Lottery Trees: Motivational Deployment of Networked Systems

John R. Douceur
Microsoft Research
Redmond WA 98052
johndo@microsoft.com

Thomas Moscibroda
Microsoft Research
Redmond WA 98052
moscitho@microsoft.com

ABSTRACT

We address a critical deployment issue for network systems, namely motivating people to install and run a distributed service. This work is aimed primarily at peer-to-peer systems, in which the decision and effort to install a service falls to individuals rather than to a central planner. This problem is relevant for bootstrapping systems that rely on the network effect, wherein the benefits are not felt until deployment reaches a significant scale, and also for deploying asymmetric systems, wherein the set of contributors is different than the set of beneficiaries. Our solution is the lottery tree (lottree), a mechanism that probabilistically encourages both participation in the system and also solicitation of new participants. We define the lottree mechanism and formally state seven properties that encourage contribution, solicitation, and fair play. We then present the Pachira lottree scheme, which satisfies five of these seven properties, and we prove this to be a maximal satisfiable subset. Using simulation, we determine optimal parameters for the Pachira lottree scheme, and we determine how to configure a lottree system for achieving various deployment scales based on expected installation effort. We also present extensive sensitivity analyses, which bolster the generality of our conclusions.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*network management*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*collaborative computing*; J.4 [**Social and Behavioral Sciences**]: *economics, psychology, sociology*; K.5.2 [**Legal Aspects of Computing**]: Governmental Issues—*regulation*

General Terms

Algorithms, Economics, Human Factors, Legal Aspects, Theory

Keywords

Incentive systems, networked systems, deployment, bootstrapping, lotteries, prospect theory, desiderata, impossibility results

1. INTRODUCTION

Network protocols, distributed systems, and communication overlays require several critical qualities to achieve deployment: They must be effective at their intended goal, compatible with existing infrastructure, robust to failures, secure against attack, incrementally deployable, scalable, and so forth. Yet in addition to these much-studied aspects, networked systems must also be attractive to the people who are needed to deploy them.

For a central planner, the end goal itself may be sufficient motivation to deploy; for example, an AS may be motivated to deploy a new intra-domain routing protocol that promises to improve resource efficiency. However, many of the most interesting networked systems proposed in recent years are intended to be deployed on end hosts, which are under the control of individuals. Some of these systems are asymmetric, in the sense that the participants contribute resources or effort to the system but receive nothing directly in return. Other systems, although symmetric insofar as the contributors are also the benefactors, rely on the network effect [20] to make the benefit of the system significant.

Symmetric network-effect systems, such as recommendation networks [13], file-sharing services [12], social forums [1], open databases [24], or collaborative reference works [3], can become self-sustaining when the scale becomes large enough for the benefit of participation to outweigh the cost. However, such systems are notoriously difficult to bootstrap, as evidenced by the numerous developed peer-to-peer systems [5], few of which have become popular.

Asymmetric distributed systems, such as BOINC [4], GPU [25] and Folding@Home / Genome@Home [21], are even more problematic. Because potential contributors are asked to provide computation, storage, or bandwidth toward a goal that does not directly benefit them, they have little or no incentive to join the system. Evidently, some people do choose to contribute, for various reasons including a selfless desire to help [10], a hope that the work may eventually benefit them [17], the "geek chic" associated with high contribution levels displayed on public ranking sites [7], and even the meager value of looking at pretty pictures on a screensaver [30]. Once such systems reach a threshold of popularity, they seem able to sustain substantial ongoing contribution. Following the principle of "a crowd draws a crowd," the media attention and buzz that accompanies a large congregation can inspire others to join. In most cases, however, potentially useful systems languish in unpopularity [7], having never managed to inspire a critical mass of participants.

The key problem that prevents a large number of symmetric and asymmetric networked systems from ever becoming popular is bootstrapping, i.e., attracting a sufficiently large initial user base. Two motivational challenges confront bootstrapping such systems. First, participants might reasonably expect their investment of effort and resources to return some palpable value, which neither asymmetric systems nor small network-effect systems provide. The simple expedient of monetarily compensating early adopters may not be a

SIGCOMM'07, August 27–31, 2007, Kyoto, Japan.

practical option, particularly for small research groups whose limited budgets may be vastly insufficient to compensate contributors at a level that many would find satisfactory.

Second, participants have little or no incentive to persuade their friends and acquaintances to join. Even for network-effect systems, wherein the value of the system grows as the population grows, the marginal benefit provided by each new participant is diffusely spread among the entire pool of participants, rather than accruing significantly to the person who solicited the new member. Thus, there is no inherent incentive that fosters system propagation.

This paper addresses these two challenges with a general mechanism for motivating bootstrap deployment of networked systems. The mechanism, which we call *lottery trees* (*lottrees*), employs the leverage of lottery psychology [29] to disproportionally motivate people to contribute to a developing system. In addition, lottrees employ a mechanism similar to a multilevel marketing scheme [9] to motivate participants to solicit other people to contribute as well. Consequently, lottrees can significantly increase the rate of network deployment and/or reduce the financial investment required to ensure rapid and eventually self-sustaining growth.

Our impetus for developing lottrees is an asymmetric distributed system we are currently building, which involves participation from a large number of geographically dispersed home computers. It did not take us long to realize that the lack of direct benefit to participants, the severe limitations of our budget, and the absence of effective and economical advertising would call for a creative solution to motivate participation, particularly when we observed the lackluster fate of so many similar projects [7].

Interestingly, there exists little or no literature in the networking community that addresses incentive mechanisms for motivational deployment. Previously researched incentive mechanisms [8, 11, 18, 19, 23, 27, 28, 31, 32, 36] operate on the premise that people participate in a networked system if the *utility* they receive from the system is higher than the *cost* of joining the system. Such mechanisms are inherently unsuited for bootstrapping asymmetric or small-sized systems from which the users get little or nothing in return. Our lottree mechanism thus marks a fundamental departure from existing incentive mechanisms in that it incentivizes participation even in these systems. Consequently, neither our formal definitions nor our theoretical proofs rely at all on the notoriously hard-to-define notion of *utility*. Even our simulation studies employ only a weak notion of comparative value, namely the "time value of money." We further note that our theoretical results apply to the full lottree mechanism, not merely to an abstracted or simplified model.

The following section describes the general lottree mechanism, including definitions we will use throughout the paper. In Section 3, we formally state seven desirable properties for a lottree, which collectively encourage participation, propagation, and fair play. Section 4 introduces some simple lottree schemes that illustrate the challenges involved in achieving our desired properties. Section 5 then presents Pachira, which is the strongest lottree scheme we have developed. Although Pachira satisfies only five of our desired properties, Section 6 proves that these five constitute a maximal satisfiable subset, insofar as any scheme satisfying these five properties cannot also satisfy the remaining two. Section 7 uses simulation to determine optimal parameters for the Pachira lottree scheme, to derive configuration parameters for specific lottree deployments, and to evaluate the sensitivity of our results to our various modeling assumptions. Section 8 addresses the relevant legal issues involved in using lottree schemes for motivational deployment of networked systems. Finally, Sections 9 and 10 present related work and our conclusions.

2. LOTTERY TREES

A *lottery tree* (*lottree*) is a mechanism that employs a lottery to probabilistically compensate people who participate in a networked system and/or who successfully encourage others to join the system as well and contribute to it. Depending on the specific networked system under consideration, contributing to or participating in this system can mean such different things as performing computation, storing information, transmitting data, testing a software application, providing recommendations, and so forth.

Regardless of the specific nature of the contributions, lottrees work as follows. Assume that there is an *executive entity* (a person, company, or research group) whose goal it is to deploy a networked system for which it needs to attract a large number of participants with sufficiently high contribution. We further assume that this executive entity of a network is willing and able to invest a certain amount of money (or any other item of value)—which we term the *payout*—for attracting a sufficient user base of this network. The function of the lottree is then, after a certain amount of time has passed, to select one contributor of the network as the recipient of the payout.[1] Ideally, a good lottree performs this selection in such a way that encourages high participation, contribution, and solicitation among participants.

More precisely, consider the network to be initialized with a single *root* node which represents the executive entity. Whenever a new person joins the network, he does so as a child of some person that is already a node in the system. For example, people might sign up their computers to the network by visiting a web site that records information and installs an application. If someone visits the site on his own, his computer joins as a child of the root. Once a member, he is able to send solicitations, perhaps in the form of coded email links, to friends and associates. Anyone who follows the coded link to the web site will join as a child of the member who sent the link, whom we call the *solicitor*. After the system has grown to a size that the executive entity judges to be sufficient, she farms out work units to the nodes and records each node's contribution. The lottree then selects a winner based on the tree structure and on nodes' contributions.

The challenge in designing a lottree scheme is how to define the rules of selecting a winner in such a way that encourages both contribution and system growth. Simple schemes that readily provide some benefits tend to fail to provide others. For example, an obvious scheme is employ a simple lottery that randomly selects a winner in proportion to its contribution to the network. Although this encourages contribution, it discourages participants from soliciting others, since any new member decreases the current members' chances of winning. What we require is a scheme that encourages contribution, solicitation, and fair play.

