Ensuring Prolonged Participation and Deterring Cheating Behaviors in a Collective

Sachin Goyal and John Carter

UUCS-08-010

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

September 25, 2008

Abstract

We are building a system that harnesses the idle resources (cpu, storage, and bandwidth) of nodes (e.g., home desktops) distributed across the Internet to build useful distributed services like content distribution or remote backup. Users are compensated in return for contributing their nodes' idle resources to the system. Collective managers bundle and manage the contributed resources and resell them to end customers.

For such a collective system to work, the system must discourage cheating (e.g., cheating users who lie about how many resources they have provided) and encourage nodes to stay in the collective for extended periods of time. To achieve these goals, we have designed an incentive system based on game theory and the economic theory behind law enforcement that motivates just these behaviors. In this paper we describe our incentive system and analyze its economic underpinnings to gain insight into how different players in the system will behave. We demonstrate how our incentive system motivates nodes to stay in the system for prolonged duration and deters cheating. For a typical system configuration, we show that even if we can only detect cheaters 4% of the time we can create sufficient economic deterrents to demotivate cheating.

Ensuring Prolonged Participation and Deterring Cheating Behaviors in a Collective

Sachin Goyal and John Carter
School of Computing, University of Utah
{*sgoyal, retrac*}@*cs.utah.edu*

Abstract

We are building a system that harnesses the idle resources (cpu, storage, and bandwidth) of nodes (e.g., home desktops) distributed across the Internet to build useful distributed services like content distribution or remote backup. Users are compensated in return for contributing their nodes' idle resources to the system. Collective managers bundle and manage the contributed resources and resell them to end customers.

For such a collective system to work, the system must discourage cheating (e.g., cheating users who lie about how many resources they have provided) and encourage nodes to stay in the collective for extended periods of time. To achieve these goals, we have designed an incentive system based on game theory and the economic theory behind law enforcement that motivates just these behaviors. In this paper we describe our incentive system and analyze its economic underpinnings to gain insight into how different players in the system will behave. We demonstrate how our incentive system motivates nodes to stay in the system for prolonged duration and deters cheating. For a typical system configuration, we show that even if we can only detect cheaters 4% of the time we can create sufficient economic deterrents to demotivate cheating.

1 Introduction

Modern computers are becoming progressively more powerful with ever-improving processing, storage, and networking capabilities. Typical desktop systems have more computing/communication resources than most users need and are underutilized most of the time. These underutilized resources provide an interesting platform for building distributed applications and services. Two important obstacles to successfully harnessing these idle resources are ensuring prolonged participation of the nodes in the system and deterring selfish behaviors.

We are building a system to harness idle resources as managed *collectives*. Rather than employing purely P2P mechanisms, a collective uses *collective managers* that manage the available resources of large pools of untrusted, selfish, and unreliable *participating nodes*. Participating nodes contact collective managers to make their resources available, in return for which they expect to receive compensation. Each participating node runs a virtual machine (VM) image provided by the collective manager(CM). CMs remotely control these VMs and use these processing, storage, and network resources to build distributed services needed by *customers*. Collectives are similar to computational grids [8] in that there is a degree of centralized management and control, but a key difference is that the nodes comprising a collective do not belong to a single administrative entity, are inherently untrusted, and join and leave the collective (churn) more rapidly than typical grid nodes. Figure 1 illustrates a possible use of a collective to implement a content distribution service that distributes large content files (e.g., movies, music, or software updates) to thousands of clients in a cost-effective way.

Since individual participants in a collective are selfish (rational) nodes, it is important to mitigate the negative effects of selfish behavior. Selfish nodes can resort to cheating for earning more than their fair share

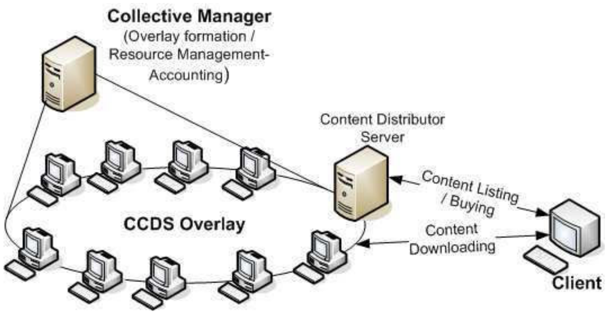

Figure 1: Collective Content Distribution Service

of compensation. Cheating behavior has been observed extensively in distributed systems, e.g., free riding in Gnutella [1] and software modifications to get more credit than earned in SETI@home [12].

Another challenge faced by collectives is ensuring that nodes stay in the system for prolonged durations, which improves the system stability and allows tasks to be scheduled more efficiently. For example, a collective manager can use historical data of node availability to make informed decisions regarding issues such how many replicas of a particular datum to maintain. A lower degree of replication needed to ensure a given level of availability using nodes that tend to persist in the collective for a long time or quickly rejoin when they temporarily leave.

To address these challenges, we have designed an incentive system based on game theory and the economic theory behind law enforcement that motivates just these behaviors. In 1968, Becker [3] presented an economic model of criminal behavior where actors compare the expected costs and expected benefits of offending, and only commit crimes when the expected gains exceed the expected costs. Since then there has been significant research extending the work of Becker – Polinsky et. al [16] provides a comprehensive overview of the research dealing with deterrents in law enforcement.

In Section 2 we present a brief overview of our system design and its novel incentive system that motivates participating nodes to remain in the system for long, predictable durations. Our incentive model employs a currency-based system that rewards work performed, as well as the consistency of the work. Further, it is a well known phenomenon in game theory that repeated interactions give rise to incentives that differ fundamentally from isolated interactions [15]. Thus, the collective manager employs offline analysis of data provided by participating nodes, partners, clients, and collective managers to determine future pay rates for each node. Consistently desired behavior leads to increased rewards, e.g., the pay rate of nodes increases in response to predictable long term availability. Undesirable behavior results in decreased rewards, e.g., the pay rate of nodes decreases in response to being caught lying about work done in an attempt to receive undeserved compensation.

