Punishment in Selfish Wireless Networks: A Game Theoretic Analysis

Dave Levin

Abstract— **In currently deployed wireless networks, rational participants have no incentive to cooperatively forward traffic for others. Though much work has focused on providing such incentives, few has done so with adequate focus on the ease of deployment; often, these systems require trusted third parties or tamper-proof hardware. In this paper, we use game theory to analyze two *internal incentive mechanisms*, which rely only on the primitives made available by standard 802.11 hardware. We show that isolating free-riders (*i.e.*, refusing to forward through or for them) is not sufficient in all scenarios, and we motivate a new incentive mechanism: punishment via channel jamming. We also show that jamming yields a fair Nash equilibrium for all nodes, *i.e.*, that all nodes can provide incentives to their neighbors to forward their packets. Lastly, we discuss some of the emergent behaviors in these equilibria, as well as guidelines for the design of a jamming strategy.**

I. Introduction

Each participant in a wireless ad hoc network is both end-host (it generates its own data and routing traffic) and infrastructure (it forwards traffic for others). Forwarding others' traffic can consume a considerable amount of battery life, yet no currently deployed wireless routing protocols provide incentives for participants to route or forward. Indeed, *rational*, self-interested nodes will free-ride from currently deployed protocols. To ensure cooperation, protocol designers should build incentives directly into the protocols [15].

Internal vs. External Incentives: Building incentives for nodes in a wireless ad hoc network to cooperate is not a new problem, but most existing systems make assumptions that are simply too strong for a reasonable deployment. These assumptions generally include introducing one of two new components to the wireless network: trusted third parties (*e.g.*, banks) or trusted, tamper-proof hardware. Since these are not inherently part of existing wireless networks, we refer to them as *external incentive mechanisms*. Few systems have focused on what one can call *internal incentive mechanisms*, which make sole use of the primitives already available in deployed wireless (802.11) networks. Yet, systems which use such mechanisms are more likely to experience a timely deployment.

To better understand internal incentive mechanisms, we develop in this paper a game theoretic framework in the form of repeated games played on a strongly connected graph (Section III). Each player's strategy set is limited to what can be feasibly done with standard 802.11 hardware (as opposed to, say, interacting with a bank). Using this model, we analyze the predominant internal incentive mechanism: isolating a node by refusing to forward to or through it [4], [12], [13]. We show in Section IV that *isolation does not always yield system-wide cooperation*.

A New Internal Incentive: Motivated by this observation, we introduce a new incentive mechanism in Section V: punishment via channel jamming. Unlike isolation, jamming does not require any cooperation among nodes to punish a free-rider; a single jammer suffices. This fundamental difference leads us to a proof that *jamming is a sufficient mechanism for encouraging cooperation in all conditions, without requiring any trusted components*. We further extend our framework in Section VI to model noise in wireless networks, which is fundamentally different from the standard game theoretic notion of trembles.

Our game theoretic model shows that jamming is a viable means of punishment, but there are of course several considerations that must be taken into account when designing a system with jamming, such as: When should a node punish another node? How should a node react to another node's punishment? How should a node act to *avoid* being punished? We discuss these in Section VII. Designing a system that addresses these questions is the main focus of our ongoing work; the model presented in this paper is intended to guide this design.

II. Model and Assumptions

We first formalize our assumptions about the network and the nodes' preferences over potential outcomes.

A. Network Model

We assume the network to be an arbitrary, connected graph, $G = (V, E)$, of selfish ad hoc nodes, V. By *selfish* we mean that any $i \in V$ will act in whatever *rational* way that will maximize i's utility over time. Formally, if (u_i^t) and (w_i^t) are sequences representing i's payoffs at time $t = 1, \ldots, T$, then i will prefer u to w if and only if there exists some $\epsilon > 0$ such that $\frac{1}{T}\sum_{i=1}^{T}(u_i^t - w_i^t) > \epsilon$. This condition is also known as the *limit of means criterion* [14].[1]

Edge (u, v) is in E if and only if u and v are within transmission range of each other. We can safely assume that edges are bi-directional, since 802.11 requires link-level acknowledgments [10]. Also, as assumed in the watchdog mechanism [13] and the Catch system [12] systems, when a node $u \in V$ sends a message (be it broadcast or unicast), all nodes in its one-hop neighborhood, $\mathcal{N}^1(u)$, overhear the message. We make this assumption so that we can analyze the resulting equilibria of systems such as Catch.