2.1 Definitions

Each participant that joins a lottree is represented as a tree node, and a directed edge from a node u to node v indicates that u was v's solicitor. Let T_r denote a tree rooted at node r. Formally, we represent a tree T as a set containing nodes n and ordered node-pairs (p, c) that indicate parent-child edges. This representation allows trees to be partially ordered using subset and superset relations. Standard tree properties are assumed to hold. We generalize the notation for a forest F_R, constructed as a union of independent trees, wherein R is the set of roots of the trees.

The following operators on trees are used in the paper: $Sub(T, n)$ is the subtree of T rooted at node n; $Path(T, n)$ is the set of nodes

[1]Alternatively, a lottree may periodically select a winner or may opt for choosing multiple winners in each period. All these mechanisms fall into the realm of possible lottery-tree strategies.

on the path from node n to the root of tree T following edges backwards, including the root, but excluding n; and $Parent(T, n)$ indicates node n's parent in T. The set of nodes in tree T is denoted by $\mathcal{N}(T)$ and the set of edges by $\mathcal{E}(T)$.

A crucial ingredient of lottrees is that every participant has a certain amount of measurable contribution. Formally, we model this *contribution* using a contribution function $C(n)$ that maps each node n to the non-negative sum of its accumulated contribution; larger values of $C(n)$ indicate greater contributions of resources to the system (e.g., more recommendations submitted, more computing cycles offered, etc.). For a set of nodes N, we use the notational shortcut $C(N) := \sum_{n \in N} C(n)$, and for a tree, $C(T) := C(\mathcal{N}(T))$.

Although different lottrees may differ in both functionality and implementation, they have in common that they select one or more lottery winners based on the topology of the tree (solicitations) as well as the contribution by individual participants. Hence, based on these commonalities, we formalize a *lottree* as a function $L(T, C, n)$ that for each node $n \in \mathcal{N}(T)$ in a tree T and a contribution function C, determines node n's *expected value*, i.e., the value that it gains from the lottery in expectation. In the sequel, it is convenient to normalize these values such that, $\sum_{n \in \mathcal{N}(T)} L(T, C, n) = 1$. Finally, throughout the paper, we denote the entire lottree, the so-called *system tree*, by T_S, and the root of the system tree is called Sys.

3. DESIDERATA

As alluded to at the end of Section 2, a lottree scheme should achieve diverse, and sometimes opposing, goals. While a lottree's main objective is to provide incentive to contribute and to solicit new participants, it should also maintain a notion of fairness and be robust against various forms of strategic behavior by participants. With these goals in mind, this section formalizes seven properties that are desirable in a lottery tree. Collectively, these properties encourage contribution to the system, encourage solicitation of new nodes, inhibit certain forms of gaming the system, and address practical considerations. We begin with a very simple property that expresses that every participant should have an interest in contributing more resources to the system.

Continuing contribution incentive (CCI):
A lottree L satisfies CCI if it provides nodes with increasing expected value in response to increased contribution. This encourages nodes to continue contributing to the system.

If node m is in the system tree: $m \in \mathcal{N}(T_S)$
and m's contribution increases: $C'(m) > C(m)$
and all other nodes maintain the same level of contribution:
$\quad \forall n \neq m : C'(n) = C(n)$
Then m's expected value increases:
$\quad L(T_S, C', m) > L(T_S, C, m)$

Value proportional to contribution (VPC):
Intuitively, we believe that participants are more likely to contribute to the system if they perceive the payout distribution to be fair relative to their contributions. We say that a lottree L satisfies φ-VPC for some $\varphi > 0$ if it ensures that each node's expected value is at least φ times the relative contribution made by that node.

If m is in the system tree: $m \in \mathcal{N}(T_S)$
and m contributes fraction c_m of all contribution:
$\quad c_m = C(m)/C(T_S)$

Then: $L(T_S, C, m) \geq \varphi\, c_m$

Strong solicitation incentive (SSI):
To encourage system growth, participants should have an incentive to solicit new participants. Formally, we say that a lottree L satisfies SSI if a node's expected value increases when that node gains a contributing descendent. This encourages nodes to solicit new nodes to join their subtrees, which is key in ensuring the growth of the overall system.

If node m is in the system tree: $T_m \subset T_S$
and m's subtree includes some node p: $p \in \mathcal{N}(T_m)$
and there is a new node n: $n \notin \mathcal{N}(T_S)$ with $C(n) > 0$
and which joins the system as a child of p:
$\quad T'_S = T_S \cup \{n, (p, n)\}$
Then m's expected value increases:
$\quad L(T'_S, C, m) > L(T_S, C, m)$

Weak solicitation incentive (WSI):
Because SSI is difficult to satisfy, we introduce a slightly weaker solicitation property, WSI. This property is satisfied by a lottree L if, when a new contributing node joins the system, an existing node's expected value is greater if the new node becomes its descendent than if the new node joins elsewhere in the tree. This property promotes competition for new descendent nodes, which encourages solicitation.

If node m is in the system tree: $T_m \subset T_S$, $C(m) > 0$
and m's subtree includes some node p: $p \in \mathcal{N}(T_m)$
but does not include some other node q: $q \in \mathcal{N}(T_S \setminus T_m)$
and there is a new node n: $n \notin \mathcal{N}(T_S)$ with $C(n) > 0$
and which in case 1 joins the system as a child of p:
$\quad T'_S = T_S \cup \{n, (p, n)\}$
and which in case 2 joins the system as a child of q:
$\quad T''_S = T_S \cup \{n, (q, n)\}$
Then m's expected value is greater in case 1:
$\quad L(T'_S, C, m) > L(T''_S, C, m)$

Unprofitable solicitor bypassing (USB):
Besides attracting contribution and providing incentives for solicitation, lottrees must also be secure against different notions of strategic behavior of its participants. If, for instance, new nodes tend to join the system not as children of the nodes that solicited them, then participants will lose interest in soliciting new nodes. We thus introduce USB, which a lottree L satisfies if a new node can never gain expected value by joining as a child of someone other than its solicitor.

If nodes m and p are in the system tree: $\{m, p\} \subset \mathcal{N}(T_S)$
and there is a new node n that may eventually solicit its own
$\quad$ subtree of nodes: $T_n \cap T_S = \emptyset$
and which in case 1 joins the system as a child of m:
$\quad T'_S = T_S \cup T_n \cup \{(m, n)\}$
and which in case 2 joins the system as a child of p:
$\quad T''_S = T_S \cup T_n \cup \{(p, n)\}$
Then n's expected value is no greater in case 2:
$\quad L(T'_S, C, n) \geq L(T''_S, C, n)$
which, by symmetry, implies: $L(T'_S, C, n) = L(T''_S, C, n)$

Unprofitable Sybil attack (USA):
An equally important property is that no participant can increase its odds by pretending to have multiple identities. That is, a lottree L satisfies USA if a node does not gain expected value by joining the system as a set of Sybil nodes [14] instead of joining singly. (This formalism employs Hilbert's ε operator. $\varepsilon x : P(x)$ means "choose some x that satisfies $P(x)$.")
If the system tree contains node p and node set Q:
$\quad \{p\} \cup Q \subset \mathcal{N}(T_S)$
and there is a new node n: $n \notin \mathcal{N}(T_S)$
which can appear as a new node set S: $S \cap \mathcal{N}(T_S) = \emptyset$

wherein S's aggregate contribution does not exceed n's contribution: $C(S) \leq C(n)$
and n may eventually solicit a forest F_H of other nodes: $F_H \cap T_S = \emptyset$
and in case 1, n joins the system as a child of p:
$$T'_S = T_S \cup \{n, (p, n)\} \cup \{(n, h) : h \in H\} \cup F_H$$
and in case 2, S joins as descendents of Q:
$$T''_S = T_S \cup S \cup \{(\varepsilon q : q \in Q \cup S, s) : s \in S\} \cup \{(\varepsilon s : s \in S, h) : h \in H\} \cup F_H$$
Then n's expected value is no greater in case 2:
$$L(T'_S, C, n) \geq \textstyle\sum_{s \in S} L(T''_S, C, s)$$

Zero value to root (ZVR):
A lottree L satisfies ZVR if the expected value to the root of the system tree is zero. In a practical lottree, the prize value should be disbursed to participants and contributors, not retained by the system: $L(T_S, C, Sys) = 0$. (Clearly, ZVR is impossible to satisfy in the degenerate case in which the root has no children.)

Discussion: Each of the above seven properties captures a specific important characteristic that an ideal lottree scheme should fulfill in order to robustly motivate significant participation. We further believe (but cannot prove) that these properties collectively characterize a lottree that would be ideal for practical use.

As a possible criticism of our formal statement of these properties, one might argue that when a person decides whether to join a specific lottree system or whether to solicit an acquaintance, he is unlikely to be guided by a rigid and detailed verification of properties such as SSI or USB. However, we believe that strictly satisfying these properties is of real importance for practical deployments, for the simple reason that lottrees involve the transfer of money. In any such system, issues of trust and security are of utmost importance. There should be no way of increasing one's odds by circumventing the rules. Solicitation properties like SSI and WSI are crucial as well, especially as we consider deployment scenarios in which purely altruistic motivations for joining have often been insufficient to yield large deployments.

4. SIMPLE LOTTREE SCHEMES

It might seem that the properties enumerated in Section 3 should be fairly trivial to satisfy. To demonstrate that this is not the case, this section constructs two fairly simple lottree schemes, and we show that they fail to satisfy several important properties.