In Section 3, we analyze the impact of our incentive model from an economic standpoint to derive key properties of our incentive system. We examine the impact of decisions made by dishonest nodes and analyze the gain vs loss possibilities for participating nodes as we vary the likelihood of bad actors being caught. We show that while we cannot prevent users from cheating, our mechanisms mitigate cheating behavior by making it economically unattractive. We show that a small probability of catching cheaters (under 4%) is sufficient for creating a successful deterrence against cheating. We further show that our incentive system can be used successfully to motivate nodes to remain in the system for prolonged durations.

2 Collective System

Design Overview

In our model, a collective supports the development of distributed services built using the idle resources of untrusted end nodes. There are two main types of nodes in a collective: *participating nodes* (PNs) and *collective managers* (CMs).

Participating nodes are end nodes that have idle compute, storage, or network resources. They are typically connected to the Internet though some sort of broadband service. PNs have different compute/communication capabilities, go up and down in unpredictable ways, and are inherently untrusted.

Collective managers are service providers to whom individual nodes provide their idle resources. A CM uses these resources to provide a set of meaningful services to clients, in return for which it compensates PNs. Multiple competing CMs can co-exist, each providing different services and/or pricing models.

A typical distributed service built on a collective consists of components that run colocated on the CM (called *service managers*) and other components that run on the PNs. A service manager is responsible for converting client service requirements into small components and distributing these components to a set of PNs. Typically each service component will be replicated to provide availability and scalability.

Figure 1 shows how we might provide a collective content distribution service (CCDS). A content distributor contracts with a CM to purchase access to resources managed by the CM. The content distributor interfaces with a service manager co-located on the CM node, which divides the content into multiple (probably encrypted) chunks and caches them across multiple PNs. Clients contact the content distributor's server to purchase content and are given the location and decryption keys of encrypted chunks of content. Clients then contact individual PNs to download the purchased content. In this scenario, the content distributor is responsible for advertising and selling content, but the actual content delivery is handled by nodes in the collective. In particular, it does not need to maintain its own content distribution network, ala Akamai [2], or data centers with large bandwidth pipes, but rather exploits idle bandwidth of hundreds or thousands of end nodes, ala BitTorrent [7]. Unlike BitTorrent, the content distributor can be receive guarantees regarding availability, average download latency, and other quality of service issues that are critical when building a successful Internet business.

Incentive Model

In a collective system, a PN's compensation is based on how much its resources contribute to the success of services running on the collective. A CM shares its profits with PNs in proportion to their contribution towards different services. For example, in the CCDS example, PNs will receive a fraction of the money paid by the content distributor roughly proportional to the fraction of the total content that they deliver. The basic unit of compensation is a CM-specific credit that acts as a kind of currency. Users can convert credits to cash or use them to buy services from the CM or associated partners.

For the incentive system to work, the CM needs an accurate accounting of each PN's contribution. The CM cannot simply trust the contribution reported by each node, since dishonest nodes can exaggerate their contributions. In this section we discuss how we discourage dishonest behavior economically.

Contribution accounting is mostly done at the service level and depends on the design of the service involved. The basic idea is to collect information from multiple sources (e.g., PNs, partners, clients, and the CM) and do offline data analysis to decide the individual node's contribution. We employ the following mechanisms:

Credits Earned $\propto$ **Work Performed**: The work performed to support a service invocation, e.g., downloading a movie, should be credited to the appropriate PNs. Each PN sends a detailed daily report of its activities to the CM. In the absence of dishonest PNs, each service activity can be credited to unique contributing PNs. If nodes are dishonest, more than one node will request credit for the same work. To resolve conflicts, the accounting system needs additional information.

Accountability: Each PN and each client is identified by a unique public/private key pair. The CM acts as the root of the public key infrastructure (PKI) employed by its collective. Each PN and client is issued a certificate signed by the CM that associates the public key of the PN or client with their unique IDs. These keys and certificates are used to create secure communication channels and to digitally sign the reports sent to the CM.

Offline Cheater Detection: To identify dishonest nodes, the system collects data from PNs, CM scheduling records, service scheduling records, partners' sales records, and even completion reports by client applications (if available). This data is used to resolve conflicts by comparing what work nodes claim they did against what other entities claim was done. Conflict resolution is done offline periodically (e.g., daily). With multiple information sources, it is possible to detect dishonest/cheating behaviors by PNs. However, we do not assume that CMs will be able to detect all instances of cheating behaviors – in Section 3.4 we show that our incentive model works even when we can only detect 4%-5% of cheating behaviors.

Variable Pay Rates (Raises and Cuts): To provide an incentive for nodes to provide stable resource levels and to penalize node churn, the amount of credits received by a node in return for work depends on the node's long term *consistency*. A node that remains in the CM's pool for long periods of time and that provides continuous predictable performance receives more credit for a unit of work than a node that flits in and out of the CM's pool.

Credit-per-unit-work (pay) rates are divided into levels. PNs enter the system at the lowest pay rate; a node's pay rate increases as it demonstrates stable consistent contributions to the collective. The number of levels and the behavior required to get a "pay raise" are configurable parameters for any given service.

To discourage dishonest behavior, the system can apply a pay cut when it identifies a node mis-reporting the amount of work it performs. The size of the pay cut can be configured on a per-service basis. Dishonest behavior in one service leads to pay cuts in other services run on that node. As an alternative, we could ban PNs from the system when they are caught cheating, but doing so eliminates nodes who might "learn their lesson" after finding that cheating does not pay in the long run. If a node continues to cheat, its pay rate becomes negative (i.e., it accumulates debt that must be worked off before being paid), which has the same effect as simply banning them.

