Distributed Computing Meets Game Theory: Combining Insights From Two Fields

Ittai Abraham
MSR Silicon Valley
ittaia@microsoft.com

Lorenzo Alvisi
UT Austin
lorenzo@cs.utexas.edu

Joseph Y. Halpern
Cornell University
halpern@cs.cornell.edu

Traditionally fault tolerance and security have divided processes into "good guys" and "bad guys". Work on fault tolerance has focused on assuring that certain goals are met, as long as the number of "bad guys" is bounded (e.g., less than one third or one half of the total number of players).

The viewpoint in game theory has been quite different. There are no good guys or bad guys, only rational players who will make moves in their own self-interest. Making this precise requires assigning payoffs (or utilities) to outcomes. There are various *solution concepts* in game theory—predictions regarding the outcome of a game with rational players. They all essentially involve players making best responses to their beliefs, but differ in what players are assumed to know about what the other players are doing. Perhaps the best-known and most widely-used solution concept is *Nash equilibrium* (NE). A *profile* $\vec{\sigma}$ of strategies—that is, a collection of strategies consisting of one strategy σ_i for each player i—is a Nash equilibrium if no player can improve his payoff by changing his strategy unilaterally, even assuming that he knows the strategies of all the other players. In the notation traditionally used in game theory, $\vec{\sigma}$ is a Nash equilibrium if, for all i and all strategies τ_i for player i, $u_i(\vec{\sigma}_{-i}, \tau_i) \leq u_i(\vec{\sigma})$: player i does not gain any utility by switching to τ_i if all the remaining players continue to play their component of $\vec{\sigma}$. (See a standard game theory text, such as [23], for an introduction to solution concepts, and more examples and intuition.)

Both the game theory approach and the distributed computing approach have something to recommend them. In fact, for many applications, it is important to take *both* fault tolerance and strategic behavior into account. That is, we are interested in solution concepts that consider strategic behavior while maintaining a level of fault tolerance.

In this paper, we briefly review the approaches to combine these concerns taken in two papers, [1] and [5], and discuss more generally the question of accounting for strategic behavior in distributed computing, and its implications.

As pointed out in [1], extending traditional game theory to incorporate fault tolerance really involves two separate steps. First, it is necessary to move beyond deviations by a *single* player, which are the focus of Nash equilibrium (and all other standard solution concepts in game theory), to allow deviations by groups of players. To understand the point, consider the following simple example.

Example 1. *Suppose that there are $n > 1$ players, all of whom must play either 0 or 1. If they all play 0, then everyone gets a payoff of 1; if exactly two players play 1, then those two players get a payoff of 2, while the remaining players get a payoff of 0; otherwise; everyone gets a payoff of 0. Clearly, everyone playing 0*

is a Nash equilibrium. No single player can do better by deviating; if one player deviates and plays 1, then all players get 0. On the other hand, a coalition of two players can do better by deviating and playing 1.

In [1], an equilibrium is defined to be k-*resilient* if the members of no group of k players can all do better by deviating. That is, $\vec{\sigma}$ is k-resilient if, for all sets K of players with $|K| \leq k$ and all strategy profiles $\vec{\tau}$, for some $i \in K$, we have $u_i(\vec{\sigma}_{-K}, \vec{\tau}_K) \leq u_i(\vec{\sigma})$: no matter how the players in K deviate, it cannot be the case that they all gain from the deviation. The Nash equilibrium in Example 1 is not 2-resilient.

The idea of resiliency is an old one, going back to Aumann [6]. It is surprising how little study the notion has received in the game theory literature. Perhaps one reason is that, while Nash [22] showed that a Nash equilibrium always exists (this was his thesis result which eventually led to him getting the Nobel prize), in general a k-resilient equilibrium does not exist for $k \geq 2$.

While resiliency deals with resistance to coalitions of rational players, it does not fully capture the idea of fault tolerance. The problem is that resilience assumes that even the members of the deviating coalition are strategic and will not deviate unless is it is in their best interests to do so. But in large systems we may well see deviations that cannot be explained, at least in what appears to be the most obvious way, by strategic behavior. For example, in a peer-to-peer network like KaZaA or Gnutella, it would seem that no rational player should share files. Whether or not you can get a file depends only on whether other people share files; on the other hand, it seems that there are disincentives for sharing (the possibility of lawsuits, use of bandwidth, etc.). Indeed, studies of the Gnutella network have shown almost 70 percent of users share no files and that nearly 50 percent of responses are from the top 1 percent of sharing hosts [3]; nevertheless, people *do* share files.