4.1 The PS (proportional selection) lottree

We first consider a very simple lottree scheme, which does not account for any solicitation structure and simply selects a winning node based on its own contribution. The *PS* (*proportional selection*) lottree scheme selects each participant $n \in \mathcal{N}(T_S)$ to be the winner with odds of $o_n = C(n)/C(T_S)$, regardless of the solicitation structure.

While providing optimal fairness (1-VPC) and robustness against various forms of gaming (satisfying USB and USA, for instance), it fails to provide any incentive for nodes that have already joined the lottree to solicit new members. It thus clearly violates both weak and strong solicitation-incentive properties (WSI and SSI).

4.2 The Luxor lottree

We next present the Luxor lottree scheme, which—unlike the PS scheme—provides a solicitation incentive to nodes in the tree. Although it is more involved than the PS scheme, it is a relatively straightforward extension wherein each node passes some of its win odds up to its parent.

Algorithm 1 The Luxor lottree - Winner Selection

Input: A lottree T_S with N peers. $C(n)$ denotes the contribution of a peer $n \in \mathcal{N}(T_S)$.
Two parameters $0 \leq \mu, \rho \leq 1$.
Output: A winner $\hat{n} \in \mathcal{N}(T_S)$ that wins the lottery.

```
1:  n̂ := ∅.
2:  Set w(n) := C(n)/C(T_S) for each n ∈ N(T_S).
3:  Randomly select one peer m from N(T_S) such that the probability of selecting peer n is w(n).
4:  With probability μ, set n̂ := m and stop.
5:  cur := Parent(T_S, m).
6:  while n̂ = ∅ and cur ≠ Sys do
7:      With probability ρ, set n̂ := cur and stop.
8:      cur := Parent(T_S, cur);
9:  end while
10: if n̂ := ∅ then n̂ := Sys.
```

Winner selection in the Luxor lottree, characterized by two parameters μ and ρ, proceeds in two passes. First, it randomly selects a node $m \in \mathcal{N}(T_S)$ in proportion to its contribution, just as in the PS lottree scheme. However, m is merely a *candidate*; it only becomes the winner with probability μ. With probability $1 - \mu$, the winner is one of m's ancestors. As shown in Algorithm 1, Luxor moves incrementally up the path $Path(T_S, m)$ from $Parent(T_S, m)$ to the root Sys, letting each successive candidate cur win the lottery with probability ρ. Upon selection of a winner $\hat{n}$, the process stops.

The parameter μ can be used to tune the tradeoff between solicitation incentive and fairness. Increasing μ increases fairness at the expense of decreasing solicitation incentive.

Algorithm 1 procedurally describes the Luxor scheme. We can also describe Luxor by formally defining its lottree function L_L as

$$L_L(T_S, C, n) = \mu \cdot w(n) + \sum_{\substack{z \in \mathcal{N}(T_n), \\ z \neq n}} w(z) p_{nz},$$

where $p_{uv} := P[\hat{n} = u | m = v]$ denotes the probability that node u wins the lottery conditioned on the event that node $v \neq u$ was initially selected as the candidate. Letting d_{uv} be the hop-distance between two nodes u and v, p_{uv} is defined by

$$p_{uv} = \begin{cases} (1-\mu)(1-\rho)^{d_{uv}-1}\rho & , u \in Path(T_S, v) \\ 0 & , \text{otherwise.} \end{cases}$$

The Luxor lottree scheme satisfies several desirable properties, as stated in Theorem 4.1. The proof is omitted due to lack of space.

THEOREM 4.1. *The Luxor lottree scheme satisfies properties CCI, WSI, USB, and φ-VPC for $\varphi = \mu$. The scheme also satisfies SSI unless for some node n, there exists a node $z \in \mathcal{N}(T_n)$ for which $p_{nz} \geq L(T_S \setminus \{(Parent(T_S, z), z)\}, C, n)$.*

The previous theorem characterizes scenarios for which the Luxor scheme satisfies the SSI property. We will later present Theorem 6.1, a consequence of which is that there must exist scenarios for which the Luxor scheme does not satisfy SSI. In addition, the following theorem states that this scheme fails to satisfy two additional properties.

THEOREM 4.2. *The Luxor scheme violates USA and ZVR.*

PROOF. It is clear that ZVR is violated because there is a nonzero probability that the root is selected as the winner. The interesting property is USA. Consider Figure 1 and assume node z is capable of contributing a total of $C(z)$ and joins as a child of n. Its expected value is therefore $\mu \cdot C(z)/C(T_S)$. In contrast, if z launches

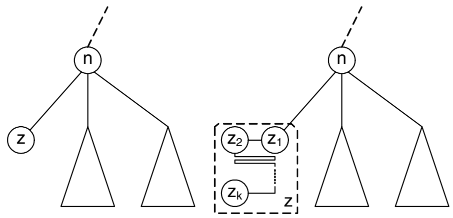

Figure 1: The Luxor scheme is vulnerable to Sybil attacks. $\mathbf{z}$ increases its expected value by splitting its contribution among Sybil nodes $\mathbf{z_1}, \ldots, \mathbf{z_n}$.

a Sybil attack by splitting itself up into two (or more) nodes z_1 and z_2 and divides its contribution such that $C(z_1)+C(z_2) = C(n)$, it can increase its expected value. Specifically, z_1 joins as a child of n and z_2 becomes a child of z_1. That way, the combined expected value $L(T_S, C, z_1) + L(T_S, C, z_2)$ exceeds $L(T_S, C, z)$, because of $p_{z_1 z_2} > 0$. In the extreme case, a new node z could split itself up into a large number of Sybil nodes $z_1, \ldots, z_k$, arranged in form of a large chain, and have $C(z_1) = \ldots = C(z_{k-1}) = 0$ and $C(z_k) = C(z)$. In this case, the cumulated expected gain of z reaches $C(z)/C(T_S)$, which is by a factor $1/\mu$ larger than if it had joined as a single node. □

The fact that Luxor does not satisfy USA and is thus not robust against Sybil attacks is particularly problematic, because it encourages gaming behavior, which can significantly undermine people's trust in the system. Since lottrees distribute money in return for participation, this lack of trust could decrease people's willingness to participate. We address this problem in the following section by presenting the *Pachira* lottree scheme, which is provably robust against Sybil attacks.

5. THE PACHIRA LOTTREE

This section introduces a general and practical lottree scheme called Pachira, which satisfies all properties satisfied by the Luxor scheme, but additionally satisfies the USA property, which Luxor fails (Theorem 4.2).

5.1 Theoretical Underpinnings

The Pachira lottree has two input parameters β and δ that trade off solicitation incentive against fairness. In its general version, the Pachira lottree is defined using a function $\pi(c)$ defined on $[0, 1]$ with the following characteristics:

I) $\pi(0) = 0, \quad \pi(1) = 1$

II) $\forall c \in [0, 1] : \frac{d\pi(c)}{dc} \geq \beta$ (minimum slope of β)

III) $\forall c \in [0, 1] : \frac{d^2\pi(c)}{dc^2} > 0$ (strictly convex)

The following two inequalities directly follow from the strict convexity of $\pi(c)$. First, for any $c_1 > c_2$ and $\epsilon > 0$,

$$\pi(c_1 + \epsilon) - \pi(c_1) > \pi(c_2 + \epsilon) - \pi(c_2). \tag{1}$$

Secondly, it holds that

$$\pi\left(\sum_{c_i \in \mathcal{C}} c_i\right) \geq \sum_{c_i \in \mathcal{C}} \pi(c_i). \tag{2}$$

In principle, the Pachira lottree can be defined using any function π that follows the above mentioned properties. In the sequel, we are going to use the following particularly convenient and intuitive function with these characteristics:

$$\pi(c) = \beta\, c + (1 - \beta) c^{1+\delta}, \tag{3}$$

Algorithm 2 The Pachira lottree - Winner Selection

Input: A lottree T_S with N peers. $C(n)$ denotes the contribution of a peer $n \in \mathcal{N}(T_S)$.
Two parameters $0 \leq \beta, \delta \leq 1$.
Output: A winner $\hat{n} \in \mathcal{N}(T_S)$ that wins the lottery.

```
Compute C(T_S) = Σ_{m∈N(T_S)} C(m)
for each n ∈ N(T_S) in post-order of T_S do
    Compute C(Sub(T_S, n)) by summing up C(n)
    and C(Sub(T_S, m)) for all children m of n.
    Compute W(T_S, C, Sub(T_S, n)) using (3) and (4).
    Compute L_P(T_S, C, n) according to (5).
end for
Select n̂ randomly such that every node is selected with probability L_P(T_S, C, n).
```

where β and $\delta > 0$ are the input parameters of Pachira. Our scheme makes use of this function in the following way: Each node in the tree computes its *weight* as the function π applied to the node's proportional contribution. Formally, for tree T and contribution function C, the weight $W(T, C, n)$ of a node n is $W(T, C, n) = \pi(C(n)/C(T))$. Also, the weight for a subtree $Sub(T, n)$ is defined as

$$W(T, C, Sub(T, n)) = \pi\left(\frac{C(Sub(T, n))}{C(T)}\right). \tag{4}$$

Finally, notice that for any leaf node n, it holds that $W(T, C, n) = W(T, C, Sub(T, n)) = \pi(C(n)/C(T))$.

