Experiences Applying Game Theory to System Design

Ratul Mahajan Maya Rodrig David Wetherall John Zahorjan

University of Washington

ABSTRACT

We applied techniques from game theory to help formulate and analyze solutions to two systems problems: discouraging selfishness in multi-hop wireless networks and enabling cooperation among ISPs in the Internet. It proved difficult to do so. This paper reports on our experiences and explains the issues that we encountered. It describes the ways in which the straightforward use of results from traditional game theory did not fit well with the requirements of our problems. It also identifies an important characteristic of the solutions we did eventually adopt that distinguishes them from those available using game theoretic approaches. We hope that this discussion will help to highlight formulations of game theory which are well-suited for problems involving computer systems.

Categories and Subject Descriptors

C.2.m [Computer-Communication Networks]: Miscellaneous

General Terms

Design, Economics

Keywords

Game Theory, Incentives, System Design, Wireless Networks, Interdomain Routing

1. INTRODUCTION

Autonomous agents with varied interests characterize many computer systems today. Game theory, a branch of economics that deals with strategic and rational behavior [15], appears to be a natural tool for both designing and analyzing the interactions among such agents. Consequently, there has been much recent interest in applying game theory to systems problems, with many successes reported (e.g., see references [20, 3, 18, 9]).

Encouraged by the potential this approach, we tried to use game theory to help formulate and analyze solutions to two systems problems of interest to us. The first problem was to induce autonomous nodes in a multi-hop wireless network to forward packets for each other [11]. The second problem was to improve the routing paths used by Internet Service Providers (ISPs) by designing mechanisms that enable inter-ISP coordination [12]. Both problems involve interacting autonomous entities, and have other characteristics that invite application of game theoretic approaches. For instance, the first problem exhibits the classical group versus individual rationality tension: nodes need to forward packets for the network to be connected, but an individual node decreases its energy and throughput by doing so. Indeed, versions of both have been studied by other researchers using these techniques (e.g, see references for the packets forwarding problem [21, 23, 6, 5, 16, 8, 22] and the ISP routing problem [9]).

However, for the specific issues we wished to address in each problem, we found a straightforward application of game theory to be difficult, and eventually turned to less formal approaches to construct and analyze solutions. In the body of the paper, we discuss in detail the issues we encountered. Briefly, supporting the inherent asymmetry in node workloads and topological placement proved hard in the wireless network case, and simultaneously supporting flexible objectives and incentive compatibility proved hard in the ISP routing case.

Despite these initial difficulties, we remain optimistic about the long-term benefits of applying game theory to system design. We hope that the discussion of our experiences will prove useful to systems researchers using game theory in their designs, by alerting them to potential stumbling blocks, and to game theorists, by highlighting the kind of formulations that are needed by systems applications. Towards the latter goal, we also identify a common characteristic of our solutions to both the problems.

The rest of this paper is organized as follows. In Section 2, we describe our two systems problems and our solutions to them. In Sections 3 and 4 we discuss the issues that we faced while applying game theory to these problems, dividing them along the lines of model formulation and solution implementation. We conclude with a retrospective reflection on our experiences in Section 5.

Authors' e-mails: {ratul,rodrig,djw,zahorjan}@cs.washington.edu

SIGCOMM'04 Workshops, Aug. 30+Sept. 3, 2004, Portland, Oregon, USA.

2. CASE STUDIES

In this section, we provide an overview of the two systems problems that we tried to solve using concepts from game theory, along with our resulting designs. Further detail can be found in the corresponding technical reports [11, 12]. We defer the discussion of our experiences to Sections 3 and 4.

2.1 Multi-Hop Wireless Networks

The nodes of emerging multi-hop wireless networks, such as community meshes [1, 2], may belong to different users. When the source and the destination nodes for a packet are not within direct transmission range of each other, they must rely on intermediate nodes to forward packets between them. While packet forwarding improves connectivity in the network, benefiting all nodes in the long-run, it is not individually rational because of the cost to the forwarder in terms of energy and bandwidth. Selfishness thus incents nodes to *cheat* by avoiding packet forwarding. The basic problem is to moderate this selfish behavior in a way that preserves network connectivity.

Relevant Prior Work Earlier work on this problem can be classified into three categories. The first category is currency-based in which nodes that forward packets are paid by the senders or recipients of the packets. The amount of payment can be either fixed by design [6, 23] or vary based on market forces [16, 8]. The second category uses reputation to incent nodes. Nodes acquire reputation based on their forwarding behavior and other nodes decide whether to forward packets for a given node based on its reputation. Examples of this approach include CONFIDANT [5] and Pathrater [13]. The third category models forwarding as a strategic game and, under certain formal assumptions, derives a forwarding rate for nodes that is a Nash equilibrium. Generous Tit-for-Tat [21] is an example of this approach. More generally, Urpi *et al.* [22] present several results characterizing enforceable policies in a setting where each node's utility function includes both bandwidth and energy terms.