Other factors can be applied to determine a particular node's pay rate. For example, nodes that are particularly important to a given service due to their location or unique resources (e.g., a fat network pipe or extremely high availability) may receive a bonus pay rate to encourage them to remain part of the CM's pool.

3 Economic Analysis

This section explores the design of our incentive system from an economic perspective. In particular, we use game theory and probabilistic analysis to gain better insight into the implications of our design choices.

Our economic analysis focuses on the two main entities in our system, collective managers and participating nodes. Participating nodes are assumed to be self interested, rational parties, which from a game theory standpoint means that they act in ways that maximize their long term financial gain even if this involves *cheating*. A collective manager is a trusted party that manages the resources of participating nodes to support commercial services. Its goal is to build a successful business providing services to external customers using its PNs' resources.

In game theory, systems are modeled as *games* played between *players*. Players are faced with a series of options from which they must choose. The outcome of each game (choice) depends on the player's choice and the choice(s) made by their opponent(s). The most famous example of game theory is the Prisoner's

Dilemma [11], where two prisoners who are both accused of a crime are separated and individually given the option of either "cooperating" (staying silent) or "defecting" (confessing to the crime and testifying against the other prisoner). If both prisoners stay silent, they receive a 6-month sentence. If both prisoners defect, they both receive a 5-year sentence. If one prisoner cooperates and the other defects, the one who defects is set free, while the one who cooperates is given 10-year sentence. In a variant of the game where the players play the game repeatedly, researchers have found that they tend to learn to cooperate with one another and thus receive light sentences [11]. We exploit this phenomenon in our incentive model.

We model the interaction between participating nodes and the collective manager using a basic game theoretic utility model. At any given time, we present PNs with two orthogonal choices: (i) should they remain in the collective or not and (ii) should they report the correct amount of work for the last reporting period or attempt to claim they did more work than they did to receive a higher (undeserved) payment from the CM. In this game at any time slot s, a rational PN node can either choose to share or not share its resources based on the expected reward of each choice. We can represent the choices available to PNs and the collective manager using simple tables like Tables 1-4. Each column represents the options available to the collective manager and each row represents the options available to a PN. Entries in table take the form a/b where a is the payoff (reward) for the row player (i.e., PN) and b is the payoff for the column player (CM). In a typical game theory situation, the two players make simultaneous decisions, but in our scenario PNs make their decision (share, no share) and then CMs make their decision (reward or not reward the PN).

The "games" played as a part of collective are non zero-sum games, meaning that one player's gain is not necessarily another player's loss (and vice versa). PNs are not assumed to be altruistic, but rather we want to derive an incentive model where it is in each node's rational self-interest to cooperate. In other words, it is our goal to design rules for the "game" such that rational actors will find cheating economically unattractive. In the remainder of this section, we consider different scenarios and determine whether the outcome realized achieves this goal.

3.1 Perfect Monitoring

We start by assuming a perfect monitoring scenario, i.e., the collective manager has perfect information about the contributions made by PNs it manages. In this case a PN cannot successfully lie to a CM about how much work it performs, because if they lie, they are guaranteed to be caught.

Table 1 shows the payoff structure for this scenario. A dash means that a particular case is not possible in this scenario, e.g., it is not possible for a PN to choose "Not Share" and have the CM choose to "Reward" it. Assuming a CM shares its income 50%-50% between itself and the PN concerned, we get the value of G_S/G_S for the PN share case. This means that if a PN share its resources, noth it and the CM receive G_S benefit. Here G_S is a positive number, which denotes the gain (payoff) received for sharing.

	Reward	No Reward
Share	G_S/G_S	-
Not Share	-	0/0

Table 1: Payoffs for Perfect Monitoring Case

If we apply standard game theory analysis to this utility table, both (share/reward) and (no-share/no-reward) are *Nash equilibrium* [11]. Informally, a strategy is a Nash equilibrium if no player can do better by unilaterally changing his or her strategy. Even though (share/reward) is pareto optimal, meaning that it leads to both players receiving their highest reward, both cases are equally possible from a Nash equilibrium point of view.

This analysis assumes that both players choose their actions independently, which as we mentioned above is not the case in our design. In our case, a CM makes its choice only after analyzing the action of the PN concerned, which is why the two dashed states are not possible. Hence, a PN knows that the CM will always choose *reward* in response to *share*, which tilts the equilibrium balance towards *(share/reward)* instead of *(no-share/no-reward)*. This behavior of the collective manager greatly simplifies our analysis of the various scenarios discussed throughout the paper and leads to pareto optimal choices for rational PNs.

Cost of Sharing: Table 1 does not model the fact that there is a cost associated with performing a job (e.g., power charges). Let c be the cost of performing a job, which typically will be small since we are exploiting idle resources, but positive. Table 2 shows a modified reward structure that accounts for this cost.

	Reward	No Reward
Share	$(G_S - c)/G_S$	-
Not Share	-	0/0

Table 2: Payoffs for Perfect Monitoring Case with cost of sharing included

3.2 Imperfect Monitoring

The previous analysis assumes that the CM has perfect knowledge regarding whether a PN is accurately reporting how much work it performs. Table 3 shows a payoff table if we assume that a CM can only detect PN lies with some non-zero probability. Here c continues to represent the cost for a PN to perform a unit of work. Rational PNs now have an additional choice available to them; they can chose to *lie* to the CM, claiming to do work that they have not done. G_{cheat} is the expected reward that a PN will receive if it lies, and L is the loss incurred by the CM due to incorrect awarding of credits. If the system cannot detect lies, then G_{cheat} is equal to G_S, in which case a rational PN will always lie, since this lets it receive a reward without doing any work. Thus, if CMs cannot detect lying PNs, the system will destabilize since cheating PNs will always claim to do work, but not do it.