The Pachira lottree scheme proceeds as follows. Each node $n \in \mathcal{N}(T_S)$ is assigned an expected value, L_P, defined as the weight of the subtree rooted at n minus the weights of all child subtrees of n. Formally,

$$\begin{aligned} L_P(T, C, n) &= W(T, C, Sub(T, n)) \\ &- \sum_{(n,m) \in \mathcal{E}(T)} W(T, C, Sub(T, m)). \end{aligned} \tag{5}$$

Notice that in general, $L_P(T, C, n) \neq W(T, C, n)$, i.e., a node's expected value is different from its weight.

As we show in the following section, this theoretical formulation of the Pachira lottree scheme easily lends itself to efficient implementation, which renders the scheme a good candidate for practical use in a variety of networked systems.

5.2 Implementation

The Pachira lottree scheme can be implemented and its winner computed in a straightforward way. Besides summing up all contributions, a single post-order traversal of the tree suffices to assign winning probabilities to each node. The details of the selection scheme are presented in Algorithm 2.

The algorithm first sums up the contributions of all nodes. It then performs a post-order traversal of the tree, considering each node only after computing results for the node's children. For each node n, Pachira first computes the total contribution $C(Sub(T_S, n))$ of n's subtree. Next, it computes the *weight* $W(T_S, C, Sub(T_S, n))$ of the subtree rooted at n by applying the function π to the ratio $C(Sub(T_S, n))/C(T_S)$. And last, it computes n's expected value $L_P(T_S, C, n)$ by taking the subtree weight $W(T_S, C, Sub(T_S, n))$ and subtracting from it the weight of n's children's subtrees (cf (5)). Once all expected values $L_P(T_S, C, n)$ are computed, the winner is selected in proportion to the expected values.

Because Pachira's winner-selection mechanism requires only a single bottom-up traversal of the tree, its running time is linear in the number of participating nodes. Computational complexity is thus not a significant impediment to practical use of a Pachira lottree. We address other practical issues in Sections 8 and 10.

5.3 Rescaling

The Pachira lottree does not satisfy ZVR, because the root node Sys may be selected as the winner. A deceptively simple solution to this problem is to re-run the winner-selection algorithm until a non-root node is selected. This is equivalent to *rescaling* the lottree by distributing the root's winning probability among the other nodes in proportion to their winning probabilities. Formally, letting l_n refer to the expected value $L_P(T_S, C, n)$ for any node n, the win odds o_n thus become

$$o_n = \begin{cases} 0 & , n = Sys \\ l_n/(1 - l_S) & , \text{otherwise.} \end{cases}$$

However, if the win odds are thus rescaled but the payout is left unchanged, the modified lottree will violate the USB property. To see why this is so, note that when a new node joins the system, although its location in the tree does not affect its own expected value, its location does affect the root's expected value, because if it joins a heavily weighted subtree, it will pull more weight away from the root than if it joins a lightly weighted subtree, due to convexity of the weight function π. Because rescaling distributes the root's expected value among the other nodes, a node can game the system by deliberately joining a lightly weighted subtree (for example, joining as a child of the root), rather than joining as a child of its solicitor. This leaves more win probability for the root, which when distributed among all other nodes, increases the newly joining node's expected value.

This violation of USB can be avoided by rescaling the payout amount to keep the expected values unchanged. This is achieved by multiplying the payout by a factor of $(1 - l_S)$. The practicality of this approach is limited by whether the payout is something (such as money) that can be arbitrarily rescaled, and by other issues as described in Section 8.

5.4 Analysis

We begin by proving an important lemma that states that the weight $W(T, C, n)$ of a node n is a lower bound for its expected value.

LEMMA 5.1. *It holds for all T, C, and $n \in \mathcal{N}(T)$ that*

$$L_P(T, C, n) \geq W(T, C, n).$$

PROOF. The property follows from $\pi(c)$'s convexity. First, it follows from the definition (5) of $L_P(T, C, n)$ that if n is a leaf in T, then $L_P(T, C, n) = W(T, C, n)$. For every n, it holds

$$\begin{aligned} L_P(T,C,n) &= W(T,C,Sub(T,n)) - \sum_{(n,m)\in\mathcal{E}(T)} W(T,C,Sub(T,m)) \\ &= \pi\left(\frac{C(Sub(T,n))}{C(T)}\right) - \sum_{(n,m)\in\mathcal{E}(T)} \pi\left(\frac{C(Sub(T,m))}{C(T)}\right) \\ &\geq \pi\left(\frac{C(n)}{C(T)}\right), \end{aligned}$$

where the inequality follows from the convexity Inequality (2) and the fact that $\frac{C(Sub(T,n))}{C(T)} = \sum_{(n,m)\in\mathcal{E}(T)} \frac{C(Sub(T,m))}{C(T)} + \frac{C(n)}{C(T)}$. This concludes the proof. □

Based in part on this lemma, we can now precisely characterize the set of desirable desiderata properties that are satisfied by the Pachira lottree scheme. We begin with the simplest one and show that Pachira always incentivizes increasing contribution.

LEMMA 5.2. *Pachira satisfies CCI.*

PROOF. Assume a node m increases its contribution, while all other contributions in the tree remain the same. The relative contribution $C(Sub(T_S, n))$ increases and, because $\pi(c)$ has positive slope (property II of π's definition), the weight of m's subtree increases. Conversely, the weights of m's children's subtrees (if it has any) decrease. It then follows by the definition of L_P that $L_P(T_S, C', m) > L_P(T_S, C, m)$. □

The following lemma follows immediately from Lemma 5.1 and shows that Pachira achieves provable fairness bounds.

LEMMA 5.3. *Pachira satisfies φ-VPC for $\varphi \geq \beta$*

PROOF. Let $c_m = C(m)/C(T_S)$. By Lemma 5.1 and the definition of $\pi(c)$, we obtain

$$L_P(T_S, C, m) \geq W(T, C, m) = \pi(c_m) \geq \beta\, c_m. \qquad \square$$

LEMMA 5.4. *Pachira satisfies WSI.*

PROOF. Recalling the definition of the WSI property, let m be a node and let a be one of m's children (if any exist, otherwise, $a = m$). Suppose that there is a node n that newly joins the lottree, either as a child of a node $p \in \mathcal{N}(T_a)$ or as a child of a node $q \in \mathcal{N}(T_S \setminus T_m)$ that is not in m's subtree. T'_S and T''_S denote the resulting trees when the new node n joins as a child of p or q, respectively. Finally, we use the following notational abbreviations:

- let $c_n := C(n)/C(T'_S)$ and $c_m := C(m)/C(T'_S)$
- let $c_A := C(T_a)/C(T'_S)$
- let $c_Z := C(T_m \setminus (\{m\} \cup T_a))/C(T'_S)$; that is, c_Z is the total contribution of all nodes in those subtrees of m that n does *not* join
- let Z be the set of m's children other than a; formally, $Z := \{z \mid (m, z) \in \mathcal{E}(T_S) \wedge z \neq a\}$
- let $w_Z := \sum_{z\in Z} \pi(C(T_z)/C(T'_S))$ be the total weight of all subtrees rooted at children of m other than T_a

With these definitions, we can now express the expected value of m both in case n joins a subtree of m, and otherwise. In both cases, we use Equality (5) and plug in (4).

$$\begin{aligned} L_P(T'_S, C, m) &= \pi(c_m + c_n + c_A + c_Z) \\ &\quad - \pi(c_n + c_A) - w_Z \\ L_P(T''_S, C, m) &= \pi(c_m + c_A + c_Z) - \pi(c_A) - w_Z. \end{aligned}$$

Clearly, it holds that $C(T'_S) = C(T''_S)$. Hence, when substituting $c_1 = c_n + c_A$, $c_2 = c_A$, and $\epsilon = c_m + c_O$, we can write the increase Δ of m's expected value if p joins its own subtree (as opposed to someone else's subtree) as

$$\begin{aligned} \Delta &= L_P(T'_S, C, m) - L_P(T''_S, C, m) \\ &= \pi(c_1 + \epsilon) - \pi(c_1) - (\pi(c_2 + \epsilon) - \pi(c_2)). \end{aligned}$$

From (1), it follows that $\Delta > 0$ and hence, $L_P(T'_S, C, m) > L_P(T''_S, C, m)$. □

LEMMA 5.5. *Pachira satisfies USB.*

PROOF. The claim can easily be verified by observing that for any $n \in \mathcal{N}(T_S)$, the expected value $L_P(T, C, n)$ is independent of the structure of tree T outside of n's subtree T_n. Hence, the initial position in the tree is irrelevant. □

Unlike the Luxor scheme, the Pachira lottree is robust against Sybil attacks:

LEMMA 5.6. *Pachira satisfies USA.*

PROOF. We must show that a node does not increase its expected value by joining as multiple nodes, even when these Sybil nodes form subtrees among each other and join as such (like the chain in Figure 1). Assume that a new node z joins the lottree. Alternatively, z can join as a set of Sybil nodes $Z = \{z_1, \ldots, z_k\}$ such that $\sum_{z_i \in Z} C(z_i) \leq C(z)$. Let T'_S and T''_S be the resulting trees in the former and latter cases, respectively. If all nodes in Z join as independent nodes, the expected value of z is

$$\begin{aligned} L(T'_S, C, z) &= \beta \sum_{z_i \in Z} c_{z_i} + (1-\beta) \left(\sum_{z_i \in Z} c_{z_i} \right)^{1+\delta} \\ &\geq \beta \sum_{z_i \in Z} c_{z_i} + (1-\beta) \sum_{z_i \in Z} c_{z_i}^{1+\delta} \\ &= \sum_{z_i \in Z} L(T'_S, C, z_i), \end{aligned}$$

which proves the lemma in this case.