Refined Problem Statement The particular version of the packet forwarding problem we wanted to solve differed from previous work in several ways in terms of the properties of the solution. The primary one was that there should be no artificial restrictions on how many packets a node can send. Wireless networks are expected to be highly heterogeneous in terms of workload and placement in the network topology: some nodes will want to send more data, some less; some will receive many forwarding requests, others perhaps none. Earlier work explicitly or implicitly tried to achieve fairness: nodes should not have to forward (many) more packets for others than they send. In contrast, we wanted to ensure that even nodes that are not well placed to forward packets be able to use the network without significant limitations. Basically, our goal was to induce behavior that results in network connectivity as equivalent as possible to that enjoyed in a fully cooperative network. As discussed in Section 3.1, this is not a natural fit for game theoretic approaches.

The other desirable properties of the solution were the following. First, it should be very hard, if not impossible, for a node to cheat without incurring a significant penalty. Second, the more egregious the cheating, the faster the node should suffer its consequences, which guards against nodes that cheat and stay in the network for a short time. Third, there should be minimal startup cost for nodes entering the network, eschewing the kind of re-entry penalties employed by some earlier approaches. Finally, the implementation overheads of the solution should be low.

Our Solution Initially, we hoped to pose the problem as a mechanism design exercise and find a provably strategy-proof solution. However, eventually the problems described in subsequent sections motivated us to abandon even the attempt to formally model the problem, and we used an informal approach to finding and validating a solution. Our protocol, Catch, assumes a backdrop of cooperation and leverages it to detect and punish cheating [11]. Catch uses anonymous messages, where the identity of the sender is hidden, to discover the true network connectivity even though a cheating node may try to hide links to reduce its forwarding obligations. The insight here is that the cheating node would want to be connected to at least one other node in the network, and because it cannot infer the sender of an anonymous message, it is forced to acknowledge connectivity to all of its neighbors. Nodes also implements a kind of watchdog [13], monitoring the behavior of their neighbors to verify that they correctly forward packets. If cheating is detected, all neighbors of the cheater (identified in the topology discovery step) are notified. Each then isolates the cheater, which effectively cuts off its network connectivity. Thus, we use the fear of being disconnected as a disincentive against cheating. The design of Catch reflects tradeoffs that make its implementation possible and effective in the heterogeneous settings we wanted to address, while in turn sacrificing the absolute guarantee that there could be no situations under which a node's selfish benefit might be maximized by operating in violation of the desired social goal.

2.2 ISP Route Negotiation

Our second problem concerns routing between ISPs in the backbone of the Internet. ISPs are competing, autonomous entities, but they must cooperate by delivering packets to each other so that the packets are able to reach their ultimate destinations. Currently, ISPs make unilateral decisions about routing, including which peering[1] link is used to send a packet to the downstream ISP. Unsurprisingly, the ISPs' routing decisions are driven by self-interest, and often do not take global consequences into account. For example, a prevalent policy is "early exit," where the upstream ISP routes a packet through the peering link that is nearest to the packet's source, even though this may lengthen the full path the packet must take relative to other available choices. The end result is that Internet backbone routing paths can be sub-optimal and even unstable, with the choices made unilaterally by one ISP impacting another. Anecdotally, operators spend much time fine-tuning their routing choices given the indirect impact of routing decisions made by their neighbors. The problem, then, is to improve upon this situation by designing mechanisms that enable ISPs to coordinate their routing decisions.

Relevant Prior Work Feigenbaum *et al.* formulate the problem of computing globally shortest paths, with respect to the cost of carrying the packet, in the backbone of the Internet as a mechanism design exercise [9]. They design a direct mechanism to achieve this goal and prove that it is strategy-proof.

[1]Interconnections between ISPs are called peering links. Large ISPs usually have multiple peering links between them.

Refined Problem Statement One important change we made to the problem statement was to modify the goal of achieving a globally optimal set of paths. This was motivated by the observation that compared to the current situation a globally optimal solution is often disadvantageous to some of the ISPs – they suffer a penalty under it relative to the decisions they would make unilaterally. Thus, a mechanism that computes globally optimal solution is not individually rational, and some ISPs would be unwilling to adopt it without introducing side payments which we viewed as an unwanted complication. Rather than global optimality, we wanted to ensure individual rationality, at least when both ISPs so desire.

Additionally, Feigenbaum *et al.*'s mechanism is direct: the ISPs exchange accurate cost information about packet delivery within their own networks. Competitive concerns, however, argue that this may not be acceptable to the ISPs (see Section 3.2.1). We therefore wanted to avoid direct mechanisms as part of our solution.