	Reward	No Reward
Share	$(G_S - c)/G_S$	-
Cheat	$G_{cheat}/-L$	-
Not Share	-	0/0

Table 3: Payoffs for Imperfect Monitoring Case

Our collective service is designed to make it nearly impossible for PNs to successful lie about their contributions. However, it is impractical to track enough information to catch all instances of a PN lying. If we assume that only a fraction of all lies will be detected, we can analyze the impact of undetected lies to determine what probability of lie detection is necessary to motivate selfish PNs to report the truth. Assume that the probability of detecting a lie (offense) is p_o. In that case, the *expected* payoff for lying (G_{cheat}) is:

$$G_{cheat} = (1 - p_o) * G_S$$

We can create a deterrent that punishes PNs when they are caught cheating, i.e., when they provide incorrect accounting information. If F is the amount we penalize PNs when we catch them lying, the *expected* payoff for lying (G_{cheat}) becomes:

$$G_{cheat} = (1 - p_o) * G_S + p_o * -F$$

We can represent F as a certain fraction of G_S, i.e., a PN is penalized a fraction (defined as b) of pay for each unit of work it falsely claims to have done. Adding this penalty results in an expected reward for lying

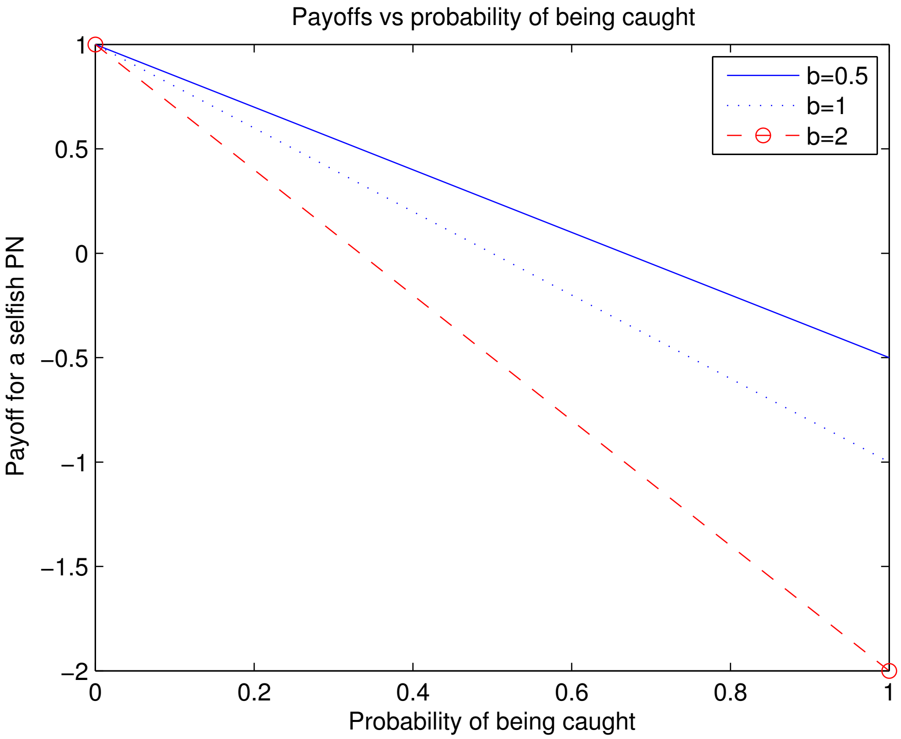

Figure 2: Expected Pay vs Probability of Being Caught

(G_{cheat}) as:

$$\begin{aligned} F &= b * G_S \text{ where } b > 0 \\ G_{cheat} &= (1 - p_o) * G_S - p_o * b * G_S \\ G_{cheat} &= G_S * (1 - p_o - p_o * b) \end{aligned}$$

This results in the payoff table shown in Table 4:

	Reward	No Reward
Share	$(G_S - c)/G_S$	-
Lie	$G_S * (1 - p_o - p_o * b)/ - L$	-

Table 4: Payoffs for Imperfect Monitoring with Penalties

Figure 3.2 plots possible payoff for a single unit of reward ($G_S = 1$) as a function of p_o, the probability of being caught. Different curves in the graph represent the payoff for different values of b, i.e., different sized penalties relative to the standard reward. We observe that the potential payoff of lying drops below zero when the probability of being caught crosses a threshold that depends on b. Specifically, we can derive $G_{cheat} < 0$ as follows:

$$G_S * (1 - p_o - p_o * b) < 0 \implies p_o > \frac{1}{1 + b}$$

So for $b = 1$, a probability of 0.5 or more is required to make fake sharing economically uninteresting to a user. A collective manager can effectively use different values of b to create different degrees of deterrence.

3.3 Variable Pay Rates

In the previous analysis, we considered only single-round games. However, in our system, PNs typical participate in a series of games, which lets us employ the game theory of repeated interactions [11] to analyze the impact of repeated interactions on the behavior of PNs.

A simple solution treats repeated interactions as independent, using the rules presented in earlier sections. In this case, we can use the sum of the individual round payoffs to understand the dynamics of repeated interactions. However, this approach does not exploit our ability to employ a variable pay rate mechanism that responds to observed PN behavior to motivate rational PNs to cooperate. We use pay variability to acheieve two types of positive behaviors from PNs: (i) to encourage nodes to remain in the collective for extended, predictable periods and (ii) to punish cheaters.

To address our first goal, that of encouraging nodes to remain in the collective for extended periods, the amount of payment that a node receives in return for work is varied depending on its long term "consistency". A node that remains in the CM's pool for long periods of time and that provides continuous predictable performance receives more credit for a unit of work than a node that flits in and out of the CM's pool.

In our design, pay rates(R) are divided into l levels, $(R_1, R_2, ...R_l)$, whereby each pay rate is a fixed constant above/below the level below/above it, as follows:

$$R_n = R_1 + I * (n-1) R_n <= R_l \tag{1}$$

PNs enter the system at the lowest pay rate (R_1); a node's pay rate increases as it demonstrates stable consistent contributions to the collective. If a node contributes successfully to collective for T_{raise} consecutive time periods, its pay rate is increased. Periods during which no work is scheduled on a node are not counted for this calculation. The number of levels (l), initial pay rate (R_1), pay rate increment (I), and effort needed to warrant a raise (T_{raise}) are configurable parameters for a given service, and are dependent on the profit margins of the service.