In terms of end-to-end connections, any two nodes $u, v \in V$ can communicate via some multi-hop path (*i.e.*, G is strongly connected). We assume that each node u knows its active connections and acts in a way that maximizes the sum goodput across these connections.

dml@cs.umd.edu. Department of Computer Science, University of Maryland, College Park. This work is supported by NSF-ITR Award #0426683 and a Sloan Foundation Fellowship awarded to Bobby Bhattacharjee.

[1] Please see [14] for thorough definitions of the game theoretic terms used in this paper.

B. Selfish Nodes' Preferences

It is not possible to formulate a general utility function to accurately capture to what degree each node prefers, say, connectivity over being disconnected. However, here, we present a reasonable set of preferences and assign nominal numeric values to different outcomes. A selfish node u can experience one of four outcomes: being disconnected or not, or (orthogonally) forwarding for other nodes or not. (We extend this to include punishment in Section V.) If the cost to forward is F, the benefit from being connected is C, and the utility gained from being disconnected is D, we have the following preferences:

- $C > D > F (< 0)$: Connectivity is the best outcome, but being disconnected at least does not expend resources, unlike forwarding.
- $C + F > D$: Nodes gain more benefit from being connected than what they lose by forwarding.

We can capture these properties by letting $C = 2$, $D = 0$, and $F = -1$; we assume each of these for the remainder of the paper only for ease of exposition, but these specific numbers do not change any of our fundamental results.

III. Ad Hoc Routing Games

We begin by formulating an *ad hoc routing game* which captures selfish nodes' preferences in a multi-hop wireless network. Such a game has many similarities to the well-known iterated prisoner's dilemma [14], repeated infinitely.[2] However, when modeling an ad hoc network, the game differs from most formulations of N-player games in the following ways:

1) Each node i plays a game with nodes in $\mathcal{N}^1(i)$.
2) The game $G(i, j)$ played between i and $j \in \mathcal{N}^1(i)$ is not necessarily independent of the games played between other nodes in i's two-hop neighborhood, $\mathcal{N}^2(i)$.
3) The payoffs of $G(i, j)$ (and therefore the dynamics of the game itself) depend on whether or not i has any interest in having j forward i's packets, and vice versa.

Hence, each game $G(i, j)$ must have an ever-changing set of payoffs, determined by others' actions and the desired end-to-end connections. We now motivate these three differences from the standard prisoner's dilemma.

Games are played between neighbors: Standard game theoretic models of N-player games generally assume that all N players may (or often must) play against one another. The network extension of games (see [1] for a nice survey) allows for a more suitable model of most networking problems, such as incentives-compatible BGP [8] and network planning [9]. Such a game includes an additional parameter to a game: a graph $G = (V, E)$, such that $|V| = N$ and games are only played between i and j if $(i, j) \in E$. We must therefore define a game for each pair of neighbors, $(i, j) \in E$. As we will see, games between different neighbors can vary significantly.

All of node i's games are interdependent: Let $\mathcal{U}_i^t(j)$ denote the utility i gains from game $G(i, j)$ at time t.[3] If all such games are independent, then the utility i gains from the system at time t is simply $\sum_{j \in \mathcal{N}^1(i)} \mathcal{U}_i^t(j)$. However, such games are not generally independent. For instance, i cannot forward packets for more neighbors at time t than the capacity of the wireless network allows. We assume for the remainder of this paper that the capacity of the wireless network is enough that all interfering nodes may successfully transmit their data in a given game, though in Section VI, we approximate interference with noise.[4] We make use of this interdependence when we introduce the notion of channel jamming as a punishment mechanism; when i is jammed, its utility for time t, $U_i(t)$, is forced to at most zero, regardless of the benefit that would have been gained from the sum of i's other games at time t.

