Systems-Compatible Incentives

Dave Levin, Neil Spring, and Bobby Bhattacharjee

***Abstract*— Selfish participants in a distributed system attempt to gain from the system without regard to how their actions may affect others. To maintain desirable system-wide properties in the presence of selfish users, designers are increasingly turning to the powerful mechanisms offered by economics and game theory. Combining the two fields of economics and systems design introduces new challenges of achieving incentive-compatibility in systems we can deploy in today's Internet.**

In this paper, we explore the interactions between systems and the mechanisms that give users incentives to cooperate. Using findings from recent work on incentive-compatible systems, we discuss several economic mechanisms and assumptions: money, punishment, and altruism. We seek to understand when these mechanisms violate system properties. Among the potential pitfalls we present is a phenomenon we call the price of altruism: altruistic peers can impose a loss of social good in some systems. We also discuss systems-compatible mechanisms that have been used in real, distributed systems, and attempt to extract the underlying design principles that have led to their success.

I. Introduction

Systems that rely on cooperating users have been remarkably successful [43]; BitTorrent [7] is widely used for distributing large files to many downloaders, and Kazaa [27] and Gnutella [19] are used for general peer-to-peer (P2P) file sharing. *Decentralized systems* such as these have no single authority, and instead allow local application of policy. Many decentralized systems assume user cooperation, but few guarantee it. When this fundamental assumption is violated, the global system properties may suffer; selfish participants will attempt to gain at the expense of others [1, 36], and when a system becomes sufficiently large or popular, misbehaving principals may want to break it for notoriety or profit [35].

The greatest feature of decentralized systems—that there is no single administrative entity—is also its weakness; there is no "police force," no entity with a clear mandate to keep the system operational or to kick out the spammers and attackers. Designing systems with incentives for participants to share their resources places this vested interest with the users.

Economic theory has provided invaluable tools in designing such systems to be robust to selfish participants' manipulation. Game theory provides a formal framework with which to understand how selfish, competing parties will interact, while mechanism design yields a rigorous methodology to create games with a particular set of desired outcomes in mind. Such tools are steeped in decades, or in some cases millennia [6], of refinement and application.

However, decentralized systems represent a vastly different set of goals and requirements than prior applications of economics. There have been many proposed incentive-compatible decentralized systems, some with strong provable properties of resilience to strategic manipulation, unfortunately many of them are unrealistic in today's Internet because they require infeasible infrastructure or system-wide knowledge. In other words, although the mechanisms may be incentive-compatible, many are not necessarily *systems-compatible*.

In this paper, we argue for the importance of aligning the goals and assumptions of the two powerful, sometimes divergent fields of systems and mechanism design. We investigate several different, common approaches in the design of selfish systems. We demonstrate potential downfalls to these approaches, along with alternative solutions that have proven successful. We find it both important and intriguing to better understand the underlying design principles that lead to successful, systems-compatible incentives.

Our investigation spans three broad approaches in particular: applying money, applying punishment, and assuming altruism. Viewing these in a systems-first manner leads us to the following insights. We demonstrate that the presence of altruistic participants can harm the social good of a system, a phenomenon we call the *price of altruism*. We also show that participants may have a *disincentive* to post negative (but truthful) feedback about others. Observations such as these lead us to conclude that many of the economic outcomes we take for granted, particularly those based on human nature or pre-existing infrastructure, must be scrutinized when applied to the vastly different space of decentralized systems on the Internet.

The rest of this paper is structured as follows. We introduce the notion of systems-compatible incentives in Section II. We discuss in turn the use of money (§III), punishment (§IV), and altruism (§V). Drawing from these case studies of specific mechanisms, we present in Section VI what we find to be the underlying design principles that have proved successful in deployed, decentralized systems. We conclude in Section VII.

II. Systems-Compatible Incentives

In this section, we present standard goals and requirements of decentralized systems and mechanism design. We also discuss some of the potential limitations and relaxations that arise when combining the two.

