remaining three.

We then look at one more important property of reward mechanisms, namely, resilience to false-name manipulations (a.k.a. Sybil attacks). The geometric mechanism family turns out to be susceptible to manipulations by users that can create false identities. In fact, we show that mechanisms that are resilient to false-name manipulations cannot guarantee a user some constant fraction of the reward of even its least influential child. Moreover, it turns out that even if one replaces the child-dependence and depth-level-dependence properties with the much weaker monotonicity

property, resilience to false-name manipulations is still impossible. On the positive side, we present and analyze two reward mechanisms that maintain resilience to false-name manipulations; the two mechanisms differ in their level of resilience and ease of implementation.

Related work. The general idea of diffusion of opinions and conventions in societies has for long been a topic of study in the social sciences [15, 9] and got attention by game theorists (e.g., [20]) and AI researchers (e.g., [16]) among others, quite a while ago. The effects of the social structure on emergent behavior and norms has also been studied, e.g., in [14, 17].

The more explicit algorithmic questions that arise when one considers an endeavor such as viral marketing have been posed more recently. The original question of how to select a good set of influential users has appeared in a seminal paper by Domingo & Richardson [8], and has later been explored (with various related models) in a series of papers, e.g. [7] and many other following works [10, 3, 5]. These generally assume that the spread of information in the social network occurs through some contagion model (i.e., that a user is more likely to be “infected” with an idea if more of her neighbors are) in which users do not explicitly exert effort. The rigorous study of incentive design for facilitating diffusion or product adoption was typically left without proper explicit treatment, and all works we are aware of in the context of viral marketing do not try to influence the amount of effort exerted to spread the information further.

The issue of incentives in social networks has however received attention in other particular contexts. For example, fair distribution of costs/gains of members in a network, using standard power indices from cooperative game theory, such as the Shapley and Banzhaf values has been a subject of study (see e.g., [13, 4, 2]). However, these do not provide a general rigorous study of reward mechanisms for social distribution. Kleinberg & Raghavan [12] consider a setting that is perhaps the most similar in spirit to our own, in which they elicit effort from agents that forward queries in a social network. Unlike our setting, they allow each agent that receives the query and forwards it to offer its own reward for a successful answer. The final rewards are only allocated along the path to the agent that gave the answer as each agent along the way receives the reward for passing the answer back along the path. A similar reward mechanism (that was more structured) was used by the team from MIT that won the DARPA network challenge [1].

Finally, our work also deals with the issue of Sybil attacks that have appeared in many other contexts such as reputation mechanisms [6], combinatorial auctions [18], and social choice [19].

2 Preliminaries

Unless stated otherwise, all trees addressed in this paper are assumed to be finite and directed from the (unique) root towards the leaves. A typical tree will be denoted by T ; its root is denoted by r . We use the standard (directed) tree notions of *parent*, *child*, *leaf*, *descendant*, and *ancestor* in their

natural sense; the parent of a (non-root) node u in T is denoted $p_T(u)$. The *degree* of node $u \in T$, denoted $\deg_T(u)$, is just the number of children u has in T . Given two nodes $u, v \in T$ such that u is an ancestor of v , we define the *distance* from u to v , denoted $\delta_T(u, v)$, as the number of hops (i.e., edges) along the unique path in T leading from u to v ; the distance from u to itself is defined to be $\delta_T(u, u) = 0$. If $\delta_T(u, v) = k > 0$, then we refer to u as the k^{th} *ancestor* of v . We denote the subtree of T rooted at u by T_u . The *height* of T , denoted $h(T)$, is defined to be the maximum distance from r to any leaf in T ; the *height* of u in T , denoted $h_T(u)$, is simply $h(T_u)$. The *depth* of u in T , denoted $\text{dep}_T(u)$, is the distance $\delta_T(r, u)$ from the root of T to u . When the tree T is clear from the context, we may omit the subscripts and simply write $p(u)$, $\deg(u)$, $\delta(u, v)$, $h(u)$, and $\text{dep}(u)$.

A reward mechanism is a function that maps a non-negative real reward $R(T)$ to every finite rooted tree T . We think of T as a (subtree of a) referrals tree and of $R(T)$ as the reward of the root r of T . (Recall that defining the reward in that manner is made possible due to the subtree constraint requiring that the reward of a node in the referrals forest depends only on its subtree.) The *profit* of r is actually $R(T) - \pi$ as r paid $\pi \geq 1$ for purchasing the product when she joined the referrals forest. By the budget constraint, it is assumed that $\sum_{u \in \mathcal{T}} R(T_u) \leq |\mathcal{T}|$. The unbounded reward constraint guarantees the existence of some positive integer d such that for every real R , there exists some tree T of maximum degree d such that $R(T) \geq R$. The notation $R_{\mathcal{M}}(\cdot)$ is used when it is important to emphasize that the function $R(\cdot)$ is associated with the reward mechanism \mathcal{M} .

3 The Geometric Mechanism

In this section we focus on the following family of reward mechanisms, referred to as *geometric mechanisms*. Given two constants $0 < a < 1$ and $b > 0$ such that $b + 1 \leq 1/a$, the reward from a referral tree T under the (a, b) -*geometric mechanism* is defined to be

$$R(T) = \sum_{u \in T} a^{\text{dep}(u)} \cdot b .$$

The constraints on a and b ensure that the amount contributed by each node to the reward of its ancestors will not exceed 1. This simple mechanism is very popular with pyramid schemes; as we will show soon, this is no coincidence. Let us begin by defining and discussing three basic properties of reward mechanisms:

Additivity (ADD): We define the operation \cup on trees such that if T_1, T_2 are trees, then $T_1 \cup T_2$ is the tree formed by contracting (or merging) the roots of T_1 and T_2 . ADD is then stated as follows: $R(T) + R(T') = R(T \cup T')$.

This property suggests that if two disjoint trees are merged at the root, then the reward of the root is exactly the sum of the rewards of the two original trees. Generally speaking this property

implies that the reward to each node can be independently attributed to the subtrees rooted at its children.

Child Dependence (CD): The reward of the root is uniquely determined by the rewards of its children. This property ensures that the actual computation of the rewards can be performed locally. In fact, we shall consider a weaker condition for this property: If the root of T has a single child u , then $R(T)$ is uniquely determined by $R(T_u)$. This is captured by a function $\chi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ (a property of the mechanism) so that $R(T) = \chi(R(T_u))$.

Depth Level Dependence (DLD): $R(T)$ is uniquely determined by the number of nodes on each depth level in T . We denote by d_k the number of nodes of T at depth level $k > 0$, and the infinite vector containing these numbers for all depth levels by $d = (d_1, \dots, d_h, 0, 0, \dots)$, where h is the height of the tree. Let \mathcal{D} be the set of all such vectors, i.e., the set of all infinite vectors over $\mathbb{Z}_{\geq 0}$ with a strictly positive prefix followed by a countably infinite suffix of zeros. Then DLD implies that there exists some function $f : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ (a property of the mechanism) such that $R(T) = f(d)$.

This property essentially means that the credit for a referral depends solely on how direct (or better said, indirect) this referral is.