Neighbors' interests may be asymmetric: Consider the example network in Figure 1(a). Node A gains utility from the system only if nodes B and C forward A's packets to node D. However, since B already has its end-to-end connection established (with C), B has no reason to ask A to forward its packets. Hence, there is an asymmetry of desire between A and B; A would gain utility with B's cooperation, but B gains no additional connectivity (and therefore no additional utility) by forwarding for A. Conversely, in the network of Figure 1(b), B and C have a mutual interest in one another, as they both would gain benefit from cooperating.

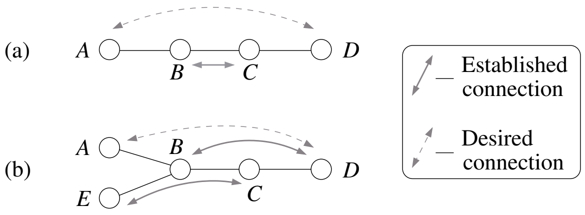

Fig. 1. Sample networks that motivate symmetric and asymmetric versions of the ad hoc routing game.

To capture players' varying desires, we use a different game for each scenario of interests: both are interested, only one is interested, or neither is interested. When both i and j are interested in having the other forward their packets, $G(i, j)$ is the standard prisoner's dilemma:

	Cooperate	Defect
Cooperate	$1, 1$	$-1, 2$
Defect	$2, -1$	$0, 0$

Game 1: The symmetric ad hoc routing game is the prisoner's dilemma. The pure strategy Nash equilibrium is (Defect, Defect).

When neither have interest, all payoffs are zero, since neither would have to spend any utility in forwarding (the other node will not request it), and neither will gain anything from having the other forward (since they have no interest), hence the

[2] Although, strictly speaking, players would not be expected to play infinitely, nodes generally do not know how long they will be in the network, so the game can be treated as infinite [14].

[3] For ease of exposition, we are making the simplifying assumption that time is slotted and that at each slot, a single round of each $G(i, j)$ is played.

[4] A model that more accurately models capacity would require a nonlinear program with constraints across all of i's games, and is an area of future work.

weakly dominant strategy is (Defect, Defect). Lastly, consider Game 2, where there is an asymmetry of interest; player 1 wants player 2 to forward but player 2 has no interest in player 1. For the uninterested player 2, Defect is a dominant strategy, since pl. 2 would gain no benefit from pl. 1 for performing this favor. Hence, the weakly dominant strategy is (Defect, Defect), meaning that any node i will not have its packets forwarded by any node who has no interest in i.

	Cooperate	Defect
Cooperate	2 , -1	0 , 0
Defect	2 , -1	0 , 0

Game 2: An asymmetric ad hoc routing game; pl. 1 wants pl. 2 to forward, but pl. 2 has no packets to forward through pl. 1. Defect is a dominant strategy for pl. 2.

Games 1 and 2 are sufficient to analyze systems that use isolation (defined in the next section) as a means of punishment [12], [13]. We show that isolation does not sufficiently account for the asymmetric game , and we introduce a new mechanism that provides incentive for all nodes to cooperate, independent of their interest in their neighbors.

IV. Punishing with Isolation

An intuitive strategy for enforcing cooperation in an ad hoc routing game is to *isolate* a free-rider f by ensuring that all nodes in $\mathcal{N}^1(f)$ play Defect in games against f, such as in Catch [12]. However, isolation (Defection) is not always a rational strategy for a node $i \in \mathcal{N}^1(f)$ to play in game $G(i, f)$. In particular, if any such $G(i, f)$ is the symmetric game (1), then i will be able to yield greater short-term utility by not isolating (Cooperating with) f. In Figure 1, B has no incentive to forward for A, hence A will attempt to isolate B. However, since C has no incentive to isolate B (at least, not in the short term), A's isolation will fail.

One could argue that, in some cases, there may exist greater long-term gain for $i \in \mathcal{N}^1(f)$ by participating in f's punishment. For instance, if the other neighbors of f were able to detect that i was not participating in f's isolation, then they could subsequently punish i. There are trivial cases where this does not work, *e.g.*, when f and i have an end-to-end connection with one another. Also, there are more general solutions that f and i could employ to make it appear that i is never forwarding for f. For instance, f could simply communicate with i over an encrypted channel, in essence resulting in a one-hop mix network [6].