A. Systems goals and requirements

The predominate distinguishing factor between distributed systems and decentralized systems is that decentralized systems allow for multiple policy domains. While a single

University of Maryland {dml,nspring,bobby}@cs.umd.edu

principal may deploy a distributed system, a decentralized system consists of multiple principals, each of whom may have their own policies regarding trust, privacy, willingness to contribute, and so on. Designing systems that appeal to a wide range of policies and allows principals to expressively solve contentions in-band [10] is made all the more difficult without knowing these policies a priori.

In addition to addressing various user policies, a successful decentralized system must be able to scale with growing demand and participation. Otherwise, a system's popularity would be its own undoing. Few could have predicted BitTorrent's popularity, but even its early design accommodated a large, ever-growing user base. Scaling to potentially millions of users renders perfect, system-wide information infeasible at best. This in turn limits the applicability of game theory; for instance, players in a standard-form games reason based on common knowledge of all other players' utility functions and strategy sets.

B. *Incentives goals and relaxations*

Mechanism design seeks to provide incentives to selfish but rational principals to follow a given protocol. Such protocols involve truthfully reporting private information, and fairly providing resources to other, competing principals. Without incentives, selfish users will attempt to gain at the potential cost of others, by lying about their private valuations or attempting to distribute their resources unfairly.

It should come as no surprise to the reader that human beings often do not act rationally. While understandably scrutinized in many economic situations, we find that it is generally reasonable to assume perfect rationality of system participants. We are not positing anything about the rationality of the users themselves, rather that it is the software, not the end-users, making the protocol-level decisions. A strategic BitTorrent client, for instance, may be designed to precisely follow a utility-maximizing strategy that end-users would not or could not perform on their own accord.

C. *Systems-compatible incentives*

An incentives mechanism is compatible with a decentralized system if it does not violate the assumptions and requirements of the system. In particular, to be systems-compatible, an incentive mechanism does not impose undue communication or computation burden, and is itself decentralized. We will demonstrate in the remainder of this paper that many incentive mechanisms require a centralized, often trusted principal. Centralized solutions do not allow for the complete expression of policy for all peers; for instance, they may not be able to choose with which set of peers to share a file, issue their own currency, or create Sybils [14]. Thus, such mechanisms are not systems-compatible.

This definition of systems-compatibility is intentionally loose, so as to accommodate a broad range of systems. For example, computationally intense incentive mechanism may be compatible with a system run at end-hosts, but would be incompatible if run on today's Internet routers.

A thorough understanding of what limitations a system places on mechanisms, as well as what windfalls it brings, can help guide designers toward more robust systems. For example, Afergan demonstrated that a detail as seemingly mundane as the number of bits used to represent prices can have profound effects on participants' strategies [3]. We believe that systems-compatibility is an increasingly important consideration to take in the study and design of incentive-compatible systems.

III. Money

Money is the cornerstone of economy. It is well understood in economics literature and practice how to leverage money to obtain a wide array of cooperative user behavior. We briefly review the vast benefits of applying money to settings of selfish participants. We also discuss some of the reasons why, unfortunately, money is not systems-compatible. We close this section with alternatives to true monetary systems, and directions for future work.

Throughout this paper, when we speak of "money" we are referring to "real" currency, such as dollars or euros, that have value outside of a given system.

A. *What does money buy systems?*

Money gives rise to immensely powerful mechanisms and results that are at best difficult without it. A common use of money is to elicit truthful statements from participants who have incentive to lie about their private information. One of the fundamental goals of auctions, for instance, is for participants to bid their true valuation of a good, so that the auctioneer can award the good to the participant with the highest valuation [11, 20, 45]. Peers in a decentralized system have private information, such as their willingness to perform work [29], job priorities [28], or valuation of a good [2, 3, 17]. Money could yield tangible benefits to these various systems problems, and some systems have been designed with money as a system primitive [2, 3, 17, 33, 49].

One of the most challenging problems of designing incentive-compatible systems is in handling cases of asymmetric interest. Two peers p and q exhibit *asymmetric interest* when p desires a service from q, but q desires nothing from p—or more generally, q desires something of greater value from p than p can offer in exchange. One of the major benefits of money is that it adds liquidity; regardless of what interest peers have in one another in the context of the system, they are always interested in money. In effect, systems that allow for monetary payments do not suffer from asymmetric interest. Lottery trees [15] are a prime example of overcoming asymmetric interest. The goal of lottery trees is to encourage users to join, and solicit others to join, a system they may have otherwise no intention of joining. The reward to these users is the chance to enter a lottery for a tangible reward outside the system, which we consider equivalent to money.