Theorem 3.1. *A reward mechanism satisfies DLD, ADD, and CD if and only if it is a geometric mechanism.*

To prove the theorem, we will need the definition of the following additional property:

Summing Contributions (SC): There exists a sequence $\{c_k\}_{k \geq 1}$ of non-negative reals such that

$$R(T) = \sum_{u \in T} c_{\text{dep}(u)} = \sum_{k=1}^{\infty} \#\text{nodes at depth level } k \cdot c_k .$$

That is, SC implies that each node in the tree T contributes some independent amount to the root, and that amount depends only on its depth. The following lemma reveals the connection between this property and the ones we have already defined.

Lemma 3.2. *A reward mechanism satisfies SC if and only if it satisfies DLD and ADD.*

Proof. We start with some useful notation. Let $d^{>k}$ denotes the infinite vector d after the first k elements have been removed, i.e., $d^{>k} = (d_{k+1}, d_{k+2}, \dots)$. We denote by $m \circ d$ the infinite vector that starts with the element m and continues with the elements of the infinite vector d , i.e., $m \circ d = (m, d_1, d_2, \dots)$. Finally, we denote the all-zeros infinite vector by $\vec{0} = (0, 0, \dots)$.

The direction in which SC implies DLD and ADD is trivial. We shall establish the converse direction using an inductive argument. Specifically, we prove by induction on k that for every $k \geq 0$, there

exist non-negative reals c_1, \dots, c_k and an additive function⁴ $g_k : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ such that

$$R(T) = \sum_{i=1}^k c_i \cdot d_i + g_k(d^{>k}) .$$

The base of the induction, for $k = 0$, is satisfied by setting $g_0(d) = f(d)$, where f is the function promised by DLD. The additivity of g_0 is then guaranteed by ADD.

For the inductive step, we assume that there exist some non-negative reals c_1, \dots, c_{k-1} and an additive function $g_{k-1} : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ such that

$$R(T) = \sum_{i=1}^{k-1} c_i \cdot d_i + g_{k-1}(d^{>k-1}) .$$

The assertion for k is established by setting

$$\begin{aligned} c_k &= g_{k-1}(1 \circ \vec{0}) ; \text{ and} \\ g_k(d) &= g_{k-1}(1 \circ d) - c_k . \end{aligned}$$

To verify that the function g_k is indeed additive, we employ the additivity of g_{k-1} , observing that

$$\begin{aligned} g_k(d + e) &= g_{k-1}(1 \circ (d + e)) - c_k \\ &= g_{k-1}(2 \circ (d + e)) - g_{k-1}(1 \circ \vec{0}) - c_k \\ &= g_{k-1}(1 \circ d) + g_{k-1}(1 \circ e) - 2 \cdot c_k \\ &= g_k(d) + g_k(e) . \end{aligned}$$

The proof of the inductive step can now be completed due to the inductive hypothesis since

$$\begin{aligned} &f(d) \\ &= \sum_{i=1}^{k-1} c_i \cdot d_i + g_{k-1}(d^{>k-1}) \\ &= \sum_{i=1}^{k-1} c_i \cdot d_i + g_{k-1}(d_k \circ d^{>k}) \\ &= \sum_{i=1}^{k-1} c_i \cdot d_i + g_{k-1}(1 \circ d^{>k}) + g_{k-1}((d_k - 1) \circ \vec{0}) \\ &= \sum_{i=1}^{k-1} c_i \cdot d_i + g_k(d^{>k}) + c_k + (d_k - 1) \cdot c_k \\ &= \sum_{i=1}^k c_i \cdot d_i + g_k(d^{>k}) . \end{aligned}$$

⁴ In this context a function $g : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ is said to be additive if $g(d + d') = g(d) + g(d')$, where the summation in $d + d'$ is coordinate-wise.

The lemma follows by taking $k = h(T)$ as every additive function g must satisfy $g(\vec{0}) = 0$ (by definition, $g(\vec{0}) = g(\vec{0} + \vec{0}) = 2 \cdot g(\vec{0})$). \square

Theorem 3.1 is established by showing that the contribution values c_k form a geometric progression.

Lemma 3.3. *A reward mechanism satisfies SC and CD if and only if it is a geometric mechanism.*

Proof. It is trivial to show that a geometric mechanism satisfies both properties, so we focus on the converse direction. Let us restrict our attention to a specific class of trees: For $n > 1$ and $m > 0$, we denote by $T(n, m)$ the tree consisting of $n + m$ nodes organized as a path of length $n - 1$ emerging from the root with the last node in this path having m children, all of which are leaves. Refer to Figure 1 for illustration.

Recall that SC implies the existence of constants c_1, c_2, \dots that determine the contribution of nodes at each depth level to the reward of the root. We first argue that c_k must be strictly positive for every $k \geq 1$. To that end, suppose that $c_{k^*} = 0$ for some $k^* \geq 1$ and consider the trees $T(k^*, m)$ and $T(k^*, m')$ for some $m, m' > 0$, $m \neq m'$. SC implies that $R(T(k^*, m)) = R(T(k^*, m'))$ since $c_{k^*} = 0$. By CD, we conclude that the same holds for $T(k^* + 1, m)$ and $T(k^* + 1, m')$, namely, $R(T(k^* + 1, m)) = R(T(k^* + 1, m'))$, because both the root of $T(k^* + 1, m)$ and that of $T(k^* + 1, m')$ have a single child whose reward is $R(T(k^*, m)) = R(T(k^*, m'))$. This implies that c_{k^*+1} must also be 0 and by induction, that $c_k = 0$ for every $k \geq k^*$. But this contradicts the unbounded reward constraint: if $c_k = 0$ for every $k \geq k^*$, then no tree T of maximum degree d can provide a reward greater than $2 \cdot d^{k^*}$.

So, assume hereafter that $c_k > 0$ for every $k \geq 1$. In attempt to simplify the analysis, we shall impose another assumption on the contribution values c_k . Specifically, we assume that each c_k is rational, so that $c_k = x_k/y_k$ for some positive integers x_k and y_k . We later on outline how this assumption can be lifted.

Let us compare the reward that is given to the root node in two specific $T(n, m)$ trees (recall that a $T(n, m)$ tree has one node at each level 0 to $n - 1$, and m nodes at level n):

$$\begin{aligned}
R(T(k-1, x_k \cdot y_{k-1} + 1)) &= \sum_{i=1}^{k-2} c_i + (x_k \cdot y_{k-1} + 1) \cdot c_{k-1} \\
&= \sum_{i=1}^{k-1} c_i + x_{k-1} \cdot x_k \\
&= \sum_{i=1}^{k-1} c_i + x_{k-1} \cdot y_k \cdot c_k \\
&= R(T(k, x_{k-1} \cdot y_k)) .
\end{aligned} \tag{1}$$

Now, observe that CD implies that if $R(T(n, m)) = R(T(n', m'))$, then $R(T(n+1, m)) = R(T(n'+1, m'))$.