Hence, as long as free-rider f has at least one neighbor with a mutual interest, (with whom it plays the symmetric Game 1), isolation is not a viable punishment. In fact, the only nodes who are guaranteed to gain no utility once isolation is in effect are the nodes for whom f was not forwarding in the first place.

V. Punishing by Jamming

We have shown that isolation does not guarantee cooperation by all rational nodes. Further, deployable isolation systems, such as Catch [12], seem to require rather strong assumptions: no collusion among nodes, MAC-level authentication, and MAC-level sender anonymity (*e.g.*, that nodes cannot use transmission power measurements to distinguish amongst its neighbors). As protocol designers, we are interested in the question: do there exist punishment strategies that *guarantee* cooperation by all rational nodes and can these assumptions be relaxed?

To this end, we consider channel jamming as a punishment mechanism. A node jams the channel by sending many broadcast packets (generally with no meaningful payload), thereby occupying the channel for all nodes within carrier sense range of the jammer (*e.g.*, its two-hop neighbors). Playing Jam costs J; we require that jamming costs more than forwarding ($|J| > |F|$), and assign J a nominal value of -2. To incorporate jamming into the ad hoc routing games (Games 1 and 2), we must capture the fact that whenever node i jams, none of the nodes in $\mathcal{N}^2(i)$ can receive any packets, and hence none gain utility from their neighbors' Cooperation. Let $c_f(t)$ denote the number of games in which f cooperatively forwards at time t, and recall that $\mathcal{U}^t_f(i)$ is the utility f gains from game $G(f, i)$ at round t. Then the ad hoc routing game with jamming is:

$$U_f(t) \overset{\text{def}}{=} \begin{cases} -2 & \text{if } f \text{ is Jamming} \\ -c_f(t) & \exists i \in \mathcal{N}^2(f) \text{ Jamming} \\ \sum_{i \in \mathcal{N}^2(f)} \mathcal{U}^t_f(i) & \text{otherwise} \end{cases}$$

Game 3: The ad hoc routing game with jamming. When no one in $\mathcal{N}^2(f)$ is jamming, the normal (symmetric or asymmetric) games are played.

Note that, although f cannot gain utility if any $i \in \mathcal{N}^2(f)$ jams, f can still pay the cost of forwarding for others (the second condition). Of course, f has no incentive to forward in this case; indeed, f achieves its minmax payoff (0) by playing Defect whenever any $i \in \mathcal{N}^2(f)$ jams:

Theorem 1: Any node i forces $j \in \mathcal{N}^1(i)$ to j's minmax payoff by Jamming; j in turn will Defect in all of its games.

Proof: The set of j's feasible payoffs when being punished is $(-\infty, 0]$, with 0 being obtained when j plays Defect in all of its symmetric games (Game 1) and asymmetric games (Game 2) where j is the node without interest. When being jammed, the asymmetric game where j is the node with interest have the same outcome, since j will never be asked to forward a packet; w.l.o.g., we can say j Defects in this case, as well. ■

Theorem 1 allows us to apply the well-known folk theorem, but first we require two definitions. A payoff profile (*i.e.*, a vector of utilities), $\mathbf{p} \in \mathbb{R}^N$, is said to be *feasible* if there exists a set of strategies that, when each node i plays its assigned strategy, its payoff is $\mathbf{p}(i)$, the i^{th} component of $\mathbf{p}$. Further, $\mathbf{p}$ is *strictly enforceable* if, for all i, $\mathbf{p}(i)$ is greater than i's minmax payoff; in effect, $\mathbf{p}$ is enforced by punishing nodes (forcing them to their minmax payoff) whenever they deviate from the strategy that would yield $\mathbf{p}$.

Theorem 2 (Folk Theorem [14]): Every feasible, strictly enforceable payoff profile of a game G is a subgame perfect Nash equilibrium payoff profile of the infinitely repeated version of G with the limit of means criterion (Section II).

Theorem 3: There exist subgame perfect Nash equilibria

(SPNE) with payoffs greater than system-wide defection that use jamming to punish free-riders.