To discourage cheating, the system can apply a pay cut when it identifies a node mis-reporting the amount of work it performs. When such an offense is detected, the PN's pay rate is reduced by the amount of pay increases that would normally accrue for T_{cut} steps (periods) of useful work. Typically T_{cut} is a multiple of T_{raise} (i.e., $T_{cut} = o * T_{raise}$ where $o \geq 1$), so pay is dropped by some configurable number of pay levels. The size of the pay cut (T_{cut}) can be configured on a per-service basis, depending upon the criticality of the offense committed.

We can represent a PN's pay rate at any time t as $R(t)$:

$$R(t) = R_1 + I * \frac{t1}{T_{raise}} - N_{detected} * I * \frac{T_{cut}}{T_{raise}}$$

Here $t1$ represents the number of timeslots where some useful work was performed or claimed to have been performed and the lie went undetected. After time $t1$, a node will receive $I * t1/T_{raise}$ pay increases. $N_{detected}$ represents the number of detected offenses; each such offense leads to a decrease in pay rate equivalent to T_{cut} steps.

3.4 Evaluating the Incentive Model

Let us use this model to analyze the accumulated payoffs for different node profiles to understand how our mechanisms affect node behavior.

3.4.1 Short Lived vs Long Lived Nodes

To analyze the difference between short-lived and long-lived players, we plot the average pay rate received by different honest nodes of similar capabilities with different active life times in the system. We assume

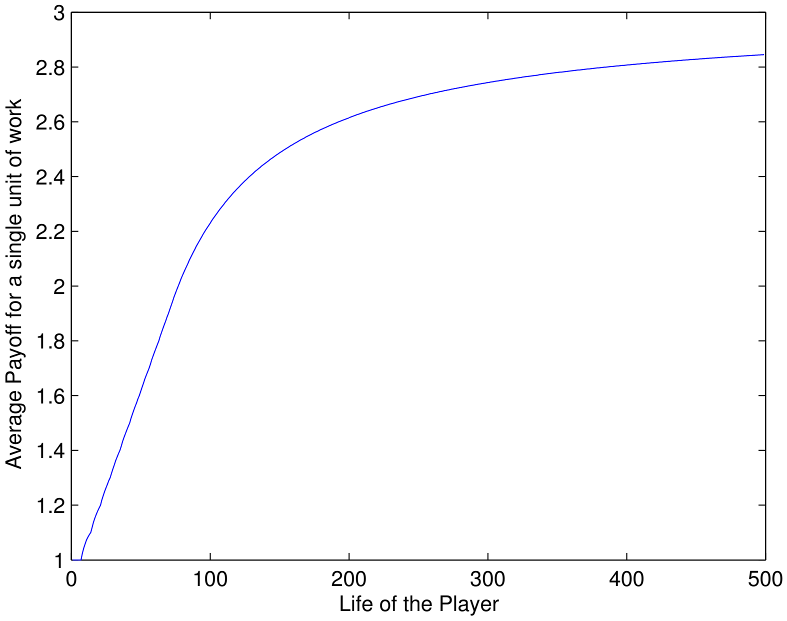

Figure 3: Short Lived vs Long Lived Player

$R_1 = 1$, a sample increment of 0.2 with 10 levels and $T_{raise} = 7$ (e.g., 7 days). Figure 3 plots the average payrate vs the life of a player in the system. This graph clearly shows that patient long-lived players gain clear advantage over short-lived players.

3.4.2 Deterring Cheating Behavior

A rational node will cheat only if the gain from cheating is more than that of honest behavior. Earlier we discussed the expected gain for a single unit of the work. Here we discuss the expected gain for a series of interactions.

A PN that performs work on behalf of a collective can complete only a limited amount of work per time unit given its available resources. In comparison, a cheating PN can fake the completion of an almost unbounded amount of work, irrespective of its resource capabilities. In this section we analyze the expected accumulated gain of a node over a period of time of n time periods, e.g., n days (*we use summation of gain over t from 1 to n to show this*). We consider two scenarios. In the first scenario nodes behave honestly, while in the second scenario nodes claim to complete more work than they really performed (i.e., they cheat).

Let G_{honest} be the expected gain of behaving honestly and G_{cheat} be the expected gain of cheating. If G_{cheat} is more than G_{honest}, then a rational node will always take the cheating route to maximize its gain. We can represent the difference between G_{cheat} and G_{honest} by D:

$$D = G_{cheat} - G_{honest}$$

To remain effective in the face of cheating nodes,D should be less than zero in our system.

We can divide a node's offenses[1] (lies about work done) into two categories, detected offenses and undetected offenses. As explained in previous sections, a detected offense not only leads to a fine but also

[1] We use the term *offense* to denote instances when a node attempts to cheat the system. This choice of terms is motivated by the fact that the following analysis is derived from the game theory associated with criminal law, where offenses refer to crimes [3, 16].

impacts a node's pay rate. Here we analyze the accumulated reward of a cheating node over a period of time to understand the long term impact of cheating.

p_o	Probability of detecting offenses
$R(t)$	Pay Rate at time period t
b	fine ratio, $Fine = b * R(t)$
N_{off}	Number of offenses per time period
N_{actual}	Number of work units that can be completed by an honest node per time period
T_{raise}	Time periods required for a pay raise
T_{cut}	Time period equivalent to a pay rate cut for an offense
o	Ratio of pay cut rate to pay raise rate ($T_{cut} = o * T_{raise}$)
I	Pay raise increment
N_{total}	Total number of offenses committed = $\sum_{t=1}^{n} N_{\text{off}}(t)$
l	Number of Levels (max pay rate = R_l)

Table 5: Glossary of Mathematical Symbols Used

We first consider the case of perfect monitoring where every offense is successfully detected by the CM. Since every offense is detected, a cheater will suffer a penalty for every offense.

$$\begin{aligned} G_{honest} &= \sum_{t=1}^{n} N_{actual} * R(t) \\ G_{cheat} &= \sum_{t=1}^{n} N_{actual} * R(t) - N_{off} * F \end{aligned}$$

Here N_{actual} represents the number of units of work per unit of time that the node can perform given its available resources, and N_{off} represents the number of units of work faked by a cheating node.