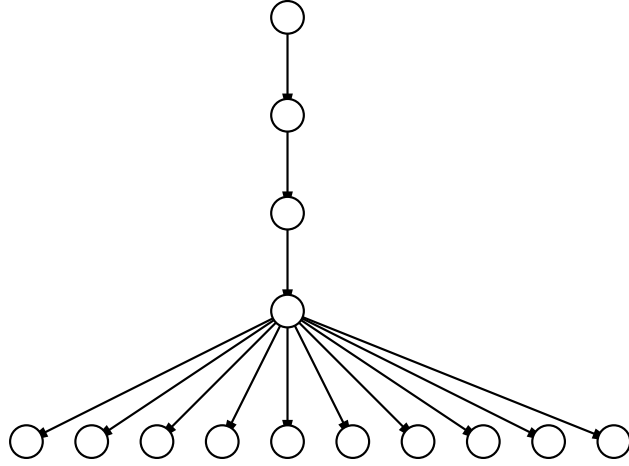


Figure 1: $T(n, m)$ for $n = 4$ and $m = 10$.

$1, m')$). By applying this observation to Equation 1, we conclude that

$$R(T(k, x_k \cdot y_{k-1} + 1)) = R(T(k + 1, x_{k-1} \cdot y_k)) .$$

SC then implies that

$$\sum_{i=1}^{k-1} c_i + (x_k \cdot y_{k-1} + 1) \cdot c_k = \sum_{i=1}^k c_i + (x_{k-1} \cdot y_k) \cdot c_{k+1} ,$$

hence $x_k \cdot y_{k-1} \cdot c_k = (x_{k-1} \cdot y_k) \cdot c_{k+1}$. It follows that

$$(c_k)^2 = c_{k-1} \cdot c_{k+1} , \tag{2}$$

which implies a geometric progression (c_k is the geometric mean of c_{k-1} and c_{k+1}).

Recall that our proof thus far only works if all c_k 's are rational numbers. We can extend the proof to irrational numbers if we add a requirement on the continuity of the function that determines the rewards of a parent from the reward of its children. If the c_k 's are not rational, it is possible to approximate them as closely as one wishes with rational numbers x_k/y_k and with the extra assumption, the derivation above results in an equation similar to Equation 2 which is modified with terms that represent the error in the approximation of c_k . As this error can be made arbitrarily small, Equation 2 holds even for irrational values. \square

Another appealing property of geometric mechanisms is that the contribution of descendants to their ancestor decreases with distance. This reflects the fact that the ancestor gets less credit for more distant indirect referrals.

3.1 Property Independence

We now show that each of the three properties we used to characterize the family of geometric mechanisms is required, i.e., if we remove one of the three properties then there exists another mechanism (outside the geometric family) for which the remaining three hold. We briefly describe such a mechanism in each case.

A mechanism without CD: We define the mechanism by determining the contribution of a child at depth k to the root. The reward of the root will then be the sum of contributions of all children. We set the contribution to be $c_k = 2^{-\lfloor k/2 \rfloor} \cdot 3^{-\lceil k/2 \rceil}$. The reward is then $R(T) = \sum_{u \in T} c_{\text{dep}(u)}$.

The mechanism described is very similar to the geometric mechanism, except that the contribution decays at a factor of $1/3$ for odd levels, and $1/2$ for even levels and so the reward of the parent cannot be computed directly from that of its direct children (the parity of the level matters).

It is easy to verify that this mechanism adheres to the three constraints listed in Section 1: It does not exceed the budget, as it always pays less than the geometric mechanism with factor $1/2$; the reward of a node depends only on the subtree below it by definition; and a tree of degree ≥ 3 can obtain infinite reward. Properties **ADD** and **DLD** are clearly satisfied since the mechanism is defined by determining the contribution values of **SC**.

A mechanism without DLD: We define a mechanism that is similar to the geometric mechanism, but distinguishes between the contributions of a leaf and an internal node (nodes that have children) in the tree. Each leaf will contribute 2^{-k} to its k^{th} ancestor, while each internal node will contribute 2^{-k-1} to its k^{th} ancestor.

This mechanism does not satisfy **DLD**, as it is not enough to know the number of nodes at each level, since their contribution depends on whether they are internal or leaf nodes (this is certain only for nodes in the last level or for levels that contain a single node).

The mechanism satisfies the three constraints of Section 1: The rewards depend only on the rooted subtrees; the budget is not exceeded as it always pays less than the $(1/2,1)$ -geometric mechanism; and it offers infinite reward to trees of degree 2 or above as it pays more than the $(1/2,1/2)$ -geometric mechanism.

Furthermore, the mechanism trivially satisfies **ADD** since merging two trees at the root maintains the leaves and internal nodes of the original trees and hence their respective contributions. Finally, the mechanism also satisfies **CD**, as the reward of each node can be computed from the rewards of its children: If the child has no reward it must be a leaf and its contribution to the parent is $1/2$. Otherwise, it is an internal node and its contribution to the parent is $1/4$ plus $1/2$ of the child's own reward.

A mechanism without ADD: Let \mathcal{M}_g denote the $(1/2,1/2)$ -geometric mechanism. Given a tree T of height h , we denote by T^+ one of the trees that is obtained from T by adding a leaf to it at level

$h + 1$. We can define a non-additive mechanism \mathcal{M} in the following manner: $R_{\mathcal{M}}(T) = R_{\mathcal{M}_g}(T^+)$. This mechanism simply credits the tree T according to the geometric mechanism and adds a small bonus as if the tree had one extra leaf. \mathcal{M} is not additive since merging two trees only gives the merged tree a bonus for a single leaf rather than two.

\mathcal{M} satisfies the three constraints of Section 1: The reward is clearly based only on the subtree rooted at the receiving node; the unbounded reward constraint is satisfied as the mechanism pays slightly more than a geometric mechanism that certainly has this property; and the budget is not exceeded, since the total payments of a $(1/2, 1/2)$ -geometric mechanism are bounded by $1/2 \cdot |T|$, therefore the bonus for an additional leaf at level $h + 1$ does not push the total payments over the limit.

It is also simple to verify that the mechanism satisfies DLD. To see that it also satisfies CD notice that we can tell the height of the tree T from the reward of its root (and so the reward of the parent can be computed from that of the child by removing the bonus of the extra leaf and using the geometric mechanism's CD property). Indeed, to determine the height of the tree, examine the representation of $R(T)/b$ in base $1/a$ (where a and b are the parameters of the geometric-mechanism used to define \mathcal{M}). The least significant digit that is non-zero will be $h+1$ places after the separator, because of the bonus of the leaf at level $h + 1$.

4 Sybil Attacks

Our goal in this section is to develop reward mechanisms which are not vulnerable to forging identities on behalf of the users. Let us begin by introducing the notions of a *split* and a *local split*. Consider some tree T and some node $v \in T$ and let u_1, \dots, u_k be the children of v in T . Intuitively speaking, a split of v refers to a scenario in which v presents itself as *several* nodes — a.k.a. *replicas* — thus modifying the (sub)tree T_v that determines its reward (possibly turning it into several trees), while keeping T_{u_1}, \dots, T_{u_k} intact. A local split refers to the special case of a split in which u_1, \dots, u_k are forced to share the same parent in the resulting tree.