Proof: By Theorem 1, jamming yields a minmax payoff. Any feasible payoff profile with payoffs greater than system-wide defection is therefore enforceable by jamming. Hence, by Theorem 2, such a profile is the payoff of at least one SPNE in which jamming is used as punishment. ■

Although Theorem 3 states that punishment can yield SPNE, it (like the folk theorem in general) does not specify precisely *how* to obtain these equilibria. Designing protocols (and punishment strategies) that yield these SPNE is a main focus of our ongoing work, and we discuss some of the necessary considerations in Section VII.

The Price of Jamming: To the best of our knowledge, jamming has only been studied as an attack, and this is not without reason. Even as a punishment mechanism, it incurs a loss of efficiency, since it pauses all connections within the jammer's two-hop neighborhood for the duration of the punishment. Additionally, it decreases the expected lifetime of the network as a whole, as nodes must expend additional energy to jam. Designing a punishment strategy that balances between this loss of efficiency and the gain of cooperation is the goal of our future work.

VI. Noisy Games

When all nodes act rationally, *and when all actions taken by i are viewed perfectly by $\mathcal{N}^1(i)$*, each node will cooperatively forward for others, and Jam will never be played. However, since wireless networks are inherently noisy, j will not overhear some of the packets i forwards on j's behalf. In the terms of our model, this means that with some probability, when i plays Cooperate in game $G(i, j)$, j will view i's action as Defect, even though i has paid the cost to Cooperate, F. For example, in Figure 1(b), B could have forwarded a packet to E for C at the same time that D sent a packet to C, resulting in a collision at C. Hence, although B cooperated, C is not able to verify; if this happens significantly more than the noise itself would cause, C must assume that B is defecting.

Nodes may not always cooperate, since they know that some of their defections could be interpreted as noise. In this section, we incorporate noise into the ad hoc routing game, and examine some of the resulting emergent behaviors.

A. Ad Hoc Routing with Noise

The notion of noise in a wireless network is fundamentally different from the standard game theoretic notion of *trembles*. In a game with trembles, when a player i chooses a strategy, there is some probability p that i tremble, *i.e.*, play a different strategy instead. If node i chose to Cooperate but trembled, it would simply play Defect instead, and vice versa, giving us the following game.[5]

	Cooperate	Defect
Cooperate	$1-p\ ,\ 1-p$	$3p-1\ ,\ 2-4p+p^2$
Defect	$2-4p+p^2\ ,\ 3p-1$	$p\ ,\ p$

Game 4: The symmetric ad hoc routing game with standard game theoretic trembles with probability p.

However, Game 4 does not accurately capture the notion of noise in a wireless network. To see this difference, observe that when i's Cooperation is not viewed by j, i still has to pay the cost of forwarding, but does not gain the benefit of cooperation. Let U_c and U_d denote the utility that a node gains when it views its *opponent* playing Cooperate or Defect, respectively. In the symmetric game and for the interested node in the asymmetric game, $U_c = 2$ and $U_d = 0$ (because they will have to retransmit), whereas for the uninterested node in the asymmetric game, $U_c = U_d = 0$. We will modify slightly the definition of p: in our model, p represents the probability that node j views i's action as Defect, given that i actually played Cooperate.[6] Then the expected utility of cooperation and defection are:

$$\begin{aligned} E_c &\overset{\text{def}}{=} \text{E[Cooperate]} &= F + (1-p)U_c + pU_d \\ E_d &\overset{\text{def}}{=} \text{E[Defect]} &= U_d \end{aligned}$$

Note that when $p = 1$, $E_c = F + U_d \leq E_d$, so player i should always defect; the obvious correlation of this is that nodes ought not attempt to forward packets when the error rate is 1. We will assume for the remainder of the paper that $p < 1$. We can now formulate the following ad hoc routing games with noise; we derive these values by plugging in the values for U_c and U_d above.

	Cooperate	Defect
Cooperate	$1-2p\ ,\ 1-2p$	-1 , $2-2p$
Defect	$2-2p$, -1	0 , 0

Game 5: The symmetric ad hoc routing game with a more accurate model of noise. When $p = 0$, this is Game 1.