If the cheating node commits one (detected) offense in every time slot, it will always be paid at or below the base pay rate, R_1. Effectively,

$$G_{cheat} \leq \sum_{t=1}^{n} N_{actual} * R_1 - N_{off} * F,$$

where F is the fine levied by a CM upon detecting an offense. In this case, $G_{cheat} > G_{honest}$, so cheating is not economically attractive. Even when nodes only cheat once in a while, the fine and lower pay rate lead to less net income than honest nodes, which is unsurprising given the assumption of perfect monitoring.

In case of *imperfect monitoring*, the system does not detect all offenses. Let p_o be the probability that a offense is successfully detected by the CM. In this case, the accumulated gain over a period of time depends upon the distribution over time of offenses performed by the node.

We first consider a case where a node performs N_{off} offenses during every time period (e.g., every day). Given N_{off} offenses in a time period, each having a probability of detection of p_o, we can represent the probability of all offenses going undetected by p_{ndo}. p_{ndo} is the cumulative probability that none of N_{off} offenses is detected, which is $(1 - p_o)^{N_{off}}$. Given p_{ndo}, we can estimate the pay rate at any time interval using the following equation:

$$\begin{aligned} p_{ndo} &= (1 - p_o)^{N_{off}} \\ R(t) &= R(t-1) - p_o * N_{off} * o * I + p_{ndo} * \frac{I}{T_{raise}} \end{aligned}$$

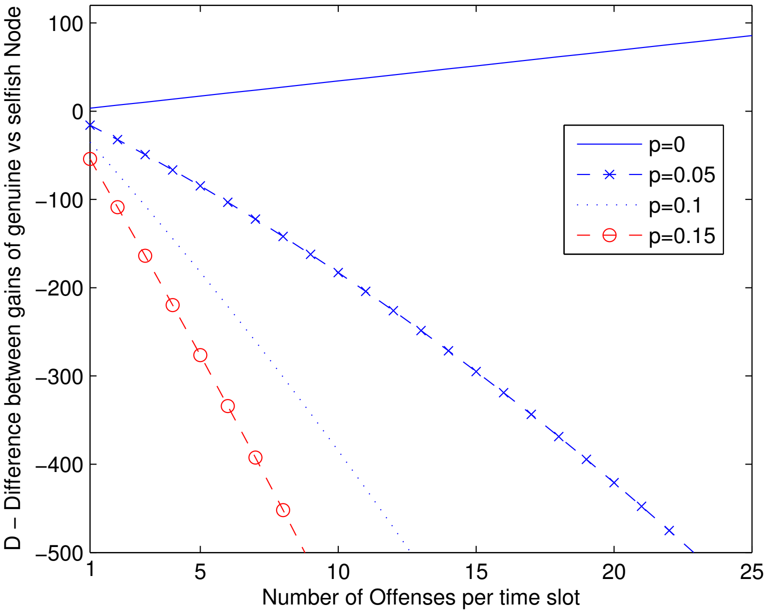

Figure 4: D for Different Number of Offenses

Here $p_o * N_{off}$ represents the expected number of detected offenses. Each detected offense leads to pay rate cut equivalent to $o * I$. Note that $R(t)$ is capped at R_l.

We can represent the accumulated gain of a cheating node, G_{cheat}, as follows:

$$\begin{aligned} SU &= (1 - p_o) * N_{off} + N_{actual} \\ G_{cheat} &= \sum_{t=1}^{n} SU * R(t) - p_o * N_{off} * F \end{aligned}$$

Here SU represents the number of units of work successfully billed by a PN, which includes both real work and undetected falsely claimed work. F represents the fine for a detected offense, which we represent as a multiple of the equivalent reward for performing a unit of work, i.e., $F = b * R(t)$ where $b > 0$

To visualize the implication of these equations, let us consider the case of accumulated gain over a period of 25 days, where each time slot is one day. We use $R_1 = 0.1$ (initial pay rate), a sample increment of 0.02 (pay increase per day of sustained honest operation) with 10 levels, $T_{raise} = 7$, $T_{cut} = 7$, $N_{actual} = 50$, and $F = 1 * R(t)$. In Figure 4 we plot D, the difference between the accumulated gain of cheating and genuine node as we vary N_{off} from 1 to 25.

The results make clear that increasing the probability of detecting offenses leads to a very sharp decrease in the value of D. If p_o is 0 meaning no offenses are ever detected, D is positive and increases with each offense, so nodes are motivated to cheat. However, D quickly becomes negative for p_o values of 0.05, 0.1 and 0.15. Thus rational nodes will determine that it is in their own best interest to not cheat even when the chance of being caught is small, and the disincentive to cheat increases as the number of offenses increases.

As an alternative to fining and reducing the pay rate of PNs when the CM catches them lying about work performed, we could simply ban users found to commit an offense. Our approach warns misbehaving nodes to mend their ways, and rational nodes will realize that there is no benefit from cheating and cooperate. If individual nodes persist in misbehaving, their pay rate will soon turn negative (due to cuts), which for practical purposes is as effective as banning the node.