Formally, we say that the tree collection $\{\tilde{T}^1, \dots, \tilde{T}^m\}$ can be obtained from T_v by a *split* of v if

- (1) for every $1 \leq i \leq k$, there exists a single $1 \leq j(i) \leq m$ such that $u_i \in \tilde{T}^{j(i)}$; and
- (2) $\tilde{T}_{u_i}^{j(i)} = T_{u_i}$ for every $1 \leq i \leq k$.

The nodes in

$$\left(\tilde{T}^1 \cup \dots \cup \tilde{T}^m\right) - \left(\tilde{T}_{u_1}^{j(1)} \cup \dots \cup \tilde{T}_{u_k}^{j(k)}\right)$$

are referred to as the *replicas* of v under that split. By definition, u_i must be a (direct) child of some replica of v for every $1 \leq i \leq k$ as otherwise, at least one of the subtrees rooted at u_1, \dots, u_k must have been changed, thus violating condition (2). Refer to Figure 2 for an illustration of a split. The split is called *local* if u_1, \dots, u_k are all children of the same replica of v .

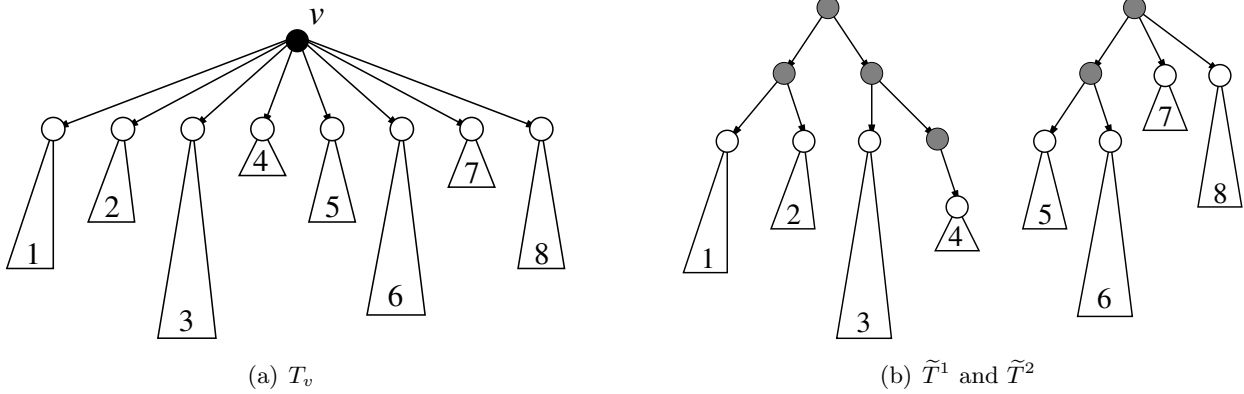


Figure 2: The tree collection $\{\tilde{T}^1, \tilde{T}^2\}$ can be obtained from T_v by a split of v . The white circles depict the children of v in T (2(a)) and their positions in \tilde{T}^1 and \tilde{T}^2 after the split (2(b)). The gray circles in 2(b) depict the replicas of v under that split.

The semantic of the aforementioned split is as follows. The user corresponding to v forges some new identities that correspond to its replicas in $\tilde{T}^1, \dots, \tilde{T}^m$. Some of these replicas purchase the product via referrals from $p_T(v)$ — they form the roots of the trees $\tilde{T}^1, \dots, \tilde{T}^m$; the rest of the replicas purchase the product via the referrals of other replicas that already purchased it. The referrals to u_1, \dots, u_k are then made from the appropriate replicas, so that eventually T_v is replaced by the trees $\tilde{T}^1, \dots, \tilde{T}^m$, the roots of which are all children of $p_T(v)$.

It is assumed that u_i does not distinguish between the original v and its new replicas. Therefore the new referral (made from a replica of v rather than from v itself) looks to u_i like a referral from v and she relies on it to purchase the product just like she did with the referral from v in the original scenario. Under local splits, this assumption is lifted as all children of v purchase the product from the same replica that may have the identity of v herself.

What's in it for v ? Clearly, v has to invest $\pi \times \#\text{replicas}$ in introducing the new replicas (purchasing new copies of the product). However, she now collects the rewards from all her replicas, which sums up to

$$\sum_{i=1}^m \sum_{\text{replica } u \text{ of } v \text{ in } \tilde{T}^i} R(\tilde{T}_u^i).$$

Thus, the profit⁵ of v changes from $R(T_v) - \pi$ to $\sum_{i=1}^m \sum_{\text{replica } u \text{ of } v \text{ in } \tilde{T}^i} R(\tilde{T}_u^i) - \pi \times \#\text{replicas}$; the split is called *profitable* for v if this change is positive. This leads to the definition of the following two properties of reward mechanisms.

Split Proof (SP): A reward mechanism satisfies **SP** if it does not admit a profitable split for any node v in any tree T .

⁵ Here, we assume that a user has no usage in more than one copy of the product. This is a valid assumption, e.g., in the context of information goods.

Local Split Proof (LSP): A reward mechanism satisfies LSP if it does not admit a profitable local split for any node v in any tree T .

The geometric mechanisms presented in Section 3 do not satisfy SP. In fact, they do not even satisfy LSP (see Section 4.1). A simple mechanism that do satisfy SP is the *single level* mechanism defined by fixing $R_{\mathcal{M}_{sl}}(T) = \alpha \cdot \deg(r)$ for some constant $\alpha \leq 1$, however, this mechanism does not adhere to the unbounded reward constraint.

In Section 4.1 we establish two negative results regarding the design of split-proof mechanisms. On the positive side, we devise two mechanisms which are resilient to false-name manipulations. The first one, presented in Section 4, satisfies the stronger property SP, however, it is complicated to implement and not very appealing. The second one, presented in Section 4.3, satisfies only LSP, but it is more natural and much easier to implement.

4.1 Negative Results

Let us now exhibit two negative results regarding the design of split-proof mechanisms. The first result shows that the reward guaranteed to a node in a split-proof mechanism cannot be a constant fraction of even its least influential child.

Lemma 4.1. *A reward mechanism that satisfies LSP cannot guarantee a node some fraction $0 < \alpha \leq 1$ of the reward of its least rewarded child.*

Proof. Assume by way of contradiction that such a mechanism exists and let T be a tree such that $R(T) > \pi/\alpha$ (such a tree must exist due to the unbounded reward constraint). Let T' be the tree in which the root u has a single child v such that $T_v = T$. Notice that T' can be obtained from T by a local split of v . The benefit to v from this split is

$$R(T') - \pi \geq \alpha \cdot R(T) - \pi > 0$$

Therefore, v gains from performing this split and the mechanism is not LSP. □

Next, we show that even a family of reward mechanisms much wider than geometric mechanisms is still not split-proof. This requires the introduction of another property which is clearly satisfied by every geometric mechanism, yet, cannot replace any of the characterizing properties listed in Theorem 3.1.

Monotonicity (MONO): If the tree T can be obtained from the tree T' by removing some leaf, then $R(T) < R(T')$.

Lemma 4.2. *A reward mechanism that satisfies MONO and ADD cannot satisfy SP.*

Proof. Let $T^p(h, d)$ denote a perfect tree of height h and degree d . Notice that under any mechanism with MONO, we have $R(T^p(h, d)) < R(T^p(h + 1, d))$.