In more detail, we wanted a negotiation protocol that honored four key constraints, mentioned here in a prioritized order. First, competitive concerns dictate that ISPs should not be required to disclose detailed information about their networks, such as the true cost of carrying traffic between two end points. Second, the protocol should support ISPs with different objective functions, as opposed to a fixed function of delay and utilization, because ISPs optimize their networks for different criteria. Third, the protocol should be flexible in the nature of outcomes it can compute [7]. Different pairs of ISPs have different relationships such as customer-provider and peer-peer (which involves no payment in either direction), as a result differing willingness to accommodate each other. For instance, peers might want a fair solution for some definition of fairness, while a provider ISP might be content with any solution in which it does not lose compared to the default routing. Fourth, the protocol should be strategy-proof – a cheating ISP should not be able to manipulate the solution in its favor.

Our Solution We began using traditional negotiation mechanisms [4, 17] to analyze and solve our problem, but for reasons discussed in Section 3.2 found it difficult to apply the existing theory. Instead of being provably optimal by some criterion, our solution is simply a negotiation framework that the ISPs can use so that both see improvement relative to the default of unilaterally decided paths [12]. In this framework, using their own optimization criteria, ISPs map their preference for a peering point for a flow to an opaque, cardinal utility. It is opaque because by just observing this utility value, the other ISP can infer neither the optimization criteria nor the value of the metric (e.g., number of milliseconds for latency). ISPs exchange their lists of preferences for each peering point for each flow. They then engage in rounds of proposals, each involving a suggested flow and peering point choice made by one of them. The other ISP either accepts or rejects each such proposal. This continues until all flows have been negotiated or one of the ISPs wants to stop. The criteria for picking the proposing ISP, the flow to propose, and acceptance or rejection of the proposal determine the nature of the final outcome. All these decisions are left to the individual ISPs to make as they see fit.

This structure satisfies the first three design goals mentioned above. The ISPs do not need to disclose transparent measures of latency or cost and are free to use any optimization criteria. The objective of the negotiation, which would likely be decided in advance by the negotiating ISPs depending on their relationship, is left to the ISPs rather than being imposed by an external mechanism. Various objectives can be met within the same framework. On the other hand, this flexibility comes at a cost: our solution is not strategy-proof or incentive compatible in the game theoretic sense. Instead we rely on the fear of getting caught and social pressure to deter ISPs from cheating (by being dishonest about their utilities).[2] In any case, an ISP can always choose to not negotiate if it loses by doing so compared to unilateral routing.

3. FORMULATION ISSUES

For both problems, we were hoping that game theory would help us with one or more of the following. First, it would help us to realistically model the problem, leading to a better understanding of the issues involved. In this effort, game theory did help us to some extent, and certain aspects of our eventual solutions resulted from this exercise. Second, we hoped that a good solution would become apparent from the model above, for instance by extending one of the existing results in the theory. This turned out to be overly optimistic; our models were too complicated because of several real-world issues. Third, we hoped that game theory would enable us to analyze our solutions, so that we could show they fulfill certain desired properties. This again, proved challenging.

We have divided the issues we faced into two broad groups. In this section, we discuss the basic problems encountered while formulating our two problems in game theoretic terms. In the next section, we discuss problems related to anticipated implementations of game theoretic solutions.

3.1 Multi-Hop Wireless Networks

While addressing the packet forwarding problem, we first assumed that all nodes were selfish and explored mechanisms to incent such nodes to forward packets. Sections 3.1.1-3.1.3 describe the techniques we considered and why we did not adopt them. We then focused on preserving cooperation in a largely cooperative setting and wanted to use to game theory to prove certain robustness properties of our system. Section 3.1.4 explains the hurdles we faced in doing so.

3.1.1 Barter

The first incentive technique we considered was based on barter. A node would forward a packet for another node only if that node returned the favor by forwarding for the first node. Such a scheme is attractive because of its simplicity and low implementation cost compared to that of virtual currency (Section 4.2).

But it turned out to be hard to build a real system on a bartering primitive. This is because barter is unable to satisfactorily deal with asymmetry in time and space, which leads to undesirable sys-

[2] Raiffa notes that when parties seek joint gains over a default contract, as is the case in our setting, "there is a great deal of incentive for each party to act honestly and non-strategically." [17].

tem behavior.[3] Asymmetry in time means that two nodes might not simultaneously require each other's forwarding services, which can cause an unpredictable delay in the time a packet takes to travel through the network. Asymmetry in space is worse. It means that while one node may require the services of a second node, the second node may not require the first node for forwarding. Such a situation can arise due to topological asymmetry in the position of the two nodes. Multi-way transactions – where more than two nodes forward for each other – can mitigate this problem to some extent, but not entirely if some nodes, such as those on the periphery of the network, are inherently not in a position to forward for others. This goes against our goal of letting all nodes, regardless of their position in the network, send their packets.