B. *Is money compatible with decentralized systems?*

Given the many potential benefits of including money in a decentralized system, we seek to understand the extent to which money can be applied in real, deployable systems. Prior work has demonstrated what is possible when applying economic operating points to networks [2, 8, 17, 33, 49]. However, none of these prior systems has experienced widespread deployment. This is in part due to the fact that they require money exchange at the protocol's time intervals, typically on a per-packet basis. Supporting such payments requires extensive infrastructure support, typically through a centralized money-clearing system, which is difficult to deploy and scale with increasing demand. Certainly, ISPs make extensive use of money, but typically on much larger, more easily supported time scales.

Another reason money is difficult to incorporate into decentralized systems is the legal concerns it raises. In the context of Lottery Trees, Douceur and Moscibroda briefly review some of the extensive legal matters of running a lottery, including the disparities of laws from region to region [15].

Last, we argue that monetary systems are inherently centralized. Users must access a typically centralized money clearing mechanism, such as PayPal, or a centralized bank. This is a clear violation of many decentralized systems' goal of having no centralized trust domain.

C. Alternatives to money

1) Digital currency: In an effort to maintain the basic semantics of money without requiring users to pay for the service, various digital currency schemes have been proposed. Digital currency differs fundamentally from money in that the former does not have value outside of a system. A major ramification of this is that peers do not have as much incentive to preserve digital currency, because they cannot remove it from the system for external goods. This may in fact be beneficial for a system, in that it creates a closed economy from which users cannot arbitrarily remove liquidity.

One of the fundamental technical challenges of a digital currency system is ensuring that users cannot double-spend. Paper currency is difficult to double-spend because of the considerable effort in making bills difficult to copy or forge. However, in a digital setting, copying is trivial. Solutions to double-spending typically involve a third-party mediator to track the history of individual units of currency [18, 46]. While not necessarily introducing a high barrier of entry—peers would not, for instance, have to register a credit card—providing such mediators does requires considerable systems infrastructure.

Some digital currency systems allow users to generate their own currency [18, 46]. This raises another fundamental technical challenge: ensuring that users do not flood the system with currency, thereby devaluing it. A standard approach is to rate-limit users' currency creation by requiring a proof of work [16], such as the solution to a cryptographic puzzle [40].

2) Mechanism design without money: Mechanism design without monetary payments is a rich area [39, Chapter 10] that, for the reasons discussed in this section, is receiving renewed interest. One way to broadly view this line of work is as an investigation into the role that money plays in the positive results obtained from mechanisms. One such result, *money burning* [23], acknowledges the infeasibility of incorporating money in a networked system, and proposes replacing money payments with decreased quality of service. For example, rather than place monetary bids in an auction, peers could bid the level of service degradation they are willing to accept. Clearly, such an approach comes at the cost of decreased social good; Hartline and Roughgarden demonstrate that optimal mechanisms typically involve money transfer [23].

3) Exploiting users' impatience: Although a general-form replacement of money may ultimately come at a cost of performance, there remain scenarios in which a money-less mechanism can ensure truthful reporting with very little negative impact to the system. We discuss here one such example: leader election among impatient participants.

Leader election can be viewed as a public good game; at least one of the players must pay the cost of being the leader so that all may benefit. In a selfish environment, the challenge is to elicit truthful reports of peers' costs to act as a leader; with no mechanism in place, all participants would have incentive to inflate the costs they would incur so as to avoid having to serve other peers.

Lee et al. observe that in multi-hop wireless networks, truthful reporting can be obtained without monetary payments; in fact, it can be obtained without explicitly reporting any values whatsoever [29]. The result leverages the specific application domain: wireless nodes wish for routing paths to be found quickly, else they may experience prolonged disconnected operation. This observation allows for the application of the *volunteer's timing dilemma* (VTD) [48]. In VTD, one out of a set of players must volunteer to complete a job; none of the players wish to be the volunteer, but all of them benefit from there being a volunteer, and benefit more the more quickly someone volunteers. The VTD game translates a player's cost-to-volunteer into an amount of time to *wait* until volunteering; the game itself consists of silent waiting until one player—the one with the lowest cost—volunteers. Lee et al. demonstrate how to apply this to a practical system; the game is periodically run, so as to cycle the leaders and maintain high system-wide battery levels.