3.4.3 Worst Case Analysis

In this section we investigate the maximum expected benefit (D_{max}) that a dishonest node can gain from cheating. We can divide the gain/loss from cheating into three categories: (i) the payoff from undetected cheating (G_{cheat}), (ii) the loss due to fines for detected cheating (L_{fines}), and (iii) the losses accrued from receiving a pay cut due to detected cheating (L_{paycut}).

$$D_{max} = G_{cheat} - L_{fines} - L_{paycut}$$

Given a particular fine, $F = b * R(t)$, we can estimate G_{cheat} and L_{fines} using the following equations:

$$\begin{aligned} G_{cheat} &= \sum_{t=1}^{n} N_{off}(t) * (1 - p_o) * R(t) \\ L_{fines} &= \sum_{t=1}^{n} N_{off}(t) * p_o * b * R(t) \\ G_{cheat} - L_{fines} &= \sum_{t=1}^{n} R(t) N_{off}(t) * (1 - (b+1)p_o) \end{aligned}$$

The maximum value of R(t) is R_l, which we can use to refine our estimate as follows:

$$\begin{aligned} G_{cheat} - L_{fines} &\leq \sum_{t=1}^{n} R_l N_{off}(t) * (1 - (b+1)p_o) \\ G_{cheat} - L_{fines} &\leq R_l * (1 - (b+1)p_o) * \sum_{t=1}^{n} N_{off}(t) \\ G_{cheat} - L_{fines} &\leq R_l * (1 - (b+1)p_o) N_{total} \end{aligned}$$

Here N_{total} is the total number of offenses over the time period and $N_{off}(t)$ represents the number of units of work faked by a cheating node for time slot t .

When a PN is caught cheating, its pay rate is decreased in addition to it receiving a fine, which decreases how much it receives for work it actually performs. Since the max pay rate is capped at R_l, the impact of a pay cut persists only until a PN's pay rate recovers to R_l, which occurs if it is honest or not caught cheating for a period of time. Thus, the impact of pay cuts is minimized when pay raises are frequent. If we assume that all cheating occurs when a PN's pay rate is R_l, we can calculate the minimum loss induced by being caught cheating.

Assume that cheaters receive a pay rate cut of $T_{cut} = o * T_{raise}$. In other words, being caught cheating reduces a PN's pay rate by the equivalent of o pay raises. In this case, we can calculate the loss a PN suffers due to the decreased pay rate from a single detected cheating event ($L_{s-paycut}$) as follows:

$$\begin{aligned} L_{s-paycut} &\geq \sum_{k=1}^{o} N_{actual} * PayRateCut(k) \\ L_{s-paycut} &\geq \sum_{k=1}^{o} N_{actual} * k * T_{raise} * I \\ L_{s-paycut} &\geq \frac{o(o+1)}{2} * T_{raise} * N_{actual} * I \end{aligned}$$

The total number of expected detected offenses can be calculated as $p_o * N_{total}$. Using this, we can refine the previous equation to find L_{paycut}:

$$L_{paycut} \geq \frac{o(o+1)}{2} * T_{raise} * N_{actual} * I * p_o * N_{total}$$

This lets us calculate D_{max} as follows:

$$\begin{array}{rcl} D_{max} & = & G_{cheat} - L_{fines} - L_{paycut} \\ D_{max} & \leq & R_l * (1 - (b+1)p_o)N_{total} \\ & & -\frac{o(o+1)}{2}T_{raise} * N_{actual} * I * p_o * N_{total} \end{array}$$

A rational node is motivated to cheat only if the gain from cheating is more than the gain from behaving honestly. For our variable pay system to deter cheating, we should select system paramaters to ensure that D_{max} is negative. Using the above equation, we can determine what conditions are necessary for D_{max} to be negative as follows:

$$p_o > \frac{R_l}{R_l(b+1) + \frac{o(o+1)}{2}T_{raise} * N_{actual} * I}$$

At first glance, this formula might appear complicated, but we can gain some intuition by solving it for a sample case. If we use the same parameters that were used for Figure 4 (1-day time slots, a pay scale with 10 levels that increases 20% per $T_{raise} = 7$ days, a pay decrease when caught cheating equal to $T_{cut} = 7$ days worth of raises, $N_{actual} = 50$, $R_1 = 0.1$, $I = 0.02$, and $b = 1$), we need only detect cheaters with a probability p_o greater than $\frac{3}{76} = 0.0395$ (roughly 4.0%). This probability remains unchanged for different values of R_1 as long as pay raise increment (I) is 20% of R_1. In contrast, if we assess fines, but not pay decreases, when a PN is caught cheating (the model derived in Section 3.2), the probability p_o of catching a cheater must be greater than 0.5 to build an effective deterrent. Thus, varying pay based on longevity and honesty is an important feature for our incentive model.

3.4.4 System Tuning

Even if we are unable to identify a cheating PN, the CM can obtain an estimate of the frequency of cheating in the system using service-level information. For example, in a collective content distribution system, clients will retry unsuccessful downloads using a different PN, which will lead to multiple PNs requesting credit for same work if the first failure was due to a cheating PN. If a CM observes a particular frequency of undetected cheating, it can tune the parameters used to calculate pay rates and fines (e.g., the fine ratio b, the pay cut ratio o, the rate of pay increases T_{raise}, and the pay rate increment I) to maintain an acceptable profit margin.

3.5 Other Issues

Motivating Critical Nodes: Other factors can be applied to determine a particular node's pay rate. For example, nodes that are particularly important to a given service due to their location or unique resources (e.g., a fat network pipe or extremely high availability) may receive a bonus pay rate to encourage them to remain part of the CM's pool. A service can define threshold criteria that are used to designate a node as an important player, e.g., delivered bandwidth more than X for more than 70% of the time over the last 15 day period. Once a node reaches this threshold, it is designated as special and extra pay levels like $R_{l+1}, R_{l+2}, R_{l+3}$ are made available to them. Additionally T_{raise} can be reduced to provide extra rewards to these nodes.