One reason that people might not respond as we expect is that they have utilities that are different from those we expect [1, 19]. Someone may like the feeling of providing everyone else with the music they are listening to. At the other end of the spectrum, someone may instead actively sabotage the service, either out of concern for copyright violations or sheer (perverse) pleasure. In other cases, "strange" behavior may be explained by faulty computers, or by users who do not know how to use their software correctly.

Whatever the reason, it seems important to design protocols that tolerate such unanticipated behavior, so that the payoffs of the users with "standard" utilities are not affected by the nonstandard players using different strategies. This observation motivates the notion of *immunity*, introduced in [1]. Informally, an equilibrium is t-*immune* if the non-deviating players are not made worse off by arbitrary (possibly coordinated) deviations by up to t players. The following example may help illustrate the difference between resilience and immunity.

Example 2. *Consider a group of n bargaining players. If they all stay and bargain, then all get a payoff of 2; anyone who goes home gets a payoff of 1; and anyone who stays if not everyone stays get a payoff of 0. Clearly, everyone staying is a k-resilient Nash equilibrium for all $k < n$. If everyone stays, then they all get a payoff of 2, the highest possible payoff. On the other hand, everyone staying is a very "fragile" equilibrium. It is not immune to even one "irrational" player going home; that is, it is not even 1-immune.*

In [1], the notions of immunity and resilience are combined into a notion of *robustness*. Intuitively, an equilibrium is (k, t)-robust if it is both k-resilient and t-immune. Nash equilibrium is the special case of $(1, 0)$-robustness; more generally, k-resilience is just $(k, 0)$-robustness, and t-immunity is $(0, t)$-robustness. However, (k, t)-robustness is more than just k-resilience and t-immunity; in particular, it says there is no deviation by a coalition of k players and by t strange players that gives all k players a higher payoff. In particular, (k, t)-robustness does not allow deviations, where the t players can help the k deviating players get a higher payoff (even if it does not hurt the remaining players), nor does it allow deviations where

players can take advantage of knowing who the "strange" players are (Example 3 shows how this can have an effect.) (The reader is encouraged to consult [2] for the formal definition.)

From a distributed computing perspective, (k, t)-robustness is a compelling solution concept for at least two reasons. First, by allowing the t strange players to adopt any arbitrary strategy, (k, t)-robustness fully captures the protean nature of Byzantine behavior [18]. Second, (k, t)-robustness offers complete protection from the (possibly malicious) capriciousness of Byzantine nodes with the guarantee of no regrets: because the equilibrium strategy remains preferable irrespective of who the t Byzantine nodes are and how they behave, (k, t)-robustness guarantees that the k colluding players will never find themselves second-guessing their decision even if the identities and strategies of the Byzantine nodes become known.

Unfortunately, (k, t)-robustness may be difficult, if not impossible, to achieve in many practical distributed systems [7, 30]. In [2], some generic impossibility results are given for games where (k, t)-robust equilibria cannot be attained, even though they can be attained with the help of a mediator. These results consider the relationship between n, the total number of players, k, and t. In some cases, the results use known impossibility results from Byzantine agreement; in other cases, new techniques are needed. The following example gives further insight, showing that (k, t) equilibrium cannot be attained if $k = 1$ and $t = 1$, no matter how large n is.

Example 3. *Consider a distributed system with n players, running a fault-tolerant protocol that provides some desirable functionality despite up to $t < n$ Byzantine failures. Think, for instance, of a system where the players run a consensus protocol, and the desirable functionality is that no two correct players decide different values. Assume that the protocol requires some pairwise communication between players, and that bandwidth is not free.*

We could model this system as a game as follows: players obtain benefit by implementing the desirable functionality provided in the original system; attaining this functionality requires communication between some pair of players, and communication incurs some cost; finally, the functionality can be attained despite t strange players. The details of the utility function are not relevant here; all that matters is that a player gets positive utility if the functionality is attained, and loses some (small) utility for each message sent.