Now consider the tree $T' = \bigcup_{i=1}^d T^p(h, d)$ (that is, d copies of $T^p(h, d)$ sharing the same root). ADD implies that $R(T') = d \cdot R(T^p(h, d))$. The root of T' can perform a strong split and simulate the tree $T^p(h+1, d)$, where it takes the role of the new root and its d children. Its profit is then increased by $R(T_p(h+1, d)) - d \cdot \pi > R(T_p(h, d)) - d \cdot \pi$. With the right choice of degree d , this can be made positive by sufficiently increasing the height h (due to the unbounded reward constraint and MONO). \square

4.2 A Split-Proof Mechanism

In this section we present a split-proof reward mechanism, denoted $\mathcal{M}_{\text{split}}$. Informally, the mechanism $\mathcal{M}_{\text{split}}$ is defined in two stages: in the first stage, we define a simple *base* mechanism, denoted $\mathcal{M}_{\text{base}}$; $\mathcal{M}_{\text{split}}$ is then defined with respect to the maximum profit a node can make under $\mathcal{M}_{\text{base}}$ from splits.

Mechanism $\mathcal{M}_{\text{base}}$. The base mechanism $\mathcal{M}_{\text{base}}$ is defined by setting $R_{\mathcal{M}_{\text{base}}}(T)$ to be the maximum $h \in \mathbb{Z}_{\geq 0}$ such that T exhibits as a subtree, a perfect binary tree⁶ B rooted at r whose height is h . In that case we say that B *realizes* $R_{\mathcal{M}_{\text{base}}}(T)$ (see Figure 3). If there are several perfect binary trees that can realize $R_{\mathcal{M}_{\text{base}}}(T)$, then it will be convenient to take the first one in a lexicographic order based on a breadth-first-search traversal and consider it as *the* perfect binary tree that realizes $R_{\mathcal{M}_{\text{base}}}(T)$.

A node $u \in T$ is said to be *visible* in T if it belongs to the perfect binary tree that realizes $R_{\mathcal{M}_{\text{base}}}(T)$; otherwise, u is said to be *invisible* in T . This definition is extended as follows: given some ancestor v of u in T , u is said to be *visible* (respectively, *invisible*) to v if u is visible (resp., invisible) in T_v . Note that if u is invisible to v , then it is also invisible to $p_T(v)$ (assuming of course that $v \neq r$). The contrary is not necessarily true: u may be visible to v but invisible to $p_T(v)$. By definition, for every node $u \in T$ and for every $j \in \mathbb{Z}_{\geq 1}$, it holds that u admits either 2^j or 0 visible depth- j descendants.

The mechanism $\mathcal{M}_{\text{base}}$ can be redefined by setting

$$R_{\mathcal{M}_{\text{base}}}(T) = \sum_{\text{visible } u \in T} 2^{-\delta_T(u,r)} .$$

This alternative view of mechanism $\mathcal{M}_{\text{base}}$ calls for the definition of *contributions*: a node $u \in T$ *contributes* 2^{-k} to the reward of its k^{th} ancestor v , $k \geq 1$, if u is visible to v ; otherwise, u does not contribute anything to the reward of v . Let $C_{\mathcal{M}_{\text{base}}}(u, v)$ denote the contribution of u to the reward of v under $\mathcal{M}_{\text{base}}$. The reward of a node can now be calculated by summing the contributions that its descendants make to it. This implies that $\mathcal{M}_{\text{base}}$ satisfies the budget constraint: the total

⁶ A perfect binary tree is a rooted tree in which all leaves are at the same distance from the root and all non-leaves have exactly two children.

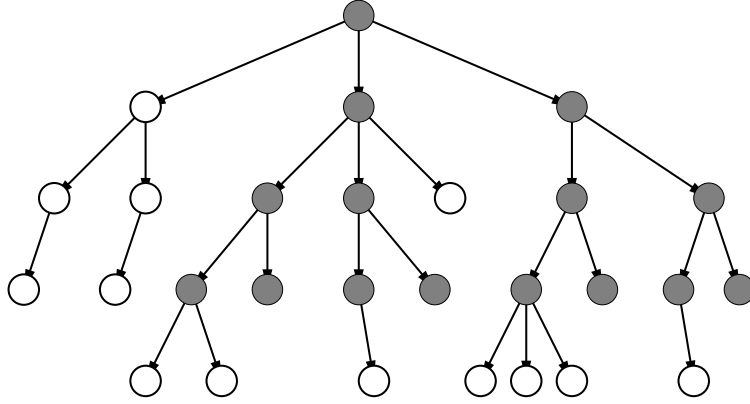


Figure 3: The tree T and the perfect binary tree that realizes $R_{\mathcal{M}_{\text{base}}}(T)$ (nodes depicted by gray circles).

contribution made by a node $u \in T$ to all its ancestors is bounded from above by the geometric sum $\sum_{j=1}^{\delta(u,r)} 2^{-j} < 1$, hence, by changing the summation, we conclude that $\sum_{v \in T} R_{\mathcal{M}_{\text{base}}}(T_v) < |T|$.

Mechanism $\mathcal{M}_{\text{split}}$. The mechanism $\mathcal{M}_{\text{split}}$ is defined by setting $R_{\mathcal{M}_{\text{split}}}(T)$ so that it reflects the maximum profit that r can get under $\mathcal{M}_{\text{base}}$ from splits. More formally, let \mathcal{S} be the collection of all tree collections that can be obtained from T by a split of r . Then $\mathcal{M}_{\text{split}}$ is defined by setting

$$R_{\mathcal{M}_{\text{split}}}(T) = \sup_{\tilde{\mathcal{T}} = \{\tilde{T}^1, \dots, \tilde{T}^m\} \in \mathcal{S}} \left\{ \left(\sum_{i=1}^m \sum_{\text{replica } v \text{ of } r \text{ in } \tilde{T}^i} R_{\mathcal{M}_{\text{base}}}(\tilde{T}_v^i) \right) - \pi \cdot (|\tilde{T}^1 \cup \dots \cup \tilde{T}^m| - |T|) \right\}.$$

To avoid cumbersome notation, we shall denote $\rho(\tilde{\mathcal{T}}) \equiv \sum_{i=1}^m \sum_{\text{replica } v \text{ of } r \text{ in } \tilde{T}^i} R_{\mathcal{M}_{\text{base}}}(\tilde{T}_v^i)$ and $\bigcup \tilde{\mathcal{T}} \equiv \tilde{T}^1 \cup \dots \cup \tilde{T}^m$ so that $R_{\mathcal{M}_{\text{split}}}(T) = \sup_{\tilde{\mathcal{T}} \in \mathcal{S}} \{\rho(\tilde{\mathcal{T}}) - \pi \cdot (|\bigcup \tilde{\mathcal{T}}| - |T|)\}$. Refer to Figure 4 for illustration.