It remains to prove the case when nodes in Z join as a forest F_H with root set H instead of independent nodes. The key ingredient in the proof is that the cumulated expected value of all nodes in a subtree T_r with root r is always equivalent to the weight of T_r. Formally, this can be derived as

$$\begin{aligned} \sum_{s \in \mathcal{N}(T)} L(T_S, C, s) &= \sum_{s \in \mathcal{N}(T)} \Big(W(T_S, C, Sub(T_S, s)) \\ &\quad - \sum_{(s,s') \in \mathcal{E}(T_S)} W(T_S, C, Sub(T_S, s')) \Big) \\ &= W(T_S, C, Sub(T_S, r)), \end{aligned} \tag{6}$$

where the second equality stems from the fact that all terms, except for the one at r, cancel out. From this, it follows that the cumulated expected value of nodes in Z is

$$\sum_{z_i \in Z} L(T_S, C, z_i) = \sum_{h \in H} W(T_S, C, Sub(T_S, h)).$$

In other words, we can shrink each tree T_h consisting of nodes z_i into a single node z_h that has the same contribution as the entire tree before changing the expected value of nodes in Z. Since these shrunk z_h are now independent nodes, the proof is finished analogously to the case in which all nodes are independent. □

It is instructive to consider the above proof in relation to the example given in Figure 1 in which the Luxor lottree proved to be vulnerable to Sybil attacks. In Luxor, the sum of the expected values of nodes in a tree T not only depends on the relative total contribution of nodes in T compared to the entire contribution $C(T_S)$. Instead, it also depends on the *topology* formed by nodes in T. The combined expected value of two nodes z_1 and z_2 joining the lottree as siblings $\{(n, z_1), (n, z_2)\}$ of a parent n is smaller than the same nodes joining the tree as child and grandchild, $\{(n, z_1), (z_1, z_2)\}$ of n. Instead, Equality (6) proves that in Pachira the total expected value of nodes is always equal to the weight of the subtree. It is this additional property that prevents Sybil attacks.

6. IMPOSSIBILITY RESULTS

The Pachira lottree scheme satisfies the five desirable properties CCI, VPC, WSI, USB, and USA, thereby providing incentives to contribute to the system, to solicit new contributors, and to avoid attempts at gaming. However, Pachira fails to achieve both SSI and ZVR, which would also be desirable. An ideal lottree should simultaneously satisfy all mutually achievable desiderata.

In this section, we prove that Pachira does, in fact, satisfy *all* mutually achievable desiderata, in the sense that no lottree can satisfy any additional property without violating at least one of the properties that Pachira satisfies, which implies that these five properties constitute a maximal satisfiable subset.

The following theorem states that satisfying VPC precludes satisfying SSI.

THEOREM 6.1. *Given an arbitrary topology T_S, there is no lottree that simultaneously satisfies both SSI and φ-VPC, for any $\varphi > 0$, on T_S.*

PROOF. Consider an arbitrary tree T_S, and assume for contradiction that there is a lottree scheme that satisfies SSI and φ-VPC on T_S. The theorem holds for any distribution of the contributions among the nodes in T_S. Let m_i, $i = 0, \ldots, x$ be a sequence of nodes joining T_S. Node m_0 joints at an arbitrary node z and each subsequent new node m_i joins as a child of m_{i-1}. We define $P_z := Path(T_S, z)$ and denote by $C_O := \sum_{n \in \mathcal{N}(T_S \setminus P_z)} C(n)$ the total contribution of all nodes node on the path P_Z. Define the contribution of node m_i to be $2^i \cdot C(T_S)$. It follows from the fairness property that the first new node m_0 needs to get an expected value $L(T_S, C, m_0)$ of at least $\varphi/2$ because $C(m_0) = C(T_S)$. For the same reason, each subsequent new node m_i also must have an expected value $L(T_S, C, m_i) \geq \varphi/2$. As each new node is added as a child of the same path, the SSI property implies that the expected value of nodes on the path $Path(T_S, z)$ or any m_i must not decrease. Hence, after inserting node m_x for $x = \lceil 2C_O/\varphi \rceil + 1$, the total expected value of nodes $m_0, \ldots, m_x$ must be at least $(\lceil 2C_O/\varphi \rceil + 1) \cdot \varphi/2 > C_O$. Since C_O was the total expected value of nodes in $\mathcal{N}(T_S \setminus P_z)$, this implies that there must be at least one new node or a node on P_z whose expected value has decreased, which contradicts the SSI property. □

One might conceivably argue that SSI is a more important property than VPC, and so a preferable lottree would be one that satisfies the former at the expense of the latter, unlike Pachira. However, the simulations in Section 7—specifically, Figure 3 (right)—demonstrate that the absence of SSI can be ameliorated with even a moderately sized initial set of participants. By contrast, in Figure 2 (middle), as fairness (lower bounded by β) decreases, the effectiveness of the lottree also decreases. (This is not as apparent in the curves for small values of δ, because for small δ, Pachira satisfies φ-VPC for $\varphi \gg \beta$.)

The Pachira scheme also fails to satisfy ZVR, as does the Luxor scheme. This turns out to be unavoidable for both.

THEOREM 6.2. *There is no lottree that can guarantee the simultaneous satisfaction of WSI, USB, and ZVR.*

PROOF. A simple counterexample suffices for the proof. Consider the two systems $T_S = \{Sys, a, b, (Sys, a), (Sys, b)\}$ and $T'_S = \{Sys, a, b, (Sys, a), (a, b)\}$, and take any contribution function C for which $C(a) > 0$ and $C(b) > 0$. Assume for contradiction a lottree scheme L that satisfies WSI, USB, and ZVR. As shorthand, for any node n, let l_n refer to $L(T_S, C, n)$ and l'_n refer to $L(T'_S, C, n)$.

By construction, the expected values of each lottree must sum to one, i.e., $l'_a + l'_b + l'_S = l_a + l_b + l_S$. By ZVR, $l'_S = l_S = 0$, so $l'_a + l'_b = l_a + l_b$. By USB, $l'_b = l_b$, so $l'_a = l_a$. However, by WSI, $l'_a > l_a$, which is a contradiction. □

The unavoidable absence of ZVR is not a major problem. As described in Section 5.3, it is often possible to rescale the lottree to ensure that a non-root node wins, without changing the expected values to each node. In this case, the root has non-zero expected value because it retains a fraction of the payout that is disbursed to the winning node.

We can now state and easily prove our main theorem, which captures the optimality of the Pachira lottree scheme.

THEOREM 6.3. *Pachira achieves a maximal satisfiable subset of desirable properties.*

PROOF. By Lemmas 5.2–5.6, Pachira satisfies all desiderata except SSI and ZVR. By Theorem 6.1, SSI is incompatible with VPC; and by Theorem 6.2, ZVR is incompatible with WSI and USB. □

7. EVALUATION

In order to gain a better understanding of the solicitation and participation generated by the Pachira lottree scheme, we conduct extensive simulations. Our goals are:

- to derive good choices for the parameters β and δ
- to determine an appropriate payout amount based on target deployment scale and expected participation effort
- to determine the required count of initial participants to avoid problems from the absence of SSI
- to analyze the sensitivity of our evaluation to our modeling assumptions and hidden parameters

7.1 Simulation framework

We use a frame-based simulator with a finite population of computer users, each frame representing one day of simulated time. The simulation procedure is presented in Algorithm 3. In outline, the simulator first establishes a small subset of the population as an *initial set* (IS) of participants. Then, on each frame, each participant decides whether to solicit other users: If the perceived gain from soliciting outweighs the cost of sending a solicitation, the participant solicits a subset of acquaintances. Each solicited person first decides whether to even consider joining; if he does consider it, he evaluates the perceived gain from joining relative to the cost of joining, and joins if the gain outweighs the cost.

7.1.1 Challenges

Because human behavior is notoriously difficult to model, simulating a system involving humans is tremendously challenging. The above simulation description implies the need for answers to the following behavioral questions:

I) Which people are acquainted with which other people?

II) How do people perceive the benefit from the lottery?

III) Does each solicitor consider other solicitors to be in competition for new participants?

IV) How do people perceive the cost of soliciting others and the cost of joining the system?

V) How likely are people to even consider a solicitation?

VI) How much will each person contribute to the system?

7.1.2 Models

In our simulation, we deal with the above challenges by employing a set of theories and models that have been used and widely accepted in literature on economics and cognitive psychology.

I) Social Network Model $\mathfrak{G}$: We model the acquaintanceship of people using a social network model. Many such models have been proposed [26], broadly classifiable as either random graphs or evolving networks. The general consensus is that a model for social networks should exhibit short average path length, high clustering, broad degree distribution, and community structure. Our default model, which satisfies these properties, is an evolving network model proposed by Toivonen et al. [33]. In this model, network growth is governed by two processes: (1) attachment to random existing nodes and (2) attachment to the neighborhood of the selected random node. The model is characterized by three parameters, $pInit$, $Range$, and $Seed$, for which our default values are those specified by Toivonen et al. This yields an average degree of roughly five.