3.1.2 Currency

To overcome the shortcomings of a barter based system, we considered the use of virtual currency. Currency is useful when an action and its reward are not simultaneous. This is true for multi-hop wireless networks: the action is packet forwarding and the reward is being able to send your own packet. Currency provides an intuitive way to model the wireless network problem. Many researchers have advocated its use to encourage nodes to forward packets for others [8, 23, 16, 6], and we considered using it as well.

However, from an overall system design perspective, we encountered two hurdles. Both of these arise from the mismatch between the rate at which currency is acquired and the workload of the node. The first hurdle results from the assumption that every node in the system is capable of earning sufficient currency to accomplish its goals in a timely manner. In our packet forwarding problem, nodes at the periphery of the network may not receive sufficient (or any) forwarding requests, rendering them too poor to be able to send their own packets. This situation is acceptable if the system goal is some kind of fairness between sending and forwarding, but not if we want to ensure that all nodes be able to send their packets. A potential solution to this problem is to periodically give some currency to all nodes. This mitigates but does not solve the problem: it allows the poor nodes to send some packets but places an artificial and arbitrary limit on the rate at which such nodes can do so even when the network might be capable of carrying more.

The second hurdle confronted in using currency is the opposite side of the same coin. Consider a situation in which there is an asymmetry in the rate at which currency is consumed and earned by a node. If the earning rate is higher than the consumption rate, as it might be for a particularly well-placed node, that node will start accumulating currency. At some point, it may no longer be interested in earning more, at least until it can use those already acquired. Such behavior is counter to the traditional economic assumption of non-satiation but is possible in a wireless network. Unless the virtual currency is tied to something of value in the physical world, a node has no incentive to accumulate an infinite amount of it. A satiated node will temporarily stop contributing to the system, effectively resulting in periods of disconnection for the other nodes that rely on it. The problem can be addressed by rebalancing the excess accumulated currency [23, 6], but this simply changes the timescales over which the problem occurs. Once again, the problem is only mitigated, not solved.

3.1.3 Strategic Forwarding

Another potential way to address our wireless network problem is to set up the packet forwarding decision as a game in which forwarding at a certain rate is the Nash equilibrium. The intuition behind this approach is that if a node cheats by forwarding at a lower rate, other nodes would observe a lower forwarding rate for their packets. These nodes would then reduce their own forwarding rate, which in turn would hurt the cheating node. Such an approach has been proposed by Srinivasan *et al.* [21] and Urpi *et al.* [22].

The key problem with this approach is that it assumes symmetric workloads, with all nodes directly or indirectly depending on each other. This may not be always true. Again, consider a peripheral node that requires a few other nodes for packet forwarding, but no other node in the network requires this node. Other nodes in the network would be able to drop packets from this node without any negative consequences. As before, this goes against our goal of enabling all nodes to use the network.

3.1.4 Evolutionary Approach

So far we assumed that all nodes are inherently selfish and tried to induce cooperative behavior in this setting. The subsections above describe the difficulties we faced in doing so.[4] These observations prompted us to modify our goal from *inducing* cooperative behavior to *preserving* cooperative behavior. That is, we wanted to design a system that stays cooperative if it starts with cooperative nodes. Cooperative nodes forwards all the packets they receive for forwarding.

With the above goal in mind we designed Catch for a largely cooperative environment. Next, we wanted to use techniques from game theory to prove that cooperation was an evolutionarily stable strategy (ESS) under Catch. Evolutionary approaches usually proceed as follows. First, consider a set of strategies a player might adopt, such as cooperate or cheat. Second, define a single interaction round of the strategic game using a payoff matrix. Each player's payoff depends on her strategy and that of the players she interacts with in that round. Third, define how players mutate strategies across rounds, typically assuming that they try to maximize their payoffs by tending towards those strategies with highest observed return in the previous round. Fourth, show that the system evolves to a point where most nodes have adopted a socially desirable strategy, or at least that such a situation is equilibrium stable.

Our attempts to use evolutionary games to show robustness against cheating in Catch were largely unsuccessful. Previous work of this sort considered pairwise interactions of nodes, a notion linked to the social goal of achieving some concept of balance between the packets forwarded by those nodes for each other. Recall, however, that our goal was that all packets should be forwarded, because of

[3]The shortcomings of a bartering economy are fairly well-known. We describe them here to highlight that they carry over to system building too.

[4]We do not discuss reputation-based systems in this section. The issues we faced with a straightforward application of such systems, such as the requirement of global, persistent identities, were implementation related, and we briefly discuss them in Section 4.

the realization that some poorly placed nodes would otherwise be unable to communicate. This implies that a node's utility function must assign effectively zero cost to forwarding, since it must never be profitable for a node to refuse to forward, irrespective of its own rate of packet transmission. Thus, utilities measure positive outcomes, such as packets delivered by the network for that node, and consequently a punishment must be in the form of a reduction in delivered packets. In order to reduce the number, all neighbors of a node must agree to do so, because otherwise the offending node could simply forward all traffic through any willing node, escaping punishment.