We first observe that $R_{\mathcal{M}_{\text{split}}}(\cdot)$ is well defined by showing that we may assume without loss of generality that \mathcal{S} is finite. To that end, we argue that it is sufficient to consider $\tilde{\mathcal{T}} \in \mathcal{S}$ such that every replica of r in $\tilde{\mathcal{T}}$ is of degree at least 2, which means that there are less than $\deg_T(r)$ replicas in total, and hence there are finitely many different options for $\tilde{\mathcal{T}}$. Indeed, assume that v is a replica of r in some tree $\tilde{T} \in \tilde{\mathcal{T}}$ so that $\deg_{\tilde{\mathcal{T}}}(v) < 2$ and let \tilde{T}' be the tree obtained from \tilde{T} by removing v and turning its sole child (in case that $\deg_{\tilde{\mathcal{T}}}(v) = 1$) into a child of $p_{\tilde{T}}(v)$ or into the new root if v is the root of \tilde{T} . Note that by the definition of splits, $\tilde{\mathcal{T}} \cup \{\tilde{T}'\} - \{\tilde{T}\}$ can also be obtained from T by a split of r . Since $\deg_{\tilde{\mathcal{T}}}(v) < 2$, we know that $R_{\mathcal{M}_{\text{base}}}(\tilde{T}_v) = 0$. Moreover, every descendant u of v in \tilde{T} must be invisible to the ancestors of v in \tilde{T} (which are also replicas of r), hence u does not contribute anything to the rewards of these ancestors under $\mathcal{M}_{\text{base}}$. The argument is established by recalling that the total contribution of v to the rewards of its ancestors in \tilde{T} is smaller than $1 \leq \pi$, which means that it was not worthwhile for r to generate the replica v .

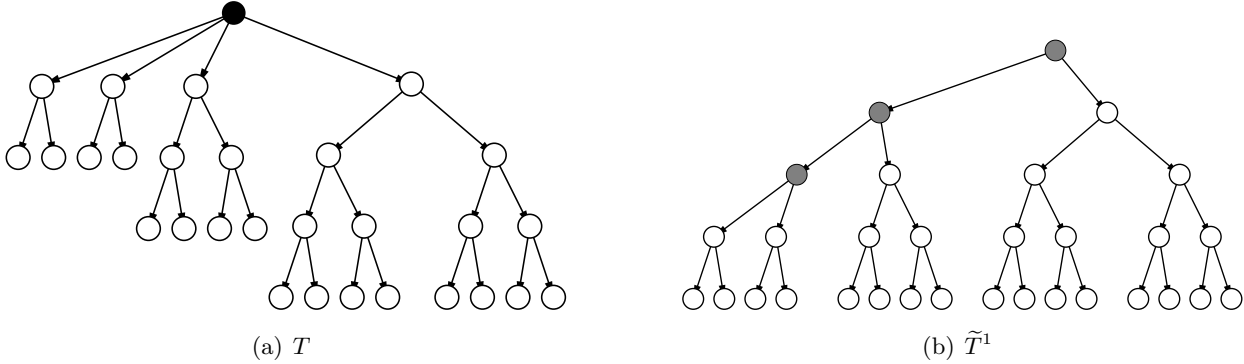


Figure 4: The tree T and a possible split of its root r (into a single tree \tilde{T}^1). The gray circles in 4(b) depict the replicas of r under that split. If $\pi = 1$, then this split realizes $R_{\mathcal{M}_{\text{split}}}(T) = 2 + 3 + 4 - 2 \cdot \pi = 7$.

We say that the tree collection $\tilde{\mathcal{T}} \in \mathcal{S}$ realizes $R_{\mathcal{M}_{\text{split}}}(T)$ if $R_{\mathcal{M}_{\text{split}}}(T) = \rho(\tilde{\mathcal{T}}) - \pi \cdot (|\bigcup \tilde{\mathcal{T}}| - |T|)$. If there are several tree collections that can realize $R_{\mathcal{M}_{\text{split}}}(T)$, then it will be convenient to choose one (arbitrarily) and consider it as *the* tree collection that realizes $R_{\mathcal{M}_{\text{split}}}(T)$.

Excessive contributions. It is easy to see that $\mathcal{M}_{\text{split}}$ satisfies the subtree constraint and the unbounded reward constraint ($\mathcal{M}_{\text{base}}$ already satisfies the unbounded reward constraint and the rewards under $\mathcal{M}_{\text{split}}$ dominates those of $\mathcal{M}_{\text{base}}$). Moreover, by definition, $\mathcal{M}_{\text{split}}$ satisfies **SP**, i.e., a node cannot increase its profit by splitting (recall that the mechanism takes every possible split into account). The difficult part is to show that $\mathcal{M}_{\text{split}}$ satisfies the budget constraint, that is, $\sum_{v \in T} R_{\mathcal{M}_{\text{split}}}(T_v) \leq |T|$.

The first step towards fulfilling this task is to extend the notion of contribution to the reward mechanism $\mathcal{M}_{\text{split}}$. Consider some node $u \in T$ and some ancestor v of u . Let $\tilde{\mathcal{T}}$ be the tree collection obtained from T_v by a split of v that realizes $R_{\mathcal{M}_{\text{split}}}(T_v)$ and let \tilde{T} be the tree in $\tilde{\mathcal{T}}$ that contains u . By definition, several replicas of v in \tilde{T} may be ancestors of u (in \tilde{T}); denote these replicas by v_1, \dots, v_ℓ . The *contribution* of u to v under $\mathcal{M}_{\text{split}}$, denoted $C_{\mathcal{M}_{\text{split}}}(u, v)$, is then defined to be

$$C_{\mathcal{M}_{\text{split}}}(u, v) = \sum_{j=1}^{\ell} C_{\mathcal{M}_{\text{base}}}(u, v_j). \quad (3)$$

We argue that

$$\sum_{v \in T} R_{\mathcal{M}_{\text{split}}}(T_v) \leq \sum_{u \in T} \sum_{\text{ancestors } v \text{ of } u} C_{\mathcal{M}_{\text{split}}}(u, v). \quad (4)$$

At first glance, inequality (4) may be surprising as its right hand side does not take into account contributions made by replicas that do not exist in T . However, $\mathcal{M}_{\text{split}}$ is defined with respect to the profit (rather than reward) that a node can get under $\mathcal{M}_{\text{base}}$ from splits. The total contribution of a node to all its ancestors under $\mathcal{M}_{\text{base}}$ is smaller than 1, hence each replica of node u contributes

less than 1 to the rewards of the other replicas of u (it does not contribute anything to any other node in T). Inequality (4) follows since π is assumed to be larger than 1. Our goal in what follows is to show that

$$\sum_{u \in T} \sum_{\text{ancestors } v \text{ of } u} C_{\mathcal{M}_{\text{split}}}(u, v) \leq |T|. \quad (5)$$

In attempt to establish inequality (5), it may seem natural to bound the total contribution that each node makes to its ancestors, showing that this is at most 1, as we did with $\mathcal{M}_{\text{base}}$. Unfortunately, this does not work: If the contribution $C_{\mathcal{M}_{\text{split}}}(u, v)$ of node u to its ancestor v under $\mathcal{M}_{\text{split}}$, as defined in equation (3), is composed from $\mathcal{M}_{\text{base}}$ -contributions to several (more than one) replicas of v , then $C_{\mathcal{M}_{\text{split}}}(u, v) > C_{\mathcal{M}_{\text{base}}}(u, v)$. Consequently, the total contribution that u makes to the rewards of all its ancestors under $\mathcal{M}_{\text{split}}$ may exceed 1. This obstacle requires a careful accounting argument that we now turn to describe.