	Cooperate	Defect
Cooperate	$2-2p$, -1	0 , 0
Defect	$2-2p$, -1	0 , 0

Game 6: The asymmetric ad hoc routing game with a more accurate model of noise. When $p = 0$, this is Game 2.

Regardless of p, the minmax payoff for node i is achieved with (Defect, Jam), resulting in a total of 0 payoff for i at time t (similarly for j). When $p < 1$, the pure strategy Nash equilibrium of Game 5 does not differ from the corresponding games without noise: (Defect, Defect).

B. Playing with a Watchdog

In practice, detecting a neighbor's strategy requires a watchdog-like system [13]. The watchdog mechanism makes the standard assumption that whenever j forwards a message, all nodes $i \in \mathcal{N}^1(j)$ overhear the message, and can therefore determine if and when j has forwarded a packet on i's request. One way to implement a watchdog is as follows: each node i stores the weighted averages of $r_i(j)$, the number of unique packets that i requested j to forward, and $f_i(j)$, the number

[5] Note that, again, the strategies listed correspond to those *chosen* by the players, not the strategies that are necessarily played.

[6] To be precise, p would be a function of i and j, depending on the available capacity at the two nodes' respective locations in the network.

of these packets that j actually did forward. In a game with no noise (Game 3), $f_i(j) = r_i(j)$ for all i, j, since all nodes will cooperate to avoid punishment by jamming. However, when there exists noise, nodes will drop as many packets as they can while still avoiding punishment.

In a watchdog and in Catch [12], each node maintains a parameter θ, the threshold value that, if $f_i(j)/r_i(j) < \theta$, then node i considers j misbehaving. This parameter could change depending on the capacity of the wireless network, which depends in part on the bandwidths of neighboring links and the two-hop neighbors' desired flow rates [11]. We use the threshold value θ in defining nodes' punishment strategy. First, we show how a low threshold value (or high amount of noise) can lead to obsequious behavior in the network.

C. Avoiding Punishment with Forward Error Correction

The more a node i's neighbor perceives a defection from i, the greater the risk i has of being punished. Nodes can react to a high error rate (p) by employing some form of forward error correction (FEC). For ease of exposition, and to gain insight into what effect FEC has on nodes' strategies, we consider a naïve form of FEC, in which node i sends each packet multiple times, thereby "replacing" p with a smaller value. Of course, in practice, such a scheme would fail in the presence of high levels of congestion, but, again for clarity, we will assume failures are independent and, as stated in Section II, that the capacity of the network is infinite. Under these assumptions, if a node retransmits a packet r times (the *r-FEC strategy*), the probability that the previous hop will not see any of these is p^r. The expected utility from cooperating with r-FEC is thus:

$$E_c^r \overset{\text{def}}{=} \text{E[Coop w/ } r\text{-FEC]} = rF + p^r U_d + (1 - p^r)U_c$$

Since U_c is gained at most once, this captures the fact that i's neighbor will not compensate a forwarded packet multiple times. The r-FEC strategy strictly dominates[7] the normal, single-transmission Cooperation when $E_c^r > E_c$, or

$$\begin{aligned} rF + p^r U_d + (1 - p^r)U_c &> F + pU_d + (1 - p)U_c \\ (U_c - U_d)(p - p^r) &> |F|(r - 1) \end{aligned} \qquad (1)$$

In other words, nodes will employ forward error correction whenever the cost to forward the extra $r - 1$ times (r.h.s.) is compensated by a greater expected value of utility (l.h.s.).

D. Emergent Behaviors

The resulting system-wide behavior of nodes depends on the punishment strategies they employ. A punishment strategy is a tuple (θ, δ), where θ is the threshold of free-riding at which to begin punishing (larger is more generous), and δ is the duration of the punishment (smaller is more generous). Hence, the *strength* of the punishment is proportional to δ/θ. For instance, a small θ and a large δ correspond to harsh, long punishments after the slightest noise or defection. Conversely, a high θ and low δ correspond to a generous node that punishes rarely and, if at all, for short durations. Here, we consider the behaviors that result from three different regimes of punishment strength.