Can we build a (k, t)-robust equilibrium in this game? There is a trivial equilibrium: no one sends any messages. Clearly this is an equilibrium. No rational player is motivated to send a message if no one else is going to send a message. The more interesting question is whether there is a nontrivial robust equilibrium, where the functionality is attained (at least sometimes). As we now show, we cannot do this if $k > 0$, no matter how large n is. For simplicity, suppose that $k = t = 1$. In the resulting $(1, 1)$-robust equilibrium, no player should receive a higher payoff from deviating unilaterally, irrespective of the strategy adopted by a single strange player. Suppose, by way of contradiction, that there is a $(1, 1)$ robust equilibrium $\vec{\sigma}$ that attains the desirable functionality. Since it does so, there must be some communication. Let the first message sent in some execution of $\vec{\sigma}$ be a message from i to j. (The algorithm may randomize, so this may not be the first message in all executions.) Now suppose that j is "strange", and never sends and never communicates with anyone. Clearly i can do better if j follows this strategy by never communicating with j; i's outcome will not change (since j never communicates with anyone), and i's communication costs are lower. Thus, a rational player i can gain by deviating from σ_i if j follows the strategy of never communicating. Since $(1, 1)$-robustness requires that no rational player can gain by deviating, no matter what the strange player does, $\vec{\sigma}$ is not $(1, 1)$-robust.

The problem here is that (k, t)-robustness guarantees k-resilience irrespective of the actions of the t Byzantine players. While this promise guarantee a mathematically well-defined way to defend against the arbitrary nature of Byzantine behavior, it seems that it will be hard to achieve if, for example, communication is not free.

The question of what the most appropriate solution concept is for questions such as these remains open. In general, equilibrium concepts assume that players are always making a best response to their beliefs, but that leaves open the question of (a) how the beliefs are represented, (b) what counts as a best response, and (c) under what conditions the response is made. We explain these points (especially the last) by example. The standard approach in economics is to assume that players' beliefs are represented by a single probability distribution, and "best response" means "response that maximizes expected utility, given those beliefs". In Nash equilibrium, the only source of probability is the players' strategies, which can use randomization and thus generate a distribution over outcomes. A player's strategy must be a best response, under the condition that what the other players are doing in the equilibrium is fixed. In a (k, t)-robust equilibrium, best response has the same meaning as in Nash, but now a player's strategy must be a best response under the condition that he can be part of a deviating group of size k, and must continue to be a a best response even if up to t of the other players are changed arbitrarily. While this requirement is strong, it is analogous to requirements often made in distributed computing that an algorithm be correct no matter what an adversary does.

But other approaches are also possible. For simplicity in the remainder of this discussion, assume that $k = 1$. In the spirit of standard game theory, we could assume that each player has a distribution over which set of t players are strange and what the strange players will do. When combined with the randomization made by strategies, this gives a distribution over outcomes. We could then require that players make a best response to that distribution. Yet another alternative is not to assume a distribution over which set of t players will be strange and what the strange players will do, and to change the notion of "best response" to mean the response that gives the best worst-case outcome (the *maximin* response). That is, we take strategy σ to be at least as good as strategy σ' if the worst-case expected payoff of σ (taken over all possible choices of t strange players and their strategies, while fixing the strategies of the remaining players) is at least as good as that of σ', somewhat in the spirit of [4]. Note that the latter two options do not place as stringent requirements on what it takes to be a best response as (k, t)-robustness.

Aiyer et al. [5] use the latter approach. Effectively, they are modeling Byzantine players as responding to any given candidate rational strategy with a behavior that minimizes the rational players' payoff for that strategy. Under this assumption, Aiyer et al. provide a cooperative backup protocol that is a Nash equilibrium (i.e., a 1-resilient strategy) if the system contains no more than $t < n/3$ Byzantine nodes. Clement et al. [8] and Li et al. [19] discuss other examples of systems that are 1-resilient when rational agents are risk-averse.

These efforts suggest that combining distributed computing and game theory in real systems is not only feasible, but that, given a model of Byzantine behavior, 1-resilience can be achieved for non-trivial applications, such as state-machine replication [5], terminating reliable broadcast [8] and live data streaming [19].

Explicitly modeling Byzantine behavior can also be useful in assessing the impact that introducing Byzantine players has on the efficiency of a system with only selfish players. In particular, Moscibroda, Schmid, and Wattenhofer [21] define the *price of malice* as the cost of tolerating Byzantine players that behave maliciously by comparing the best that selfish players can do in a system where there are no malicious players to the best they can do in the presence of malicious players.