Multiple Identities: A cheating PN can easily change identities in an attempt to avoid any penalties it receives. Our variable pay rate incentive system is designed to make this behavior unprofitable. A new user starts at a low pay rate, and only gets pay raises after successfully completing work for a considerable time period of time. When a node changes identities, its pay rate drops to the low base pay rate when it rejoins the collective. It would be better off to remain in the collective and behave honestly. We envision CMs only paying nodes every 15 to 30 days based on the amount of work they have performed, similar to how web ad services like Google Adsense [10] are administered. If the probability of detecting offenses is above the low 4%-5% threshold needed to discourage malfeasance, a 15- to 30-day pay period is sufficient to ensure that persistent cheater loses money by cheating. Overall, we expect rational nodes to learn that they earn more from proper behavior than from cheating, and are willing to accept nodes recycling their identity to make a fresh start after they learn this lesson.

4 Related Work

Cheating behaviors have been observed extensively, e.g., free riding in Gnutella [1] and software modifications to get more credits in SETI@home [12]. Our mechanism to handle cheating behaviors based on multi-party accounting is similar to the role of accountability in dependable systems [22]. Our system can use a payout system similar to the one used by Google adsense program [10] that allows website publishers to display ads on their websites and earn money.

Unlike SETI@home [17] and Entropia [5], we harness idle storage and networking resources of PNs in addition to idle processing resources. SETI@home rewards user credits similar to ours, but has no concept of penalties or incentives for long term participation.

Unlike P2P systems like Kazaa [13] or Gnutella [9], we do not assume that PNs are altruistic. Our PNs are rational nodes that are interested in maximizing income, not selflessly helping others.

Many other projects, e.g., BitTorrent [7], have focused on bartering as an incentive model for exploiting idle resources. In such models, nodes typically participate in the system only long enough to perform a particular transaction such as downloading a song. At other times, that node's idle resources are not utilized unless the node's administrator is altruistic. In the collective, a CM will have much larger and more diverse pools of work than personal needs of individual participants; thus a CM will be better able to consume the perishable resources of PNs. PNs, in turn, will accumulate credit for their work, which they can use in the future however they wish (e.g., for cash or for access to services provided by the CM). Many recent projects [14, 18] have applied game theory techniques to build incentives models based on bartering. These projects model nodes as rational self-interested parties similar to us.

Currencies have been used extensively in the systems community in various contexts [21, 4]. Recent projects [20, 19] have used currencies to handle the problem of free riding in peer to peer systems. Though none of these incentives techniques address the issue of motivating nodes to stay in the system for extended durations. Also these projects do not provide any mechanisms for deterring cheating in presence of undetected offenses in the system.

Systems like computational grids [8] also deal with distributed resources at multiple sites, though again their main focus is on trusted and dedicated servers.

We can compare a collective to the formation of organizations/firms in real life [6]. Similar to employees in firms, PNs in a collective need to be motivated to do better work and demotivated from shirking away from work.

5 Conclusions

In this paper, we present an analysis of the incentive model we employ in a distributed designed to harness the idle cpu, network, and storage resources of large pools of untrusted, selfish, and unreliable nodes. Our analysis focuses on two important challenges: ensuring prolonged participation by nodes in the collective and discouraging dishonest behavior. An analysis of the economic underpinnings of the system allowed us to gain important insights into the likely behavior of different players in the system, which we used to derive an incentive model that achieves our goals.

The most important contribution of the paper is to demonstrate how a mix of rewards and punishments can be used to successfully motivate to behave in ways that benefit the collective. We also show how a real system can sustain profitability even in presence of undetected offenses or deviations from the desired behavior, as long as we are able to detect even 4%-5% of dishonest behaviors.

References

[1] E. Adar and B. Huberman. Free riding on gnutella. *First Monday*, 5(10), October 2000.

[2] Akamai. `http://www.akamai.com/`.

[3] G. S. Becker. Crime and punishment: An economic approach. *The Journal of Political Economy*, 76(2):169–217, 1968.

[4] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic models for resource management and scheduling in grid computing. *The Journal of Concurrency and Computation: Practice and Experience*, 14(13-15), 2002.

[5] B. Calder, A. Chien, J. Wang, and D. Yang. The entropia virtual machine for desktop grids. In *International Conference on Virtual Execution Environment*, 2005.

[6] R. H. Coase. The nature of the firm. *Economica New Series*, 4(16):386–405, 1937.

[7] B. Cohen. Incentives build robustness in bittorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, 2003.

[8] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid - enabling scalable virtual organization. *Internation Journal of Supercomputer Applications*, 15(3), 2001.

[9] Gnutella. `http://www.gnutella.com`.

[10] Google Adsense. `http://www.google.com/adsense`.

[11] S. P. Hargreaves-Heap and Y. Varoufakis. *Game Theory: A Critical Introduction*. Routledge, 2004.

[12] L. Kahney. Cheaters bow to peer pressure. *Wired*, 2001.

[13] Kazaa. `http://www.kazaa.com`.

[14] K. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Workshop on Economics of Peer-toPeer Systems*, 2003.

[15] G. J. Mailath and L. Samuelson. *Repeated Games and Reputations*. Oxford University Press, 2006.

[16] A. M. Polinsky and S. Shavell. *The Theory of Public Enforcement of Law*, volume 1 of *Handbook of Law and Economics*. North Holland, Nov 2007.

[17] SETI@home. `http://setiathome.ssl.berkeley.edu`.

[18] J. Shneidman and D. Parkes. Rationality and self-interest in peer to peer networks. In *2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.

[19] M. Sirivianos, X. Yang, and S. Jarecki. Dandelion: Cooperative content distribution with robust incentives. In *NetEcon*, 2006.

[20] V. Vishnumurthy, S. Chandrakumar, and E. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. In *P2P Econ*, 2003.

[21] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *Software Engineering*, 18(2):103–117, 1992.

[22] A. R. Yumerefendi and J. S. Chase. The role of accountability in dependable distributed systems. In *First Workshop on Hot Topics in System Dependability*, 2005.