II) Payout Valuation Model $\mathfrak{V}_P$: To evaluate how people perceive the lottree payout, we use a model based on the cognitive psychology of lotteries and sweepstakes [29]. The generally accepted economic model that has replaced expected utility theory is *prospect theory*, proposed by Tversky and Kahnemann [34]. Based on empirical studies, it describes how individuals evaluate losses and gains in lotteries. It applies a nonlinear transformation of the probability scale, which over-weights small probabilities and under-weights moderate and high probabilities. For the model's two key parameters, α (power) and γ (probability weighting), we use the values derived by Tversky and Kahnemann in their refinement of prospect theory known as *cumulative prospect theory* [35].

Algorithm 3 Simulation procedure

Input: Parameters *STC*, *JTC*, *SAF*, *initSize*, and $\hat{\tau}$,
  Undirected social network graph $\mathfrak{G}$,
  Lottree scheme L and payout amount A_P,
  Valuation models for payout $\mathfrak{V}_P$ and time $\mathfrak{V}_T$,
  Diffusion model $\mathfrak{D}$ and contribution model $\mathfrak{C}$.
Output: Lottery tree T_S.
Initialize T_S to $\{Sys\}$
Select random node subset *IS* of size *initSize* from $\mathcal{N}(\mathfrak{G})$
for each $n \in \mathcal{N}(IS)$: add n to T_S as child of Sys
for each simulation frame **do**
  for each $n \in \mathcal{N}(T_S) \setminus \{Sys\}$ **do**
    Using L, A_P, and $\mathfrak{V}_P$, evaluate absolute and relative perceived gains from successfully soliciting new participants
    Using *SAF*, compute overall perceived gain
    Using $\mathfrak{V}_T$, evaluate solicitation time cost *STC*
    if perceived gain $>$ solicitation cost **then**
      Set $\theta =$ neighbors of n in $\mathfrak{G}$ s.t. $\theta \cap \mathcal{N}(T_S) = \emptyset$
      Set $\tau = min(|\theta|, \hat{\tau})$
      Select random subset M of size τ from θ
      for each $m \in M$ **do**
        Using $\mathfrak{D}$, decide whether m considers joining
        if m considers joining **then**
          Using L, A_P, $\mathfrak{V}_P$, and $\mathfrak{C}$, evaluate perceived gain from joining system
          Using $\mathfrak{V}_T$, evaluate join time cost *JTC*
          if perceived gain $>$ join cost **then**
            Add m to T_S as child of n
          end if
        end if
      end for
    end if
  end for
  advance simulation time by one day
end for

III) Solicitation Assumption Factor *SAF*: When a person evaluates the gain from soliciting an acquaintance, there are two alternatives the person might imagine would result from not making the solicitation. First, the non-member might remain not part of the system; second, the non-member might join the system as a child of someone else. The distinction between these two is essentially the distinction between the SSI and WSI properties, and the person's assumption about what would happen if he does not solicit has an effect on how he values the gain from the solicitation. We model this assumption with a *Solicitation Assumption Factor* (SAF), which expresses the believed likelihood that an acquaintance will join the tree even without a solicitation from the person making the evaluation. Our default value for *SAF* is 0.5.

IV) Time Valuation Model $\mathfrak{V}_T$: As a simple estimate for the costs of soliciting others and of joining the system, we characterize the efforts by the temporal cost of performing the task. The time required to send a solicitation is the *Solicitation Time Cost* (STC), and the time required to join the system is the *Join Time Cost* (JTC). For comparison against perceived values obtained through the payout valuation model, we must convert these temporal values into monetary values. For this, we employ a probability distribution of *income per minute* for each person, using the US income distribution in 2005 [2].

V) Diffusion Model $\mathfrak{D}$: To determine whether a person considers each solicitation, we employ a *diffusion model*, which characterizes the flow of influences through a social network. These models are built on the premise that a person's tendency to accept an idea (or, in our case, to consider joining a lottree) increases monotonically as it receives more recommendations (or solicitations). Our default model is the *independent cascade model* [16], wherein each solicitation succeeds with a certain fixed probability p. In our sensitivity analysis, we also consider two other models: the *diminishing cascade model* [6], wherein the success probability p decreases by a factor of $q < 1$ with each repeated solicitation, and a model derived from empirical data [6] of LiveJournal community joining. For all our diffusion models, once a person decides to consider a solicitation, she will never consider it again.

VI) Contribution Model $\mathfrak{C}$: To model how much each person contributes to the system, our default model is based on the distribution of computer availability [15], which would be appropriate for systems in which the contribution is related to machine time. In our sensitivity analysis, we also consider a random uniform distribution and a constant uniform distribution.

7.1.3 Parameters

Most of the above models come with a rich set of parameters that can be set to tune the model behavior. Table 1 shows our default values for these parameters (wherever possible taken from the corresponding literature). All our evaluations are performed using these values unless stated otherwise. Our values of β and δ in Pachira are derived in Section 7.2.2, and sensitivity of these values with regard to changes in the other parameters and environmental factors is evaluated in Section 7.2.6.

7.2 Results

In order to reduce variance, all our simulation results were repeated 20 times and the respective average values are plotted. In all cases, the experienced variance is small (within 10%) and error bars are therefore omitted. We call the number of people that end up joining the lottree, after simulating one year of deployment, the *penetration*.

Solicitation Assumption Factor (SAF)	0.5
Payout	1000
Solicitation Time Cost (STC)	30 seconds
Join Time Cost (JTC)	30 minutes
Population Size	10^7
Diffusion Probability p	0.1
Diminishing Cascade Factor q	0.9
Social Network Model: $pInit$, $Range$, $Seed$	Toivonen et al. 0.95, 3, 30 [33]
Prospect Theory: α, γ	0.88, 0.61 [35]
Income Distribution Model	USA 2005 [2]
Contribution Model: Availability parameters	$b = 0.3$; $c = 2.7$; $g = 9.2$; $r = 11$ [15]
Pachira Parameter Settings: β, δ	0.5, 0.08

Table 1: Baseline simulation configuration

7.2.1 Population effects

Before reporting actual results, we first show that the size limitations of our simulation are not an issue. Our simulated population is ten million people, which is significantly smaller than the population that could be reached in the real world. However, Figure 2 (left) shows that the penetration substantially levels out as the population increases. By the time we reach a population of 10^6, there is little additional penetration from increasing population.

7.2.2 Optimal β and δ

Our first order of business is to determine optimal values for the Pachira lottree parameters β and δ. Figure 2 (middle) shows the penetration as a function of these parameters. Only a small range of δ values is shown because higher and lower δ results in significantly reduced penetration. The plots show that, although the optimal choice of β and δ depends upon specific environmental factors, the curves exhibit a regular shape. Based on these and other results, we select default values of $\beta = 0.5$ and $\delta = 0.08$. Although not optimal in every setting, our sensitivity analysis in Section 7.2.6 shows that these settings exhibit good behavior even as we vary the different model parameters.

7.2.3 Deployment tuning

A key question for an executive who wishes to use a lottree for system deployment is how much money to offer as a payout. The answer is a function of the desired penetration and the expected effort of sending solicitations and of joining the system. Figures 3 (left) and (middle) show how the penetration depends on the selected payout, the JTC and STC. As an example, to obtain 100K participants in a system with a STC of 30 seconds and a JTC of 30 minutes, Figure 3 (left) shows that we should offer a payout of 5000 dollars.

Discussion: We can see that expected penetration increases with increasing payout and decreasing JTC; and high values of STC have a negative impact on penetration. However, it is interesting to compare the impact of the STC for different values of JTC. If JTC is 10 minutes, the achieved penetration nearly doubles when STC is reduced from 1 minute to 30 seconds. Conversely, for JTC of 100 minutes, the penetration is virtually equal regardless of whether STC is higher or lower. The reason for this behavior is that in the former case, because JTC is so low, the bottleneck that limits growth are solicitations. In contrast, if JTC is large, people stop becoming new participants because their perceived value predicted by Prospect theory becomes too small faster than solicitations becoming a bottleneck. The plot shows that the equilibrium point in which joining and solicitations start become limiting factors at roughly the same size is reached at about JTC of 30 minutes.

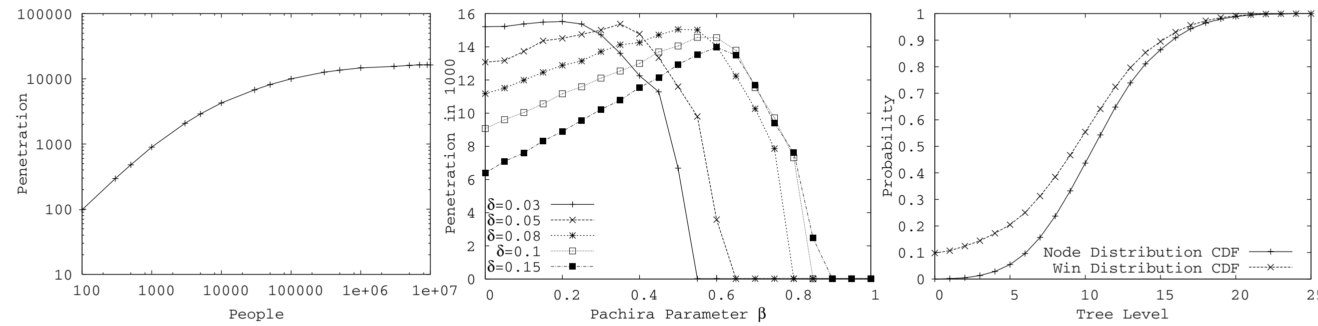

Figure 2: Impact of population size (left). Impact of varying β and δ (middle). Tree-depth distribution of nodes and wins (right).