Note that the approach of creating a model of Byzantine behavior does not limit the set of strategies that Byzantine players can adopt; what it *does* limit is the set of Byzantine strategies against which it guarantees k-resilience. If the Byzantine players do play an unanticipated strategy, then the strategy used by the rational players may be painfully far from a best response.

While predicating one's best response on a model of Byzantine behavior does not provide the same level of guarantee as (k, t)-robustness, it can be realized in real systems and it may suffice for many practical applications.

In fact, in a number of situations there are standard forms of "irrational" behavior that a system designer may want to model. For example, in [11, 14], *scrip systems* (systems with virtual money) are considered. In a scrip system, money is a tool to buy services. There is no utility attached to the money itself, only to the services that the money can buy. In a scrip system, if someone needs a service, players can volunteer to provide it. A player gains utility by having the service performed, but must pay a dollar for the service; the player performing the service gets the dollar, but suffers a small utility penalty. Because we assume some discounting (a unit of utility today is worth more than the same unit tomorrow), players will not volunteer to provide service if they feel that they already have enough money. It is shown that there is a rather natural equilibrium in this setting, where all players use a *threshold strategy*: they volunteer to provide the service if they have below a particular threshold of money, and otherwise do not volunteer.

In such a scrip system, two particular types of "irrational" behavior are particularly prominent. In a large scrip system, there will almost certainly be players who hoard money—they continue to volunteer and collect money well beyond their threshold. We might also find the occasional "altruist", who performs services for nothing. (Think of people who post music on KaZaA and Gnutella, or those who seed on BitTorrent.) Rather than trying to protect against arbitrary types of irrational behavior, a system designer using a scrip system may want to focus on dealing with specific types of irrational behavior that are likely to arise. (As shown in [14], hoarders and altruists can have some subtle effects on scrip systems.) This is the analogue of focusing on, say, crash failures or omission failures rather than Byzantine failures when designing a fault-tolerant system, but now the "types of failures" become "types of irrational behavior".

However, provable robustness against Byzantine and selfish behavior is but one of the many properties one may desire from a system. In particular, there is no reason for an equilibrium to be a *good* outcome—one where players get high payoffs. For example, the only guarantee provided by Nash equilibrium is stability; players will typically not deviate from a Nash equilibrium. In practice, the price for achieving an equilibrium may be to limit the freedom to design practical solutions.

For example, k-resilient systems such as BAR-Backup [5], BAR Gossip [19], and Equicast [17] do not allow dynamic membership, require nodes to waste network bandwidth by sending garbage data to balance bandwidth consumption, and provide little flexibility to adapt to changing system conditions.

One option, of course, is to renounce rigorous guarantees, use incentives informally, and argue that rational players are unlikely to deviate from a given protocol—this is the approach used in KaZaA [15] and BitTorrent [9], whose incentive structure has been shown to be vulnerable to subtle exploits [16, 24, 28].

A perhaps more desirable approach is to consider approximate equilibria. Dodis et al. [10] define a notion of *computational Nash equilibrium*, where no polynomially bounded player can gain a non-negligible advantage by not following its strategy. Abraham et al. [1, 2] define ϵ–(k, t)-robust equilibria, where rational players cannot increase their utility by more than ϵ by deviating. Li et al. [20] use the notion of approximate equilibria to design a peer-to-peer application for live data streaming that limits selfish deviations rigorously, but allows the flexibility necessary to address other practical concerns. Their system, which models Byzantine players as malicious, provides a 1-resilient ϵ-equilibrium, where ϵ is now a multiplicative factor, not an additive factor. That is, rational players cannot gain more than a factor of ε by deviating. With $\varepsilon = 0.1$, the system provides virtually jitter-free streaming, supports dynamic membership, and caps upload bandwidth so that the protocol is accessible to users behind cable or ADSL connections.

Considering approximate equilibria seems quite reasonable in practice. It reflects the intuition that if deviating involves developing a new protocol, or going through the headache of installing new software—especially with the risk that new software will be buggy or malicious—it may just not be worth it.