Consider some node $u \in T$. Let $\Gamma(u)$ denote the set of all ancestors v of u in T such that $C_{\mathcal{M}_{\text{split}}}(u, v) > 0$. Assuming that $\Gamma(u) \neq \emptyset$, let $\varphi(u)$ be the node $v \in \Gamma(u)$ that maximizes $\delta_T(v, u)$. Now, consider some node $v \in \Gamma(u)$. Let $\tilde{\mathcal{T}}$ be the tree collection obtained from T_v by a split of v that realizes $R_{\mathcal{M}_{\text{split}}}(T_v)$, let \tilde{T} be the tree in $\tilde{\mathcal{T}}$ that contains u , and let \tilde{v} be the highest ancestor of u in \tilde{T} such that u is visible to \tilde{v} . Note that \tilde{v} must be a replica of v in \tilde{T} since $v \in \Gamma(u)$. Moreover, $\delta_{\tilde{T}}(\tilde{v}, u) \geq \delta_T(v, u)$ and the contribution of u to v 's reward under $\mathcal{M}_{\text{split}}$ is $C_{\mathcal{M}_{\text{split}}}(u, v) = \sum_{j=\delta_T(v, u)}^{\delta_{\tilde{T}}(\tilde{v}, u)} 2^{-j}$. Thus, the parameter $\delta_{\tilde{T}}(\tilde{v}, u)$ plays an important role in the calculation of $C_{\mathcal{M}_{\text{split}}}(u, v)$ — denote it by $\Delta(u, v)$, so that $C_{\mathcal{M}_{\text{split}}}(u, v) = 2^{1-\delta_T(v, u)} - 2^{-\Delta(u, v)}$.

Deficits and surpluses. Recall that $C_{\mathcal{M}_{\text{base}}}(u, v) = 2^{-\delta_T(v, u)}$ and that such a contribution would have resulted in the desired bound of 1 on the total contribution of node u to all its ancestors. Informally speaking, if $\Delta(u, v) > \delta_T(v, u)$, then u exhibits a *deficit*. On the other hand, if $\varphi(u) = v$, then u does not contribute anything to the rewards of the ancestors of v , which leaves us with a small *surplus*. Our argument will rely on covering the total deficits with the total surpluses.

Formally, given some nodes $u \in T$ and $v \in \Gamma(u)$, define

$$\begin{aligned} \text{deficit}(u, v) &= C_{\mathcal{M}_{\text{split}}}(u, v) - C_{\mathcal{M}_{\text{base}}}(u, v) \\ &= 2^{-\delta_T(v, u)} - 2^{-\Delta(u, v)} \end{aligned}$$

and

$$\text{surplus}(u) = 2^{-\Delta(u, \varphi(u))}.$$

Using these notions, we have

$$\begin{aligned}
& \sum_{v \in \Gamma(u)} C_{\mathcal{M}_{\text{split}}}(u, v) \\
&= \sum_{v \in \Gamma(u)} 2^{1-\delta_T(u,v)} - 2^{-\Delta(u,v)} \\
&\leq \sum_{k=1}^{\delta_T(\varphi(u),u)} 2^{-k} + \sum_{v \in \Gamma(u)} 2^{-\delta_T(v,u)} - 2^{-\Delta(u,v)} \\
&= 1 - 2^{-\delta_T(\varphi(u),u)} + \sum_{v \in \Gamma(u)} \text{deficit}(u, v) \\
&= 1 - \text{deficit}(u, \varphi(u)) - \text{surplus}(u) \\
&\quad + \sum_{v \in \Gamma(u)} \text{deficit}(u, v) \\
&= 1 - \text{surplus}(u) + \sum_{v \in \Gamma(u) - \{\varphi(u)\}} \text{deficit}(u, v) .
\end{aligned}$$

We establish the fact that $\mathcal{M}_{\text{split}}$ satisfies the budget constraint by showing that

$$\sum_{u \in T} \sum_{v \in \Gamma(u) - \{\varphi(u)\}} \text{deficit}(u, v) \leq \sum_{u \in T} \text{surplus}(u) , \tag{6}$$

thus “covering” the deficits with the surpluses. This is done in the following manner.

Consider some node $v \in T$ and let $\tilde{\mathcal{T}}$ be the tree collection obtained from T_v by a split of v that realizes $R_{\mathcal{M}_{\text{split}}}(T_v)$; consider some tree $\tilde{T} \in \tilde{\mathcal{T}}$. Fix some replica \tilde{v} of v in \tilde{T} and let h be the height of the perfect binary tree that realizes $R_{\mathcal{M}_{\text{base}}}(\tilde{T}_{\tilde{v}})$. Given some $1 \leq \ell \leq h$, let $D_{\tilde{v},\ell}$ be the set of descendants u of \tilde{v} in \tilde{T} that satisfy: (1) \tilde{v} is the highest ancestor of u in \tilde{T} to which u is visible; and (2) $\Delta(u, v) = \delta_{\tilde{\mathcal{T}}}(\tilde{v}, u) = \ell$. It is interesting to point out that different nodes u in $D_{\tilde{v},\ell}$ may have different values of $\delta_T(v, u)$, although they all have the same value of $\delta_{\tilde{\mathcal{T}}}(\tilde{v}, u) = \ell$.

A crucial observation is that for every two nodes $u \in T$ and $v \in \Gamma(u)$, there exists a unique choice of \tilde{v} and ℓ such that $u \in D_{\tilde{v},\ell}$. Let $D_{\tilde{v},\ell}^+$ be the set of all nodes $u \in D_{\tilde{v},\ell}$ such that $v = \varphi(u)$, i.e., those nodes that do not contribute to the reward of any ancestor of v . Let $D_{\tilde{v},\ell}^- = D_{\tilde{v},\ell} - D_{\tilde{v},\ell}^+$. Inequality (6) is established by proving the following lemma.

Lemma 4.3. *The nodes in $D_{\tilde{v},\ell}^-$ satisfy $\sum_{u \in D_{\tilde{v},\ell}^-} \text{deficit}(u, v) \leq \sum_{u \in D_{\tilde{v},\ell}^+} \text{surplus}(u)$.*

Proof. Denote $D^- = D_{\tilde{v},\ell}^-$ and $D^+ = D_{\tilde{v},\ell}^+$. Recall that v has either 2 or 0 children which are visible to $p_T(v)$; the latter implies that $D^- = \emptyset$, so assume in what follows that the former holds and let v_1, v_2 be the children of v in T which are visible to $p_T(v)$. Let $D_i^- = D^- \cap T_{v_i}$ for $i = 1, 2$. By definition, $D^- = D_1^- \cup D_2^-$ and $D_1^- \cap D_2^- = \emptyset$. Moreover, all nodes u in D_i^- have the same value of $\delta_T(v_i, u)$ — denote it by k_i .