Interestingly, even without money, participants' dominant strategy is to act truthfully. This is similar to money-burning mechanisms in that participants degrade the network's service by silently waiting for their neighbors to volunteer. The system by Lee et al. demonstrate that, in practice, this sub-optimality can be amortized over time, and serves as a considerable improvement to extensive infrastructure changes [29].

To summarize, money is a remarkably powerful tool in ensuring cooperation among peers who may otherwise have no reason to interact. It is difficult to replace money in general, but recent work has demonstrated the power of tailored mechanisms.

IV. Punishment

An alternative to the carrot of money is the stick of punishment. Punishment comes in forms both extreme—such as jamming a wireless channel [31]—and more subtle—such as propagating negative reputation information [26, 30, 47]. From a systems design perspective, punishment is appealing in the sense that it typically does not require an extensive infrastructure or high barrier of entry like those of a monetary system. We discuss in this section the properties of a punishment-based mechanism game theory requires to ensure provable incentive to cooperate.

A. Credible threats

In a punishment-based mechanism, it is not the punishment itself that maintains cooperation. Rather, it is the *threat* of punishment that keeps selfish participants from defecting from the protocol. For a potential defector to take a threat from user u seriously, it must be clear to the defector that u would be willing to follow through with the punishment. In game theory parlance, this means that threats of punishment must be *credible*.

In other words, peers require incentive to punish one another. A mere act of "revenge" is not necessarily sufficient.

Further, the threat must typically be to punish for at least as much as the defector gained from performing a punishable act. For instance, if a peer cheated to obtain an additional 10 units of utility, then a subsequent punishment would have to incur a cost greater than 10 to ensure that the peer has enough disincentive to not cheat.

B. Punishing with negative reputation

Reputation systems allow participants to exchange with one another information regarding the interactions they have had with others. When two peers interact for the first time, they can draw from others' experiences to infer how trustworthy one another is. There are two general forms of reputation: positive and negative. Peers clearly have incentive to accrue positive reputation, and considerable work has gone into ensuring that they cannot arbitrarily inflate their positive reputation [9, 26].

In this section, we focus on negative reputation. When a peer p is mistreated by another peer m, p forms a local, negative view of m, and may either degrade service to m or cease communicating with m altogether. Sharing this experience with others, in other words establishing a negative reputation for m, is a powerful way to quickly weed out free riders and malicious peers from the system.

We study the feasibility of such an approach by first observing that providing negative reputation is a form of punishment. It is rather subtle, especially when compared to BAR gossip's blacklisting [34] or jamming a wireless channel [31]. However, it is indeed a form of punishment: peers implicitly threaten one another with posting negative "feedback," and having accrued a negative reputation can result in a loss of utility, such as fewer willing traders. As with any form of punishment, it is crucial to understand whether or not threats thereof are credible.

1) Is negative reputation a credible threat?: Suppose that peer p has been mistreated by peer m. Although p has implicitly made the threat of punishing m by reporting a negative reputation score, does p have incentive to follow through with this threat?

In many systems, peers compete with one another to obtain service from others. BitTorrent peers, for example, attempt to upload more to their neighbors than others do, so as to receive reciprocal bandwidth [12, 41]. If p realizes that m is a "lost cause"—that m does not return any data or that m consistently returns corrupted data—then p certainly knows to no longer upload to m. Instead, p will upload to a set of peers G he has observed to be good. However, p may wish for *other peers* to upload to m, as it diverts bandwidth they may have otherwise spent at peers in G.

In systems such as this, peers have a disincentive to report "bad deals," as it may draw greater competition for good deals.