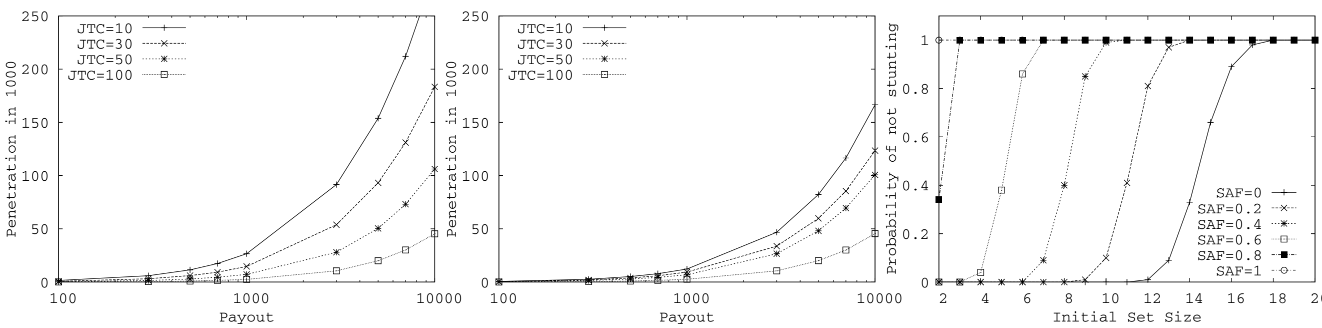

Figure 3: Impact of payout and JTC for STC= 30 seconds (left) and STC= 1 minute (middle). Impact of IS size on stunting (right).

7.2.4 Distribution of nodes and wins

It is insightful to view the tree structure and its induced win-probability distribution resulting from the Pachira lottree. For our default parameters, the lottree grows in average up to depth of 25. Figure 2 (right) shows the distribution of nodes and win probabilities relative to the different levels of the tree (the root level is 0). These CDFs show that Pachira generates a bell-shaped distribution in which the majority of the nodes are contained in the middle levels. Relative to the distribution of nodes, the distribution of win probability is shifted slightly towards the higher levels, reflecting the effect of the solicitation incentive that rewards nodes with many descendents. The figure shows that our default parameters of $\beta = 0.5$ and $\delta = 0.08$ strike a subtle balance between fairness (probability curve follows the node distribution curve) and solicitation incentive (left shift of the probability curve).

7.2.5 WSI vs. SSI: Stunting and initializing

In the sensitivity analysis of Section 7.2.6, we will show that a Pachira lottree's penetration is not significantly affected by varying the solicitation assumption factor SAF, which indicates that the absence of SSI is not a critical weakness for Pachira. However, SAF does have an impact on lottree deployment; specifically, the SSI property can be violated when the lottree is very small, so low values of SAF can lead to stunted deployment that never exceeds a small factor over the initial set size.

Figure 3 (right) shows the probability of stunting with different values of SAF and different sizes of IS. What we see is that the probability of achieving sustained growth becomes unity for a sufficiently large initial set. At an *initSize* of 20, no run was ever stunted. What is particularly interesting to observe is that the critical initial set size required to guarantee sustained growth crucially depends on the given SAF. The higher this value, the higher participants weigh their marginal perceived gain in terms of WSI rather than SSI, the smaller an IS suffices. In the extreme case in which SAF equals one, stunting never occurs.

These observations lead to the conclusion that Pachira's violation of SSI weighs particularly heavy when the number of participants in the tree is small. Empirically, this shows that while Pachira is not guaranteed to satisfy SSI, such violations occur only at the very initial state of the lottree's growth. This shortcoming can therefore be circumvented by starting the lottree-based motivational deployment of a networked system with a sufficiently large IS.

7.2.6 Sensitivity analysis

Through the above experiments, we have derived $\beta = 0.5$ and $\delta = 0.08$ as our default values for the two Pachira parameters. In order to verify these choices, we conducted an extensive sensitivity analysis with regard to all our model parameters and environmental assumptions. We also evaluated these parameters by substituting entire model blocks.

The sensitivity analysis is based on the following methodology. We pick a specific environmental factor (for instance JVT or a parameter from prospect theory) and vary its value. For each sample point, we determine (1) the average penetration P_{our} for this set of parameters when using Pachira with our own choice of β and δ, and (2) the average penetration P_{opt} when using the optimal values of β and δ for this particular point in the parameter space, which we denoted by β' and δ'. We then define the *competitive ratio* P_{our}/P_{opt} as the fraction of penetration achieved by $\beta = 0.5$ and $\delta = 0.08$ compared to the optimum choice of β and δ for that specific setting.

Because finding β' and δ' operation involves a complex search over a two-dimensional parameter space in which each random sample point may experience variance, finding optimal values for β and δ given a set of configuration parameters is a computationally intensive task. For this reason, we have conducted our sensitivity analysis with a reduced population size of 10^6. Resorting to this smaller population is justified by Figure 2 (left). Our implementation of the search procedure itself is based on a hill-climbing algorithm with decreasing step-size.

A few examples of our sensitivity results are shown in Figure 4, and a summary of our main results is presented in Table 2. The

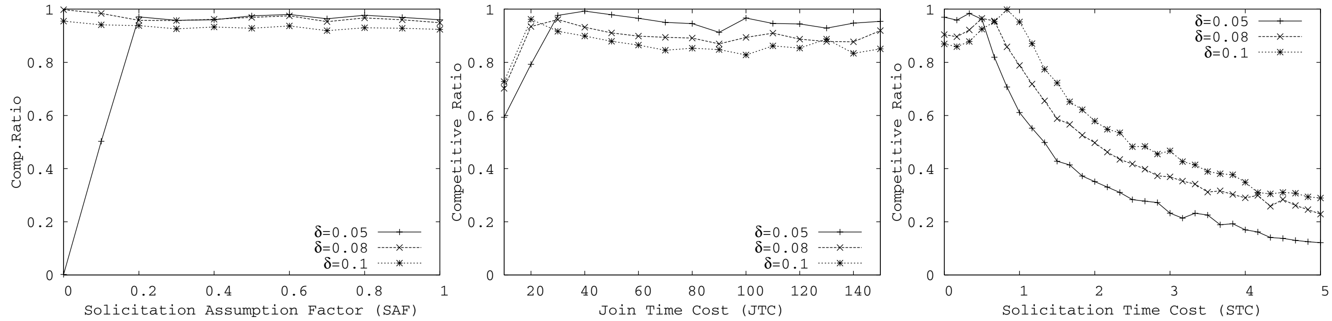

Figure 4: Sensitivity Analysis for SAF, JTC, and STC.

SAF	0.0 – 1.0	0.949
Payout	10^3 – 10^6	0.977
STC	0 – 5 minutes	0.228
JTC	10 – 150 minutes	0.702
Toivonen: CInit	0.0 – 1.0	0.939
Toivonen: CRange	1 – 10	0.967
Contribution Models	Uniform Random	0.972
	Uniform Constant	0.976
Prospect Theory α	0.4 – 1	0.82
Prospect Theory γ	0.4 – 1	0.886
Independent Cascade p	0.05 – 1	0.958
Diminishing Cascade p	0.05 – 1	0.957
Diminishing Cascade q, $p = 0.1$	0.5 – 1	0.803
Diminishing Cascade q, $p = 0.2$	0.5 – 1	0.924
Diffusion Model	LifeJournal [6]	0.993

Table 2: Sensitivity analysis—competitive ratios

table shows the worst competitive ratio achieved by our choices of β and δ across the simulated range. With one exception, our choice of β and δ achieves a competitive ratio of at least 0.7 for all parameter sweeps. The exception is STC as shown in Figure 4 (right); the competitive ratio of our parameters starts dropping significantly as STC increases. In all other cases, our choices were robust to changes in model settings. Interestingly, this holds even as we substituted entire model blocks (social network model, diffusion model, etc...) by other models. For instance, our choices of β and δ achieve a competitive ratio of more than 0.9 when replacing the availability-based contribution model [15] by a uniform contribution model, or when replacing the independent cascade diffusion model with a model that is based on real diffusion data reported for the LiveJournal community in [6].

8. LEGAL ISSUES

This section[2] *applies only to laws of the United States; laws in other countries may differ considerably.*

There are three classes of law that have technical bearing on the lottree mechanism as presented in this paper: *promotion law*, *tax law*, and the *CAN-SPAM Act* of 2003.