So far we have discussed rational players and "strange" or Byzantine players. But in the fault-tolerant distributed computing, besides "bad" or "Byzantine" players, there are also assumed to be "good" players,

who follow the designer's protocol. It is thanks to the actions of the "good" (non-faulty) processes that fault-tolerant protocols can tolerate the misdeeds of the "bad" processes. Considering good players makes perfect sense in a game-theoretic context as well. People often follow a recommended protocol, even if it is not optimal, as long as it seems reasonable. Most people using BitTorrent continue to follow recommended protocol, seeding even after they have downloaded their desired file(s), even though this is arguably irrational (at least, under some minimal assumptions about utility). Such players are recognized explicitly in the BAR (Byzantine, Altruistic, Rational) model [5], where they are called "altruistic", although they are not, as the altruists in the context of scrip systems, acting to increase well-being; rather, they do what they have been asked to do. They are perhaps better viewed as "obedient" (or, in deference to the acronym, "acquiescent") than altruistic, although there are situations where obedient players can be viewed as acting altruistically.

The idea of altruistic or obedient agents appears in the literature on cooperation, particularly in biology. In variants of prisoner's dilemma, the altruistic/obedient strategy involves punishing players who do not cooperate. Punishing a non-cooperator is typically costly for the punisher; that is, it is not "rational" to punish. But as long as there are obedient players who are willing to punish, then in fact, there is essentially no defection; it becomes rational to cooperate. Thus, both the punishers and the rational players get the same payoff. (See [13, 25, 27] and the references therein for more details.)

In the context of distributed computing, it may well be reasonable for players to be obedient. Even if defecting can give them some small gain in the short run, they are all better off if everyone cooperates. Can better algorithms be designed by taking advantage of the likelihood that a reasonable percentage of players will follow a recommended strategy, provided that it is not too unreasonable? (Of course, we must still make precise what it means for a strategy to be "not too unreasonable".)

Recent work [29] shows that having altruistic/obedient players can in fact help promote cooperation among rational players. But there is clearly scope for more work along these lines.

As we hope this review has made clear, there is much to be gained by taking a game-theoretic perspective to distributed computing problems. Thinking in terms of rational agents opens the door to a wider class of algorithms. It would be well worth trying to understand how specific fault-tolerant algorithms can be modified to deal with rational agents. Moreover, there are a host of new research issues to consider. We consider just three examples here. The first is asynchrony. Game theory implicitly assumes that the world is synchronous. Most strategies proceed in rounds. How do things change if we consider an asynchronous environment? The second issue is taking computational cost into account more seriously. Game theory implicitly assumes that computation is free. Recently, Halpern and Pass [12], continuing a line of research started by Rubinstein [26], introduced a general model of game theory with computational costs. The idea is that players choose Turing machines to play for them, and there is an abstract *complexity* associated each Turing machine M and input x (for example, the complexity of (M, x) could be the time or space used by M on input x). A player's utility can then depend on the complexity of the TM he chooses and the input. This allows us to model how an agent may rationally choose a good heuristic over an exact algorithm that may have much longer running time. Once we add complexity to the picture, we need to consider the strategic aspects. When playing a game against opponents, an agent needs to consider how much time is worth investing in computing a better answer. Combining these ideas with intuitions regarding fault tolerance leads to new insights in security (see [12] for more details). Again, there seems to be scope for much more work to be done. Finally, as we have suggested at various points, altruism and Byzantine behavior can both be seen as instances of having "unexpected" utilities. It would be of interest to consider other broad classes of utility functions that represent behaviors observed in real-life scenarios, and to then try to develop strategies that deal with such behaviors.

Acknowledgments: The work of Alvisi was supported in part by NSF grants CSR-0905625 and CNS-0509338. The work of Halpern was supported in part by NSF grants IIS-0534064, IIS-0812045, and IIS-0911036, and by AFOSR grants FA9550-08-1-0438 and FA9550-09-1-0266, and ARO grant W911NF-09-1-0281. We thank Idit Keidar for her encouragement to write this paper, as well as her helpful comments on an earlier draft.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J. Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proc. 25th ACM Symposium on Principles of Distributed Computing*, pages 53–62, 2006.

[2] I. Abraham, D. Dolev, and J. Y. Halpern. Lower bounds on implementing robust and resilient mediators. In *Fifth Theory of Cryptography Conference*, pages 302–319, 2008.

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

[4] M. Aghassi and D. Bertsimas, Robust game theory. *Mathematical Programming, Series B*, 107(1–2):231–273, 2006.

[5] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J. P. Martin, and C. Porth. BAR fault tolerance for cooperative services. In *Proc. 20th ACM Symposium on Operating Systems Principles (SOSP 2005)*, pages 45–58, 2005.