Generosity Leads to Free-Riding: A node can be *generous* toward its neighbors by assuming a considerable amount of noise (*i.e.*, choosing a large θ) and punishing for a short duration (a small δ). Increased generosity allows for free-riding, as nodes exploit the large difference between how much they must forward and how much they are requested to forward ($f_i(j)$ and $r_i(j)$ from our watchdog). They do so without having to pay much price, even when they are discovered (since δ is small). However, generosity may be the best strategy when a level of trust is established between neighbors; this course of action will be the most resilient to spikes in noise or non-stop failures.

Stronger Punishment Leads to Obsequiousness: When δ/θ is high, there can be a large difference between U_c and U_d in Eq. (1), thereby making even our naïve version of FEC a viable strategy. As δ/θ continues to grow, the obvious price is the efficiency of the network as a whole; rampant jamming can vastly degrade capacity, and, since it expends more energy, the lifetime of the network will decrease, eventually leading to a disconnected network. One potential method to keep nodes from excessive jamming is to punish them by jamming in return, but this of course carries its own loss of efficiency (at least in the short term, until the nodes react to the punishment and change their strategy).

Efficiency by Matching Noise Levels: These two extreme punishment strategies (very low and very high δ/θ) incur a loss of efficiency: the former due to free-riding, the latter due to excessive jamming. We expect that the ideal outcome would be one in which the punishment strategy is *as tightly coupled to the given noise level as possible.* Studying such strategies is a focus of ongoing work.

VII. Choosing a Jamming Strategy

Varying punishment strategies can yield vastly different system-wide behaviors, ranging from incurring low overhead while allowing rampant free-riding, to punishing beyond any reasonable expectation of cooperation. Clearly, these two extreme points ought to be avoided, but the fundamental question as protocol designers is: *what punishment strategy yields the most system-wide efficiency and/or fairness?* We do not present a formal punishment strategy here, but we briefly discuss some guidelines worth consideration. Recall that a punishment strategy is a tuple, (θ, δ), consisting of the threshold θ (as defined by our watchdog) at which to begin punishing, and the duration of the punishment, δ.

The strategy should be adaptive: A viable punishment strategy must be adaptive, allowing θ to change as the available capacity of the wireless network changes to reflect new (or completed) connections. Along these same lines, the punishment strategy should ideally incorporate measurements of the available capacity into its calculation of θ. Available capacity can be measured locally at each node by calculating the link-level error rates and bandwidths, as well as the fraction of time for which the wireless channel is idle.

Punishments ought not echo: When node i punishes some $j \in \mathcal{N}^1(i)$ by jamming, all of the nodes within carrier sense range ($\mathcal{N}^2(i)$) are affected and, worse yet, the nodes

[7] r-FEC weakly dominates under equality of Eq. (1).

in $\mathcal{N}^3(i) \setminus \mathcal{N}^2(i)$ do not necessarily know that i is even punishing. Hence, node $m \in \mathcal{N}^3(i) \setminus \mathcal{N}^2(i)$ could perceive the defection of $k \in \mathcal{N}^2(i)$ as a response to $G(k,m)$, and not as k playing its minmax strategy against i's punishment. Node m may therefore punish k, which then raises the same issue for i, since $i \in \mathcal{N}^3(m)$, and so on.

In effect, a single node's jamming can *echo* throughout the network, potentially indefinitely. To address this, it may be necessary to only jam with some probability small enough to limit the extent of such an echo. Addressing the echo of jamming is an area of future work.

Sharing one's views: Given issues such as the hidden node problem, it is not always the case that i knows when noise even takes place between itself and its neighbor. To help nodes understand the level of noise, p, each node i could forward to each of its neighbors, j, the values i's watchdog is storing to compute j's level of defection: $f_i(j)$ and $r_i(j)$. This is precisely what is done to compute link-level error rates for path metrics such as ETT [7]. Based on $f_i(j)$ and $r_i(j)$, j could estimate p and, if need be, choose an r with which to play r-FEC. An open question is how to ensure truthfulness in reporting $f_i(j)$ and $r_i(j)$.

VIII. Related Work

We briefly review existing systems that provide incentives to forward in wireless networks, as well as some known results about games played in noisy environments.