Recall that $\text{deficit}(u_i, v) = 2^{-k_i-1} - 2^{-\ell}$ for every node $u_i \in D_i^-$. Since $|D_i^-| = 2^{k_i}$, we conclude that

$$\begin{aligned} \sum_{u_i \in D_i^-} \text{deficit}(u_i, v) &= 2^{k_i} \left(2^{-k_i-1} - 2^{-\ell} \right) \\ &= \frac{1}{2} - 2^{k_i-\ell} \end{aligned}$$

and

$$\sum_{u \in D^-} \text{deficit}(u, v) = 1 - 2^{k_1-\ell} - 2^{k_2-\ell} .$$

On the other hand, $\text{surplus}(u) = 2^{-\ell}$ for every node $u \in D^+$. Since $|D^+| = 2^\ell - |D_1^-| - |D_2^-| = 2^\ell - 2^{k_1} - 2^{k_2}$, it follows that

$$\begin{aligned} \sum_{u \in D^+} \text{surplus}(u) &= 2^{-\ell} \left(2^\ell - 2^{k_1} - 2^{k_2} \right) \\ &= 1 - 2^{k_1-\ell} - 2^{k_2-\ell} \end{aligned}$$

which establishes the assertion. □

4.3 Resilience to Local Splits

The mechanism presented in Section 4.2 is resilient to splits, but it is far from being intuitive. Moreover, users who recruit a large number of buyers may find that they are not rewarded for many of them (due to limited visibility). We therefore briefly describe another mechanism (in fact, a family of mechanisms) — this time with resilience only to local splits — that is much simpler than the one we have previously shown.

The mechanism $\mathcal{M}_{\text{local}}$ is similar to the geometric mechanism (with some choice of parameters $0 < a < 1$ and $b > 0$ such that $b+1 \leq 1/a$), only that here, we ignore the contributions coming from one of the subtrees rooted at r 's children, say, the largest subtree. More formally, let v_1, \dots, v_k be the children of r in T and assume without loss of generality that $|T_{v_1}| \geq \dots \geq |T_{v_k}|$. Then,

$$\mathbf{R}_{\mathcal{M}_{\text{local}}}(T) = \sum_{u \in T_{v_2} \cup \dots \cup T_{v_k}} a^{\text{dep}_T(u)} \cdot b .$$

Note that a node with one child obtains no reward.

It is easy to see that $\mathcal{M}_{\text{local}}$ satisfies the subtree constraint and the unbounded reward constraint. Moreover, since the rewards under $\mathcal{M}_{\text{local}}$ are dominated by those of a geometric mechanism, it also satisfies the budget constraint. The following lemma establishes the resilience of $\mathcal{M}_{\text{local}}$ to local splits.

Theorem 4.4. *The reward mechanism $\mathcal{M}_{\text{local}}$ satisfies LSP.*

Proof sketch. Let r be the root of T and let v_1, \dots, v_k be its children. By definition, in any local split of r , the nodes v_1, \dots, v_k are children of the same replica of r . A straightforward calculation shows that r does not gain anything from replicas that do not have v_1, \dots, v_k as descendants, hence it suffices to consider local splits that replace T with a tree T' obtained by identifying the root of T with the (sole) leaf of some directed path P . (In that case, the replicas of r are the nodes of P .) The design of $\mathcal{M}_{\text{local}}$ implies that the total reward of r from such a local split remains $R_{\mathcal{M}_{\text{local}}}(T)$ which obviously turns the split into a non-profitable move. \square

5 Conclusions

We have presented a theoretical framework for multi-level marketing mechanisms. Our framework gives host to many possible reward mechanisms, with varying properties. A full characterization of the geometric mechanism family, which is recognized in particular with Pyramid schemes, is established. While we find that geometric mechanisms are not resilient to false-name manipulations, we have instead shown two different mechanisms that can withstand such manipulations.

There are many open avenues for future work. First, many different characterizations can be found for the mechanisms that we presented. It is possible to think of other intuitive properties and attempt to substitute them for the ones we have defined. In addition, the split-proof mechanisms that we have shown may be hard to implement (they are hard to explain to users and deny rewards for some of the referrals that are made). It will be nice to find mechanisms that improve these aspects as well. It will also be interesting to explore other models for marketing in the context of social networks, including models that take the underlying structure of the network into account, and perhaps other details such as the percentage of successful referrals and the timing of messages. Finally, we believe that much can be learned from experimental evaluation, or even deployment of mechanisms that we have suggested. User trials can reveal defects in these reward schemes, and show how readily human users are willing to accept them.

References

- [1] DARPA network challenge. <http://balloon.media.mit.edu/mit/rules/>.
- [2] Thomas Ågotnes, Wiebe van der Hoek, Moshe Tennenholtz, and Michael Wooldridge. Power in normative systems. In *AAMAS*, pages 145–152, 2009.
- [3] David Arthur, Rajeev Motwani, Aneesh Sharma, and Ying Xu. Pricing strategies for viral marketing on social networks. In *Proceedings of the 5th International Workshop on Internet and Network Economics*, WINE '09, pages 101–112, Berlin, Heidelberg, 2009. Springer-Verlag.

- [4] Yoram Bachrach and Jeffrey S. Rosenschein. Computing the banzhaf power index in network flow games. In *AAMAS*, 2007.
- [5] Ning Chen. On the approximability of influence in social networks. *SIAM J. Discret. Math.*, 23:1400–1415, September 2009.
- [6] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, P2PECON '05, pages 128–132, New York, NY, USA, 2005. ACM.
- [7] Jon Kleinberg David Kempe and Eva Tardos. Maximizing the spread of influence in a social network. In *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.
- [8] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. ACM.
- [9] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83:1420–1443, 1978.
- [10] Jason Hartline, Vahab Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 189–198, New York, NY, USA, 2008. ACM.
- [11] Jon Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. In E. Tardos V. Vazirani N. Nisan, T. Roughgarden, editor, *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [12] Jon M. Kleinberg and Prabhakar Raghavan. Query incentive networks. In *FOCS*, pages 132–141, 2005.
- [13] Nimrod Megiddo. Computational complexity of the game theory approach to cost allocation for a tree. *Mathematics of Operations Research*, 3(3):189–196, 1978.
- [14] Stephen Morris. Contagion. *Review of Economic Studies*, 67:137–146, 2003.
- [15] Thomas Schelling. *Micromotives and Macrobehavior*. Norton, London: Penguin, 1978.
- [16] Yoav Shoham and Moshe Tennenholtz. On the emergence of social conventions: Modeling, analysis, and simulations. *Artif. Intell.*, 94(1-2):139–166, 1997.
- [17] M. Tennenholtz. Convention Evolution in Organizations and Markets. *Computational and Mathematical Organization Thoery*, 2(4):259–283, 1996.

- [18] Taiki Todo, Atsushi Iwasaki, Makoto Yokoo, and Yuko Sakurai. Characterizing false-name-proof allocation rules in combinatorial auctions. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 265–272, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.
- [19] Liad Wagman and Vincent Conitzer. Optimal false-name-proof voting rules with costly voting. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 1*, pages 190–195. AAAI Press, 2008.
- [20] Peyton Young. *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*. Princeton University Press, 1998.