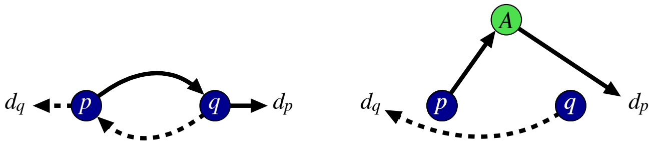
This raises two natural questions: can systems provide peers incentive to truthfully report *all* interactions they have had with others, and can they do so in a systems-compatible manner? To the best of our knowledge, the most apropos systems that achieve this are accountability systems, such as PeerReview [22] and Nysiad [24]. These systems augment an existing protocol with an "accountability layer" that adds signed digests of messages sent in the underlying protocol. Both of these systems require considerable infrastructure, such as a PKI, and communication overhead, typically superlinear in the number of messages from the underlying protocol.

This result demonstrates that even a seemingly innocuous form of punishment like that of forwarding negative reputation can in fact require extensive supporting infrastructure to ensure truthful, cooperative behavior.

V. Altruism

Much of the study and pursuit of systems that are resilient to selfish participants arose from the finding of widespread free-riding in the Gnutella file sharing system [1, 25]. Because Gnutella provided no incentives to share—certainly none to overcome the potential legal issues of sharing copyrighted data—the majority, 70%, of users downloaded files without giving any in return. We focus in this section not on this majority of free-riders, but on the remaining peers who altruistically provided service.

It is tempting to assume that at least some fraction of participants are altruistic when designing an otherwise incentive-compatible system. However, it is important to recall that it is typically not the users themselves making the system-level decisions, but rather the software itself. Many users choose to use software that promises better performance, perhaps because they are unfamiliar with the potential impact to the rest of the system, or because, in a faceless environment, they simply do not experience the same social pressures that drive altruism in person. We therefore find it safest to design a system that correctly and efficiently performs even when *all* participants are selfish.



(a) Without altruism, selfish peers must exchange resources to benefit.

(b) Altruistic peers potentially remove opportunities for selfish peers to exchange.

Fig. 1. The price of altruism. Were p to have unbounded demand, there would remain the possibility for cooperation in the presence of an altruistic peer.

Nonetheless, there may be some users who do wish to give to a system altruistically. However, few systems explicitly design for how an altruistic peer should act, merely relegating them to performing any work requested of them without asking anything in return. We demonstrate in this section that such blind altruism can result in a *decrease* in social good, a phenomenon we term the *price of altruism*.

A. Bounded demand

We demonstrate this result within the context of systems with bounded demand. We say that a system has *bounded demand* if there exists an upper bound on users' utility with respect to the amount of resources they obtain. Examples of systems with bounded demand include BitTorrent—a BitTorrent peer's demand is bounded by their download speed—and video streaming, wherein demand is bounded by the video's stream rate.

We consider, as a concrete case study, forwarding in Internet routing overlays. Suppose users p and q have a bounded amount of traffic they wish to send to destinations d_p and d_q, respectively, as in Figure 1. In an Internet routing overlay, peers do not necessarily forward directly to their destinations, rather to other peers in the overlay, who then forward the traffic on their behalf. Overlay routing systems such as RON [4], SOSR [21], and PeerWise [38] have demonstrated how this simple premise can improve reliability, recovery times, and latencies over standard Internet routing.

Now suppose that p and q, both selfish users, could forward traffic for one another (Fig. 1(a)). In the context of PeerWise, these users are said to exhibit *mutual advantage* [37]. We propose, as a simple definition of an overlay routing system's social good, the number of end-to-end connections forwarded through the system.[1] Were p and q to forward for one another, the social good of this instance would reach its maximal value of 2: both end-to-end connections would be forwarded through the system.

B. The price of altruism

Let us now consider the effects of introducing an altruistic peer A (Fig. 1(b)). A is willing to forward for others, and has no demand of her own. If A can serve both p and q's demands, then both selfish users will forward through A, maintaining a maximal social good of 2. However, there are several reasons for which A would only be able to serve a single peer's demands; A may have insufficient upload capacity, or in the case of some routing overlays, A may simply not provide improvements over standard Internet routing for both p and q. If A can only forward for, say, p, then p has no incentive to service q's request. As a result, there are no peers able and willing to serve q's demand, and q cannot forward his traffic through the routing overlay. The social good as a result of introducing an altruistic peer has therefore decreased by half.