Two aspects of promotion law impact lottree deployment. First, depending on the effort required, the installation and running of a distributed-system component may be judged to be "consideration", meaning that it is legally regarded as a transfer of tangible value from the participant to the executive. If so, the executive is obligated to provide an *alternate means of entry* (AMOE) by which a person can become eligible for the payout without participating in the system. This is the reason behind the "no purchase necessary" disclaimers that typically accompany commercial sweepstakes. The potential impact on a lottree deployment is that some small fraction of "participants" may not actually be contributing in any way to the system. Second, promotion law generally disallows variable prize pools, which precludes use of the rescaling technique described in Section 5.3.

The main issue involving tax law is that any payment in excess of the 1099 threshold requires the filing of a 1099-MISC tax form. This threshold is set for each tax year and is 600 dollars for tax year 2007. The impact on lottree deployment is that a payment in excess of 600 dollars may be more burdensome for both the system executive and the lottree winner.

The CAN-SPAM Act of 2003 was designed to legally inhibit companies from sending unwanted commercial email. It also restricts the degree to which a company can encourage others to send commercial email on the company's behalf. This law impacts lottree deployments in two ways: First, the lottree must not encourage email solicitations in preference to other modes of solicitation. Second, the lottree should limit the number of solicitations each participant can issue per day; in our simulations, we limited this number to three.

We believe that the Pachira scheme, which we plan to employ to spur deployment of our own networked system, complies with all the above laws.

9. RELATED WORK

There exists a vast literature on incentive mechanisms and techniques in the networking literature. However, most prior schemes are relevant only to symmetric systems in which every node has a rational interest in participating or contributing. As pointed out in the introduction, such mechanisms are unsuited to be employed as a means for motivational deployment to bootstrap asymmetric systems or symmetric systems that require a sufficiently large network effect to become self-sustained. Much of this work, for instance, is tailored to specific peer-to-peer applications, including file sharing [32], routing [8], content distribution [31], and multicast [27]. There has also been research into application-generic symmetric incentives such as bartering [11], economic systems [36], tit-for-tat [23] as in BitTorrent, or robustness in BAR models [22]. In the context of asymmetric systems, CompuP2P [18] is a peer-to-peer system that constructs decentralized markets for buying and selling computing resources. CompuP2P assumes the availability of a electronic payment mechanism to scalably and securely transfer funds from the system's beneficiaries to its contributors, in exchange for use of the contributors' resources. Kamvar *et al*. [19] consider a pay-per-transaction file-sharing system, wherein peers are in competition for the opportunity to profit by providing requested file content. They show that such competition can lead to non-cooperation, similar to a lottree without the WSI property.

[2] Although we address only those legal issues with direct technical impact, it exceedingly important to respect all applicable laws when deploying a lottree system. It is a misdemeanor to run any system in which some form of value is distributed randomly, if not constructed in accordance with applicable laws.

10. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the question of how to motivate people to join or contribute to a networked system that does not (or not yet) offer them inherent participation benefit. In answer, we proposed a lottery tree, a mechanism that probabilistically rewards each participant in a manner dependant on its contribution as well as on the contributions of others whose participation it has solicited.

Lottrees are most effective at spurring deployment when systems are small or medium-sized, which are the scales at which motivational deployment is most challenging. As the system scale increases, the lottree's effectiveness begins to wane, just as the self-sustaining aspects of the networked system can be expected to become active.

We formally defined seven desirable properties for lottrees and constructed the Pachira lottree scheme, which simultaneously satisfies a maximal satisfiable subset of these properties. We further showed relatively straightforward work-arounds for the two properties that Pachira does not satisfy.

We then conducted extensive simulations, with which we derived good choices for the Pachira lottree's parameters, determined an appropriate payout amount based on target deployment scale and expected participation effort, and determined the required count of initial participants to preclude stunted deployment. We also performed a wide range of simulation experiments to analyze the sensitivity of our evaluation to our modeling assumptions and hidden parameters.

We conclude that Pachirais a practically ideal candidate for deploying real networked systems, and we plan to employ this scheme as part of an ongoing distributed-systems project requiring contributions of CPU and bandwidth from a large number of PC users.

A looming open problem is *auditing*. The lottree mechanism is inherently based on the assumption that each participant's contribution can be reliably and securely measured and reported to the executive entity. Depending on the properties of the system in question, this may be anywhere from thoroughly straightforward to exceedingly challenging.

It may be interesting to consider generalized versions of lottree systems, such as those not constrained to a tree structure. This could be relevant, for example, in cases in which a potential participant is concurrently solicited by more than one active member of the system.

11. REFERENCES

[1] myspace.com: a place for friends. http://www.myspace.com/.

[2] US Census 2005, Income data, 2005.

[3] WikiPedia: The Free Encyclopedia. http://www.wikipedia.org/.

[4] D. P. Anderson. BOINC: A System for Public-Resource Computing and Storage. In *Proc. 5th IEEE/ACM GRID*, Nov 2004.

[5] S. Androutsellis-Theotokis and D. Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, 36(4):335–371, 2004.

[6] L. Backstrom, D. Huttenlocher, J. Kleiberg, and X. Lan. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In *Proc. 12th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[7] Berkeley Open Infrastructure for Network Computing. BOINC Combined Statistics. http://boinc.netsoft-online.com/, 2006.

[8] A. Blanc, Y.-K. Liu, and A. Vahdat. Designing Incentives for Peer-to-Peer Routing. In *2nd NetEcon*, 2004.

[9] M. Brossi. *Multilevel marketing: A legal primer : a handbook for executives, entrepreneurs, managers and distributors*. Direct Selling Association, 1991.

[10] C. Christensen, T. Aina, and D. Stainforth. The challenge of volunteer computing with lengthy climate model simulations. In *E-SCIENCE*, pages 8–15. IEEE Computer Society, 2005.

[11] B. Chun, Y. Fu, and A. Vahdat. Bootstrapping a Distributed Computational Economy with Peer-to-Peer Bartering. In *1st NetEcon*, 2003.

[12] B. Cohen. Incentives Build Robustness in BitTorrent. In *1st NetEcon*, 2003.

[13] P. Dewan and P. Dasgupta. PRIDE: Peer-to-Peer Reputation Infrastructure for Decentralized Environments. In *WWW 2004*, 2004.

[14] J. R. Douceur. The Sybil Attack. In *1st IPTPS*, 2002.

[15] J. R. Douceur. Is Remote Host Availability Governed by a Universal Law? *SIGMETRICS Performance Evaluation Review*, 31(3), 2003.

[16] J. Goldenberg, B. Libai, and E. Muller. Using Complex System Analysis to Advance Marketing Theory Development. *Academy of Marketing Science Review*, 2001.

[17] L. Guernsey. Project Uses Simulations to Research Flu Vaccines. *The New York Times*, December 2000.

[18] R. Gupta and A. K. Somani. CompuP2P: An Architecture for Sharing of Compute Power In Peer-to-Peer Networks With Selfish Nodes. In *2nd NetEcon*, 2004.

[19] S. Kamvar, B. Yang, and H. Garcia-Molina. Addressing the Non-Cooperation Problem in Competitive P2P Networks. In *1st NetEcon*, 2003.

[20] M. Katz and C. Shapiro. Systems Competition and Network Effects. *Journal of Economic Perspectives*, 8(2):93–115, 1994.

[21] S. M. Larson, C. D. Snow, M. R. Shirts, and V. S. Pande. Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology. *Computational Genomics*, 2002.

[22] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. BAR Gossip. In *Proceedings of the 7th Symposium on Operating System Design and Implementation (OSDI)*, 2006.

[23] Q. Lian, Y. Peng, M. Yang, Z. Zhang, Y. Dai, and X. Li. Robust Incentives via Multi-level Tit-for-tat. In *5th IPTPS*, 2006.

[24] Magix. freeDB.org. http://www.freedb.org/.

[25] T. Mengotti. *GPU, a Framework for Distributed Computing over Gnutella*. ETH Zurich, 2004. CS Masters Thesis.

[26] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45, 2003.

[27] T.-W. Ngan, D. S. Wallach, and P. Druschel. Incentives-Compatible Peer-to-Peer Multicast. In *2nd NetEcon*, 2004.

[28] V. Pai and A. E. Mohr. Improving Robustness of Peer-to-Peer Streaming with Incentives. In *1st NetEcon*, 2006.

[29] P. Rogers. The Cognitive Psychology of Lottery Gambling: A Theoretical Review. *Journal of Gambling Studies*, 14(2):111–134, 1998.

[30] M. Shirts and V. S. Pande. Screensavers of the World, Unite! *Science*, 290(5498):1903–1904, December 2000.

[31] M. Sirivianos, X. Yang, and S. Jarecki. Dandelion: Secure Cooperative Content Distribution with Robust Incentives. In *1st NetEcon*, 2006.

[32] K. Tamilmani, V. Pai, and A. E. Mohr. SWIFT: A System With Incentives For Trading. In *2nd NetEcon*, 2004.

[33] R. Toivonen, J.-P. Onnela, J. Saramki, J. Hyvnen, and K. Kaski. A Model for Social Networks. *Physica A*, 371(2), 2006.

[34] A. Tversky and D. Kahneman. Prospect Theory: An Analysis of Decision Under Risk. *Econometrica*, 47(2), 1979.

[35] A. Tversky and D. Kahneman. Advances in Prospect Theory: Cumulative Representation of Uncertainty. *Journal of Risk and Uncertainty*, 5, 1992.

[36] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. KARMA: A Secure Economic Framework for P2P Resource Sharing. In *1st NetEcon*, 2003.