It is difficult to map the above interaction directly to rounds of an evolutionary game. The payoff matrix must account for interactions between a node and all of its neighbors. It must also account for uncertainty in the occurrence and observation of certain events such as packet losses (Section 4.3). At this detailed level, the number of possible strategies to escape detection or prevent consensus building for punishment is overwhelming, as a strategy could depend on the topology, workload, routing behavior of the network, and even falsely accusing a neighbor of misbehavior. We do not know how to build a model that is general enough to capture this level of detail and still be amenable to analysis. One workable simplification is to simply assume that the detection of cheating is reliable to some fixed degree, independent of the particular cheating strategy, and that enforcement is similarly reliable. But that abstracts away exactly the problem we had hoped a game theoretic analysis would help with: showing that no (detailed) strategy would defeat our objective.

From a system design perspective, the implication of the inability to consider all possible strategies is similar to that faced by security practitioners. One cannot prove that the system is absolutely secure, but can only "raise the bar" for attackers by showing that the system can resist certain well-known attacks.

3.2 ISP Route Negotiation

The key challenges in obtaining a strategy-proof negotiation protocol were competitive concerns, which limited us to using opaque utilities, and the presence of many side channels, which limited us to incentive compatible mechanisms. Incentive compatibility in turn conflicted with the desire for flexibility.

3.2.1 Direct Mechanisms vs. Competitive Concerns

We first considered strategy-proof, direct mechanisms in which ISPs revealed complete and accurate information about their utility functions, that is, the true costs for using specific routes as the basis for route selection. This approach was taken in earlier work on lowest-cost BGP routing [9].

However, direct mechanisms are problematic given real-world competitive concerns: ISPs are unlikely to reveal their costs for carrying packets to their potential competitors and may value this secrecy more than any potential routing improvements. Knowing an ISP's operational cost and major sources of revenue (e.g., carrying traffic between San Francisco and Melbourne) enables a competing ISP to pry away customers and plan its own network to steal a share of the other ISP's profits. While this information cannot be used by competing ISPs to benefit themselves in the game in the short-term, it can certainly be used by them outside the game in the long-term.

More generally, much incentive compatible mechanism design focuses on eliciting truth by engineering situations in which truth-telling is the best strategy because: i) a player can only hurt itself by not doing so; and ii) other players cannot unfairly profit by knowing this player's information. However, the latter proof is misleading when information revealed as part of the game can be used outside of the game by other players to benefit themselves or hurt their opponents.

To address the concern above, we turned to the exchange of opaque utilities, chosen arbitrarily by the ISPs, rather than true costs. Because the ISPs are free to compute these utilities as they see fit, they can control the amount of information they might leak, with no externally imposed requirements. With that change we hoped to make use of mechanism design, stating the desired objective of the negotiation in terms of these utilities.[5] The next two subsections describe the problems we encountered in doing so.

3.2.2 Private Information vs. Side Channels

By definition, game theoretic solutions account only for information that is modeled within the game. However, it can be difficult to capture all of the key factors in straightforward models, resulting in a solution that is vulnerable to extra-game issues. For our ISP negotiation problem, we confronted this problem when attempting to formulate a game in which the ISPs were unaware of each other's utility functions.

In our initial approach to this problem, we considered a construction where two ISPs exchange preference lists at the same time. Our hope was that it will be easier to design a strategy-proof mechanism when neither ISP can bias the game by choosing its preferences in light of those of the other.

However, while the assumption that some information is private is an attractive way to define the game, and may be justifiable when considering only a single instance of it, it misrepresents the actual situation in ways that undermine the results. For instance, ISP negotiation is an ongoing process, and so history is an important channel that leaks utility information: an ISP's utility for a given path is not likely to change drastically with time, and so past actions are likely to be good predictors of future behavior. While this invites modeling the situation as a repeated game, exactly what can be gleaned from past actions and other side channels is likely to be situation dependent and hard to characterize. For instance, it may be possible be glean some information about an ISP utility for a given path by measuring the latency of packets along that path. We therefore felt it was ill advised to model the problem this way, since potential inaccuracies in the assumptions about what information could be indirectly inferred would make the results suspect.

Our expectation is that versions of the situation above arise in many systems applications, as information about the other player's utility might be inferred through side channels. A game formulation that involves players having limiting knowledge of each others' private

[5]Note that the utilities of two ISPs are not directly comparable, which means that certain social goals, such as fairness measured using true costs, are not achievable.

information would rely on unverifiable assumptions about the nature of this information.