We define the *price of altruism* as the ratio of social good lost as a result of introducing altruistic participants. Specifically, the price of altruism is one minus the ratio between the social good obtained in the presence of altruistic users and the maximum social good obtained from all selfish users:

$$\mathsf{PoA}\ell \overset{\mathsf{def}}{=} 1 - \frac{\text{Social good with altruism}}{\text{Social good without altruism}} \qquad (1)$$

In the prior example, the price of altruism is $1/2$; half of the social good has been lost as a result of adding an altruistic user. Altruism reduces social good in this scenario because p can provide a service to q that A cannot, but A removes p's incentive to do so. More concretely, because p had bounded demand, q has nothing to offer to p as incentive to forward for q.

C. Overcoming the price of altruism

We review two promising approaches to overcoming the price of altruism. Ultimately, the price of altruism arises when there are users users who *could* service others' requests but who do not because their demands are met elsewhere, Because it is the altruists' actions that result in this free riding, both of the approaches we discuss here focus on actions that the altruistic peer can take to yield cooperation.

1) Super-seeding in BitTorrent: The price of altruism is prevalent in BitTorrent. *Seeders* upload even after having completed downloading the file. Because they give out blocks for free, a viable strategy—indeed, a *dominant* strategy—is for all peers to attempt to connect to as many seeders as possible. This is the so-called large-view exploit [44], and the insight behind the BitThief client [36].

Altruistically providing blocks in exchange for nothing can lead to a decrease in selfish peers' cooperation. Some peers may simply download from seeders while uploading nothing in return; this is increasingly common in nations where uploading illegal content is prosecuted, but strictly downloading is not. Further, some peers may be demand-bounded, pegging their download bandwidth by contacting a sufficient number of highly provisioned seeders.

Super-seeding was proposed as a means of simultaneously allowing for altruistic seeders while ensuring that the peers to whom they are uploading are not shirking their responsibilities. A super-seeder uploads a block b to a peer for free, and does not upload anything else to that peer until he observes that another peer has b. While this particular

[1] Any excess demand that has not been serviced by the routing overlay can be sent via standard Internet routing, at a potential loss of performance [32].

strategy is open to several attacks, such as Sybil attacks [14] and colluding peers, the tenet of ensuring that peers who benefit from altruism "pay it forward" to others is promising.

2) Long-term incentives: One of the fundamental open problems in systems consisting of selfish participants is that of providing incentives in cases of asymmetric interest. Altruism is appealing, and may in fact deceptively appear to be necessary, when a peer needs another's participation yet has nothing to offer in return.

We view systems that are currently designed with altruism assumed as opportunities to apply what we call *long-term incentives*. A long-term incentive spans multiple instances of a system, such as providing incentives for peers to act as a seed in one BitTorrent swarm in exchange for help in future swarms.

Long-term incentives have been proposed in several application domains. One example of this approach is the Samsara backup system [13]. Asymmetric interest arises in this domain when one peer wishes to backup its data while none of its neighbors have anything they wish to backup. An uninterested Samsara peer p backs up its interested neighbor q's data in exchange for q agreeing to return the favor when p has data he wishes to backup. To ensure q has the capacity to return the favor, p effectively stores garbage data at q, and overwrites it with meaningful data in the future. Another example is one-hop reputation systems [42], in which uninterested peers perform work in exchange for the assurance of future payback. The insight behind limiting reputation to one-hop comes from the observation that the diameter of the interaction graph among BitTorrent peers is typically small; a more complex means of exchanging reputation information does not appear to be necessary.

Perhaps the most prevalent example of long-term incentives is that of BitTorrent communities [5], wherein each user's upload ratio is stored on a trusted, centralized server. We believe that one of the reasons this mechanism is powerful and broadly applicable is that it provides a simple metric—total bytes provided versus total bytes consumed—that can be used in multiple contexts. User contribution can easily be tracked over time, users have a clear understanding of how to improve their rank, and the free riders are easily distinguished.

Common to each of these long-term incentive mechanisms is the need to maintain state across separate system instances. The infrastructure to support this is non-trivial, and is in part why existing solutions either require a trusted third party [5] or have yet to receive wide adoption [13, 42]. However, a unified infrastructure that supports interaction not just across separate instances of the same system, but across multiple different systems is a promising area of future work.