[6] R. J. Aumann. Acceptable points in general cooperative n-person games. In A. W. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV, Annals of Mathematical Studies 40*, pages 287–324. Princeton University Press, Princeton, N. J., 1959.

[7] A. Clement, J. Napper, E. Wong, H. Li, J. P. Martin, L. Alvisi, and M. Dahlin. Theory of BAR Games. Department of Computer Sciences, The University of Texas at Austin. Report# TR-06-63, 2006.

[8] A. Clement, H. Li, J. Napper, J. P. Martin, L. Alvisi, and M. Dahlin. BAR Primer. In *Proc. Int. Conference on Dependable Systems and Networks (DSN 2008), DCCS Symposium*, pages 287–296, 2008.

[9] B. Cohen. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[10] Y. Dodis, S. Halevi, and T. Rabin. A Cryptographic Solution to a Game Theoretic Problem. In *Advances in Cryptology - Crypto 2000*. LNCS vol. 1880, Pages 112-130, 2000.

[11] E. J. Friedman, J. Y. Halpern, and I. Kash, Efficiency and Nash equilibria in a scrip system for P2P networks, In *Proc. Seventh ACM Conference on Electronic Commerce*, pages 140–149, 2006.

[12] J. Y. Halpern and R. Pass. Game theory with costly computation. In *Proc. First Symposium on Innovations in Computer Science*, 2010.

[13] C. Hauert, A. Traulsen, H. Brandt, M. Nowak, and K. Sigmund. Via freedom to coercion: the emergence of costly punishment. *Science*, 316(5833):1905–1907, 2007.

[14] I. Kash, E. J. Friedman, and J. Y. Halpern. Optimizing scrip systems: efficiency, crashes, hoarders, and altruists. In *Proc. Eighth ACM Conference on Electronic Commerce*, pages 305–315, 2007.

[15] Kazaa. http://www.kazaa.com.

[16] Kazaa Lite. http://en.wikipedia.org/wiki/Kazaa.Lite.

[17] I. Keidar, R. Melamed, and A. Orda. Equicast: Scalable multicast with selfish users. In *Proc. 25th ACM Symposium on Principles of Distributed Computing*, pages 63–71, 2006.

[18] L. Lamport, R. Shostak and M. Pease. The Byzantine Generals Problem *ACM Trans. Program. Lang. Syst.*, 4(3):382-401, 1982.

[19] H. Li, A. Clement, E. L. Wong, J. Napper, L. Alvisi, and M. Dahlin. BAR Gossip.. In *Proc. 7th Symposium on Operating System Design and Implementation (OSDI 2006)*, pages 191–204, 2006.

[20] H. Li, A. Clement, M. Marchetti, E. Kapritsos, L. Robison, L. Alvisi, and M. Dahlin. Flightpath: obedience vs. choice in cooperative services. In *Proc. 8th Symposium on Operating System Design and Implementation (OSDI 2008)*, pages 355–368, 2008.

[21] T. Moscibroda, S. Schmid, and R. Wattenhofer. When selfish meets evil: Byzantine players in a virus inoculation game. In *Proc. 25th ACM Symposium on Principles of Distributed Computing*, pages 35–44, 2006.

[22] J. Nash. Equilibrium points in n-person games. *Proc. National Academy of Sciences*, 36:48–49, 1950.

[23] M. J .Osborne and A. Rubinstein. A course in game theory. MIT Press, Cambridge Mass., 1994.

[24] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent? *Proc. of 4th Symposium on Networked Systems Design and Implemententation (NSDI 2007)*, pages 1–14, 2007.

[25] D.J.F. de Quervain, U. Fischbacher, V. Treyer, M. Schellhammer, U. Schnyder, A. Buck, and E. Fehr. The neural basis of altruistic punishment. *Science*, 305:1254–1258, 2004.

[26] A. Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, 39:83–96, 1986.

[27] K. Sigmund. Punish or perish? Retaliation and collaboration among humans. 22(11):593–600, 2007.

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

[29] E. L. Wong, J. Leners, and L. Alvisi. It's on me! The benefit of altruism in BAR environments. In *Proc. of 24th International Symposium on Distributed Computing (DISC 2010)*, pages 406–420, 2010.

[30] E. L. Wong, I. Levi, L. Alvisi, A. Clement, and M. Dahlin. Regret-freedom is not free. Under submission, 2011.