3.2.3 *Fixed vs. Flexible Objectives*

In reaction to the above observation, we wanted a mechanism that an ISP cannot bias in its favor even if it had complete information about the other's utilities. Thus, we wanted an incentive compatible mechanism in which an ISP's best strategy is to truthfully reveal its utilities irrespective of the other ISP's utilities.

However, we could not design a strategy-proof mechanism because we wanted to leave maximum flexibility to the ISPs in determining the outcomes they wanted to achieve. Allowing them to advertise (essentially) arbitrary utilities was one aspect of that. Defining the goal of the negotiation was a second. It was not clear how to express "any outcome mutually agreeable and beneficial to the negotiators" as the objective of the mechanism. Such problems will arise in other systems where the designer does not want to enforce a particular objective, but wants to leave it open to the interaction between the players [7].

In fact, it is known that in some cases it is impossible to reconcile flexibility and incentive compatibility. Myerson and Satterthwaite proved that for the simple negotiating scenario of a single seller, an object, and a buyer, in the absence of a disinterested third party acting as a subsidizer, appraiser or arbitrator no mechanism exists that is both incentive compatible and able to implement all feasible solutions in which both seller and buyer profit [14]. Many systems do not have a natural third-party. We believe that a similar impossibility result would hold for them, and the designers would then be forced to choose between flexibility and incentive compatibility.

4. SYSTEM IMPLEMENTATION ISSUES

Like any modeling tool, game theory involves abstraction. Part of its power is that a set of cleanly defined concepts can be used to reason about complicated systems, reaching conclusions that might otherwise be obscured by the details. However, our ultimate goal was to build workable implementations of solutions to our two systems problems. We encountered several difficulties translating from the abstract solutions to real implementations. Unlike the problems in the last section, these problems are not fundamental and can likely be resolved with some amount of effort. But taken together they tend to make the game theoretic models complex or costly, which reduces their value in practice. Most of the issues that we discuss here are relevant to the wireless network problem.

4.1 Identity

Many game theoretic formulations assume that players have persistent, global identities, perhaps because of its roots in social interaction. For example, in the packet forwarding problem, reputation based systems such as CONFIDANT [5] require global identities to be effective: the idea of reputation implies that multiple players contribute to the assessment a of node, implying that they share an understanding of the node they are talking about (and hence that a global identity exists for that node). Similarly, currency based systems, such as Sprite [23], require strong identities because the currency must be unforgeable. This in turn ultimately requires a trusted mechanism that vouches that some nameable node indeed legitimately possesses the offered currency.

While many game theoretic approaches require persistent, global identities, in many computer systems agents may have no identity, or an identity whose meaningfulness is limited in time and space. For example, nodes in our wireless network do not have a reliable global identity. (Strong identities are present in the ISP negotiation scenario, though.) The closest thing to an identity they possess is the hardware address of their wireless interface, which is not global because nodes not in direct transmission range cannot verify each other. In fact, it is also not persistent or unique if nodes can easily send packets using arbitrary hardware addresses. As another example, Lai *et al.* have looked at the problem of constructing reputation-based systems for a peer-to-peer file sharing application where identities are globally unique but users can freely acquire a new identity [10]. One way to interpret their results is that, generally speaking, the situation is hopeless.

This mismatch between the assumptions of the theoretical framework and what is realistically feasible in actual systems is a significant stumbling block to deploying solutions derived from game theoretic approaches. As one example, in our wireless network the additional cost of implementing strong identities, for instance using public keys, made some incentive-based solutions unattractive.

4.2 Cost-Effectiveness

In many systems getting the players to cooperate with each other is not the only goal. In fact, it may not even be a key goal. For instance, in a multi-hop wireless network, functions such as routing, power management and media access would probably rank higher than inducing cooperation. Thus, the cost of cooperation inducing mechanisms must be low and commensurate with its benefits.

The identity issue in Section 4.1 can be thought of as an example of this issue. While one could imagine the deployment of, say, a network-wide public key infrastructure (PKI) to provide identities and largely solve the problem, the cost to do so would vastly outweigh the benefit.

As another example, consider a currency-based solution to packet forwarding problem. A straightforward implementation of virtual currency requires i) a trusted, central authority, such as an almost-always-accessible software service that enables virtual currency or per-node, tamper-proof hardware for correct accounting; ii) an implementation of strong identities; and iii) the insertion of accounting information into every packet, which represents added throughput and energy cost. Additionally, if the price of carrying packets changes dynamically, additional mechanisms for estimating and disseminating pricing information are required.

Contrast the above with the implementation cost of Catch. It does not require any trusted central authority or global identities. Local, but persistent, identity such as the hardware address is sufficient. No extra information is added to data packets, and the bandwidth overhead of control packets is negligible [11]. Finally, all information sharing among nodes is limited to two-hop neighborhoods.

4.3 Uncertainty

In real systems, the outcome of an event of interest can sometimes be uncertain, complicating the application of game theory to such situations. Below we describe two examples from the packet forwarding problem and a potential issue with ISP negotiation.