VI. Design Principles

Throughout this paper, we have discussed mechanisms with varying degrees of systems-compatibility. We summarize our findings in this section by attempting to extract the design principles that have lead to successful, systems-compatible incentives.

A. Money

In Section III, we discussed the infeasibility of incorporating a "true" monetary system into a decentralized system, as well as the system performance costs of general-form money replacements. Successful alternatives for money have leveraged domain-specific properties, such as wireless nodes' limited battery life and desire to maintain connectivity [29]. This leads us to the following design principle:

> *Avoid money when possible; look for domain-specific replacements that are backed by an intrinsic good.*

B. Punishment

Extreme forms of punishment have been demonstrated to be powerful in a theoretical sense [31], but present opportunities for acts of malice. In Section IV, we demonstrated that there are also more subtle forms of punishment, such as negative feedback in a reputation system. The necessity of making a threat of punishment credible leads us to the following design principle:

> *Prefer carrots to sticks when possible. Expose subtle forms of punishment, and ensure the threats thereof are credible.*

C. Altruism

Altruism can be a great boon to a system, but as we demonstrated in Section V, relying on good will alone can result in a decrease in system performance. We also argued that altruism in today's systems represents an opportunity for long-term, cross-system incentives. We conclude with the following design principle:

> *Give peers the opportunity to be altruistic, but ensure recipients of this altruism "pay it forward." Replace assumptions of altruism with long-term incentives if possible.*

VII. Conclusions

Designing systems consisting of selfish participants is an increasingly interdisciplinary field. Understanding the interactions between systems design and mechanism design is, we believe, crucial to these systems' continued success. We have presented several insights from applying mechanism design to decentralized systems. These insights demonstrate that the setting of a decentralized system can yield outcomes that may be unexpected in standard economic settings. This work represents a small step toward the much larger vision of building practical systems where users are both safe and motivated to contribute their resources to others.

Acknowledgments

We thank Vijay Gopalakrishnan, Dov Gordon, Seungjoon Lee, and Cristian Lumezanu for their insightful comments and helpful discussions. This work was supported in part by NSF Awards CNS 0626629 and ITR 0426683, and a Microsoft Live Labs fellowship.

References

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

[2] M. Adler and D. Rubenstein. Pricing multicasting in more practical network models. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2002.

[3] M. Afergan. Using repeated games to design incentive-based routing systems. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2006.

[4] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proc. ACM Symposium on Operating Systems Principles (SOSP)*, 2001.

[5] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu. Influences on cooperation in BitTorrent communities. In *Proc. Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, 2005.

[6] R. J. Aumann and M. Maschler. Game theoretic analysis of a bankruptcy problem from the Talmud. *Journal of Economic Theory*, 36(2):195–213, 1985.

[7] BitTorrent. `http://www.bittorrent.com/`.

[8] R. Chakravorty, S. Banerjee, S. Agarwal, and I. Pratt. Mob: A mobile bazaar for wide-area wireless services. In *Mobicom*, 2005.

[9] A. Cheng and E. Friedman. Sybilproof reputation systems. In *Proc. Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, 2005.

[10] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in cyberspace: Defining tomorrow's Internet. In *Proc. SIGCOMM Conference on Data Communication*, 2002.

[11] E. Clarke. Multipart pricing of public goods. *Public choice*, 11, 1971.

[12] B. Cohen. Incentives build robustness in BitTorrent. In *Proc. Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, 2003.

[13] L. P. Cox and B. D. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proc. ACM Symposium on Operating Systems Principles (SOSP)*, 2003.

[14] J. R. Douceur. The Sybil attack. In *Proc. Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[15] J. R. Douceur and T. Moscibroda. Lottery Trees: Motivational Deployment of Networked Systems. In *Proc. SIGCOMM Conference on Data Communication*, 2007.

[16] C. Dwork, M. Naor, and H. Wee. Pebbling and proofs of work. In *CRYPTO*, 2005.

[17] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In *PODC*, 2002.