Systems with Incentives-Compatible Forwarding: Previous such systems can be categorized into two classes:

Payment schemes generally involve a trusted third party (TTP) [2], [17] or tamper-proof hardware [5] that generates digital currency. Peers pay others with tokens to forward data and route requests.

Detection and avoidance systems consist of two parts: (i) a *watchdog* that each node runs locally to determine when one of its neighbors is not forwarding data its data, and (ii) a policy to avoid sending to or forwarding on behalf of these defectors [4], [12], [13].

Neither of these types of systems can be used to apply incentives in a general setting. For instance, TTP-based payment schemes generally assume that wireless nodes have access (albeit infrequent) to the TTP itself. Perhaps future hardware will contain trusted, tamper-proof components that would remove the need for TTPs in payment schemes, but recent trends in digital rights management (DRM) indicate that this deployment would be slow, expensive, and hardly trusted after all [3].

Theory of Games in Noisy Environments: Previous game theoretic work on noise in the prisoner's dilemma (Game 1) has focused on the notion of trembles, some of the most influential work by Axelrod et al. Wu and Axelrod experimentally analyzed several strategies in an environment where nodes trembled, *i.e.*, when a player chose to play an action, the other action was, with some probability p, played instead [16]. Wu and Axelrod showed that, in the presence of more trembles, it is better for nodes to accept their punishment (*i.e.*, play "contrite tit-for-tat") when they defect than it is to simply act generously. This result is reflected in our proposed solution for stopping the echo of punishments. However, since the notion of a tremble is so different from that of noise in wireless networks (Section VI), it is not clear to what extent Wu and Axelrod's results apply.

IX. Discussion and Future Work

In this paper, we have developed a game theoretic model to analyze existing internal incentive mechanisms in wireless networks (*i.e.*, mechanisms that require only the primitives available in 802.11), and have introduced a new mechanism: punishment via channel jamming. We showed that isolation does not always ensure cooperation. On the other hand, jamming, though seemingly malicious, is a viable means by which to enforce cooperation of each node in the system, even when there are neighbors acting in a collusive manner by communicating only with one another. The price of jamming, if not engineered in a careful manner, can be high; jamming could, for example, echo throughout the network, resulting in a significant loss of efficiency. The main focus of our future work is in developing a viable punishment strategy that balances between the price of jamming and the gain of provable system-wide cooperation.

Acknowledgments

The author thanks Seungjoon Lee, Vijay Gopalakrishnan, and Bobby Bhattacharjee for their helpful discussion and comments throughout this work.

References

[1] E. Altman, T. Boulogne, R. Azouzi, and T. Jimenez. A survey on networking games in telecommunications, 2000.

[2] L. Anderegg and S. Eidenbenz. Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proc. of MobiCom*, 2003.

[3] R. Anderson. 'Trusted Computing' Frequently Asked Questions. Online: http://www.cl.cam.ac.uk/˜rja14/tcpa-faq.html.

[4] S. Buchegger and J.-Y. L. Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks. In *MobiHoc*, 2002.

[5] L. Buttyán and J.-P. Hubaux. Enforcing Service Availability in Mobile Ad-Hoc WANs. In *Proceedings of ACM MobiHoc*, pages 87–96. IEEE Press, 2000.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security*, 2004.

[7] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. of Mobicom*. ACM Press, 2004.

[8] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based Mechanism for Lowest-Cost Routing. In *PODC*, 2002.

[9] A. Gupta, A. Srinivasan, and É. Tardos. Cost-Sharing Mechanisms for Network Design. In *APPROX-RANDOM*, 2004.

[10] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE 802.11 Standard, 1999.

[11] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic Aspects of Capacity in Wireless Networks. In *ACM SIGMETRICS*, 2005.

[12] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining Cooperation in Multi-hop Wireless Networks. In *NSDI*, 2005.

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

[14] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[15] C. Papadimitriou. Algorithms, Games, and the Internet. In *STOC*, 2001.

[16] J. Wu and R. Axelrod. How to Cope with Noise in the Iterated Prisoner's Dilemma. *Journal of Conflict Resolution*, 1995.

[17] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *Infocom*, 2003.