First, transmissions in a wireless network have a non-zero error rate, and so a packet may have to be transmitted multiple times before it correctly reaches the receiver. Whether a packet was correctly received is known only to the receiver; the sender has to take the receiver's word for it. The number of retransmissions, which represents the forwarding cost for the sender, depends on the error rate and is hard to accurately predict in advance. This unpredictability in the forwarding cost complicates the use of incentives because the degree of incentive expected by a player depends on the incurred cost. A number of authors have discussed this problem [6, 23].

Second, we used a detect-and-punish strategy to discourage cheating, and a tit-for-tat mechanism to discourage false implications: a node punishes the implicating node when it is wrongly accused of cheating. But due to some fundamental limitations of the wireless domain, our detection mechanism is not foolproof – it can incorrectly conclude that an honest node is cheating. This uncertainty does not play well with tit-for-tat.

Complications concerning uncertainty are not limited to the wireless problem. Our discussion is limited to it because we have not yet looked at implementations of the negotiated settlement among ISPs, but we can foresee at least one potential problem. After the negotiation ends, ISPs are expected to implement the negotiated paths. Assume that one of the ISPs does not implement a negotiated path that was burdensome for it. In this scenario, the second ISP would be uncertain whether this happened because the first ISP is cheating or because there was a failure inside the first ISP's network after the negotiation ended.

Modeling these situations was challenging because of the simultaneous presence of various factors. First, a formulation using trembling hand equilibria – a common way to account for uncertainty – may not be enough by itself because we are not (exclusively) concerned with players' mistakes. Second, the frequency of occurrence of uncertain events such as packet losses in the wireless network or failures in an ISP network is not known in advance and can vary widely. Third, information about certain events can be asymmetric (known only to some of the players) which opens up gaming possibilities. For instance, whether a packet was lost in transmission is known only to the receiver.

As an aside, we deal with uncertainty in Catch using statistical tests. To avoid penalizing honest nodes that might appear to be cheating, due to uncertainty in packet losses, the tests are designed to be liberal. As a result, Catch may not punish nodes that cheat very little. But egregious cheaters are always detected reliably.

5. CONCLUDING REMARKS

The multi-hop wireless networks ISP negotiation problems are our first attempt to apply game theory to systems problems. When we began our work, we hoped that game theory would help us better understand the problems, suggest solutions, and aid in analyzing the properties of the solutions we arrived at. Game theory did help us to some extent and certain aspects of our solutions are derived from common game theoretic concepts. But for most part, we have not yet succeeded. and the previous two sections discuss several contributing factors. Despite these difficulties, we remain optimistic that the application of game theory to systems problems will be of great benefit in the future. We hope that identification of these issues will lead to formulations of the theory that place more emphasis on systems factors such as highly skewed workloads and abilities of players, uncertainty, implementation cost, competitive concerns and so forth. We hope that our experience will also be useful to systems designers looking to use game theory as part of their design.

We conclude this paper with a higher level reflection on our experiences and their implications. Perhaps we made a fundamental mistake in approach. We had in mind a set of goals and set out to build something to achieve them. We also had in mind that game theory would help with this in a constructive sense, ideally leading us to clever solutions, and at least providing prior results on which to build. In retrospect, however, it seems more likely that our original goals were provably unrealizable than buildable, and we should have looked to game theory to show that, and (if indeed necessary) to navigate towards achievable ones. For instance, our ISP routing problem required a negotiation that revealed no information, arrived at a provably optimal result, and was efficiently computable. Our packet forwarding problem required a strategy-proof mechanism that would induce a node to forward unboundedly more packets than it itself sent, and to achieve this without a node trusting anything but itself.

A lesson for systems builders that is not novel, but our experiences indicate bears repeating, is that formal techniques may be as valuable in eliminating design goals as in achieving them. A lesson for theorists is that this idea is usually counter-intuitive to systems builders. New results, or the application of existing ones in a systems context, succinctly identifying the impossible would be significant contributions. We appreciate that it may be more difficult to prove these "lower bound" results than constructive ones.

In the end, we did construct solutions to both our problems, using a somewhat ad hoc approach. It is instructive to consider how this was possible, given that we were thwarted in applying game theory to them. Fundamentally, system building involves tradeoffs. Researchers may disagree on the evaluation of a tradeoff, but there is little quibbling about the process. This allows us to construct systems that, in some sense, are flawed in every dimension – they do not attain some pure goal in any of them, but overall the flaws are outweighed by what those systems manage to achieve.