[18] F. D. Garcia and J.-H. Hoepman. Off-line karma: A decentralized currency for peer-to-peer and grid applications. In *Applied Cryptography and Network Security Conference (ACNS)*, 2005.

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

[20] T. Groves. Incentives in teams. *Econometrica*, 41, 1973.

[21] K. Gummadi, H. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall. Improving the reliability of Internet paths with one-hop source routing. In *Proc. Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

[22] A. Haeberlen, P. Kuznetsov, and P. Druschel. PeerReview: Practical accountability for distributed systems. In *Proc. ACM Symposium on Operating Systems Principles (SOSP)*, 2007.

[23] J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *STOC*, 2008.

[24] C. Ho, R. van Renesse, M. Bickford, and D. Dolev. Nysiad: Practical protocol transformation to tolerate Byzantine failures. In *Proc. Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[25] D. Hughes, G. Coulson, and J. Walkerdine. Free riding on Gnutella revisited: The bell tolls? *IEEE Distributed Systems Online*, 6(6):1, 2005.

[26] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *WWW*, 2003.

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

[28] K. Lai, L. Rasmusson, E. Adar, S. Sorkin, L. Zhang, and B. A. Huberman. Tycoon: An implemention of a distributed market-based resource allocation system. *Multiagent and Grid Systems*, 1(3):169–182, Aug. 2005.

[29] S. Lee, D. Levin, V. Gopalakrishnan, and B. Bhattacharjee. Backbone construction in selfish wireless networks. In *Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2007.

[30] S. Lee, R. Sherwood, and B. Bhattacharjee. Cooperative peer groups in NICE. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2003.

[31] D. Levin. Punishment in selfish wireless networks: A game theoretic analysis. In *Proc. Workshop on the Economics of Networked Systems (NetEcon)*, 2006.

[32] D. Levin, R. Baden, C. Lumezanu, N. Spring, and B. Bhattacharjee. Motivating participation in Internet routing overlays. In *Proc. Workshop on the Economics of Networked Systems (NetEcon)*, 2008.

[33] D. Levin, A. Bender, C. Lumezanu, N. Spring, and B. Bhattacharjee. Boycotting and extorting nodes in an internetwork. In *Proc. Joint Workshop on the Economics of Networked Systems and Incentive-Based Computing (NetEcon+IBC)*, 2007.

[34] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. BAR gossip. In *Proc. Symposium on Operating Systems Design and Implementation (OSDI)*, 2006.

[35] J. Liang, N. Naoumov, and K. W. Ross. The index poisoning attack in P2P file sharing systems. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2006.

[36] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer. Free riding in BitTorrent is cheap. In *Proc. Workshop on Hot Topics in Networks (HotNets)*, 2006.

[37] C. Lumezanu, R. Baden, D. Levin, B. Bhattacharjee, and N. Spring. Symbiotic relationships in Internet routing overlays. In *Proc. Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[38] C. Lumezanu, D. Levin, and N. Spring. PeerWise discovery and negotiation of faster paths. In *Proc. Workshop on Hot Topics in Networks (HotNets)*, 2007.

[39] N. Nisan, T. Roughgarden, and Éva Tardos. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[40] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu. Portcullis: Protecting connection setup from denial-of-capability attacks. In *Proc. SIGCOMM Conference on Data Communication*, 2007.

[41] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent? In *Proc. Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[42] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson. One hop reputations for peer to peer file sharing workloads. In *Proc. Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[43] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of Internet content delivery systems. In *Proc. Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[44] M. Sirivianos, J. H. Park, R. Chen, and X. Yang. Free-riding in BitTorrent networks with the large view exploit. In *Proc. Workshop on Peer-to-Peer Systems (IPTPS)*, 2007.

[45] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16, 1961.

[46] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. KARMA: A secure economic framework for P2P resource sharing. In *Proc. Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, 2003.

[47] K. Walsh and E. G. Sirer. Experience with an object reputation system for peer-to-peer filesharing. In *Proc. Symposium on Networked Systems Design and Implementation (NSDI)*, 2006.

[48] J. Weesie. Incomplete Information and Timing in the Volunteer's Dilemma. *Journal of Conflict Resolution*, 38(3), 1994.

[49] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2003.