What were the tradeoffs in our systems, not likely sensibly arrived at by a game theoretic approach? Surprisingly, the common high-level feature of both solutions is that they take a mostly cooperative behavior as given, rather than trying to induce it using external mechanisms. For instance, in Catch, we expect a node to isolate a cheating neighbor even when that neighbor is correctly forwarding the node's own traffic. We assume a priori that the neighbors of a cheating node are cooperative and will punish it, rather than arriving at this result via game theory. This in turn results in a solution with low implementation cost. Similarly, the negotiation procedure assumes that ISPs will mostly cooperate in the interest of maintaining good, long-term, business relationships, even when cheating by lying about their utilities might benefit them in the short-term.

A key question looking forward is how to best apply game theoretic analyses and models in this situation. Traditionally, non-cooperative game theoretic formulations assume that all players are both selfish and rational, that is, that their utility functions represent their private gain and that they operate in ways that serve to maximize those functions. Introducing mechanisms that moderate the selfish behavior to achieve some common goal as an outcome often leads to a solution that incurs a high implementation cost. In reality, most players may be willing to operate somewhat cooperatively, because they realize that their long term goal of being part of the system will never be fulfilled if everyone is selfish in each realization of the game.[6] This suggests a cooperative formulation from the outset, in which the focus is on preserving cooperation by sufficiently "raising the bar" on cheating. Catch is an example of this approach: it punishes the cheaters rather than rewarding the cooperators. The challenge for the theory then is to show that cheating will never become widespread in this setting.

We believe that game theoretic formulations targeted at mostly cooperative settings would be useful for many other applications such as peer-to-peer systems. Properly quantifying the desire to cooperate is challenging, though, and depends on the problem domain. But if achieved successfully, we believe that it will make game theoretic analyses applicable to a larger class of systems problems. From a systems perspective, the benefits of giving up the notion of perfect selfishness in problems that involve interacting, autonomous agents is analogous to the benefits of relaxing the notion of perfect availability when building distributed systems. In both cases, a slight relaxation of the requirement can result in dramatically lowered implementation costs.

6. ACKNOWLEDGEMENTS

We thank Tom Anderson, Steve Gribble, Anna Karlin, and David Kempe for useful discussions during the course of our work. We also thank the anonymous reviewers for their helpful comments. This work was supported in part by Microsoft Research and DARPA grant F30602-00-2-0565.

7. REFERENCES

[1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. Technical Report MSR-TR-2003-41, Microsoft Research, June 2003.

[2] D. Aguayo, J. Bicket, S. Biswas, D. S. J. D. Couto, and R. Morris. MIT roofnet implementation. http://www.pdos.lcs.mit.edu/roofnet/design/, Aug. 2003.

[3] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, and S. Shenker. Selfish behavior and stability of the Internet: A game-theoretic analysis of TCP. In *ACM SIGCOMM*, Aug. 2002.

[6] As one data point, Saroiu *et al.* found that a large fraction of users in a peer-to-peer file sharing system were cooperative even in the absence of any incentive to cooperate or disincentive against cheating [19].

[4] S. J. Brams. *Negotiation Games: Applying game theory to bargaining and arbitration*. Routeledge, 1990.

[5] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes — fairness in dynamic ad-hoc networks. In *MobiHOC*, June 2002.

[6] L. Buttyan and J. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5), Oct. 2003.

[7] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in cyberspace: Defining tomorrow's Internet. In *ACM SIGCOMM*, Aug. 2002.

[8] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring. Modelling incentives for collaboration in mobile ad hoc networks. In *WiOpt*, Mar. 2003.

[9] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In *ACM Principles of Distributed Computing*, July 2002.

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

[11] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Encouraging cooperation in multi-hop wireless networks. Technical Report CSE-04-06-01, University of Washington, June 2004.

[12] R. Mahajan, D. Wetherall, and T. Anderson. Interdomain routing with negotiation. Technical Report CSE-04-06-02, University of Washington, June 2004.

[13] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating router misbehavior in mobile ad-hoc networks. In *ACM MobiCom*, Aug. 2000.

[14] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2), Apr. 1983. Cited in Brams [4].

[15] M. J. Osborne and A. Rubenstein. *A course in game theory*. MIT Press, 1994.

[16] B. Raghavan and A. C. Snoeren. Priority forwarding in ad hoc networks with self-interested parties. In *Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[17] H. Raiffa. *The art and science of negotiation*. Harvard University Press, 1982.

[18] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2), Mar. 2002.

[19] S. Saroiu, K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking*, Jan. 2002.

[20] S. Shenker. Making greed work in networks: A game-theoretic analysis of switch service disciplines. *IEEE/ACM Transactions on Networking*, 3(6):819–831, Dec. 2003.

[21] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *IEEE INFOCOM*, Mar. 2003.

[22] A. Urpi, M. Bonuccelli, and S. Giordano. Modelling cooperation in mobile ad hoc networks: A formal description of selfishness. In *WiOpt'03 Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Mar. 2003.

[23] S. Zhong, Y. Yang, and J. Chen. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In *IEEE INFOCOM*, Mar. 2003.