Malicious Users in Unstructured Networks

George Theodorakopoulos and John S. Baras
Department of Electrical and Computer Engineering and the Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: {gtheodor, baras}@isr.umd.edu

***Abstract*—Unstructured networks (like ad-hoc or peer-to-peer networks) are networks without centralized control of their operation. Users make local decisions regarding whether to follow the network protocol or not. While providing scalability benefits, this degrades the performance, which is compounded by the potential presence of Malicious Users. In general, these users are trying to disrupt the operation of the network, and prevent the legitimate users from achieving their objectives. More specifically, they could try to break the connectivity of the network, or waste the resources of the legitimate users.**

In this work we use game theory to examine the effect of Malicious Users. All users are modeled as payoff-maximizing strategic agents. A simple model, *fictitious play*, is used for the legitimate user behavior, but no limits are imposed on the Malicious Users strategies. We look for the worst case equilibrium: the one that gives Malicious Users the highest payoff. We identify the importance of the network topology.

I. Introduction

We call unstructured networks those networks that have no hierarchy, and in which all users are equal when it comes to duties or responsibilities. We do not assume anything about the topology, except that the users are connected to each other either directly or through others. With each such network, there is an associated protocol which is the way that the network is supposed to operate.

We abstract the role of users of unstructured networks in the following way: They can choose whether to participate in the operation of the network or not and, if yes, to what degree (all the time? some of the time?). Moreover, there is a cost associated with choosing to participate, so, in general some users will decide to participate and some not.

For example, in the case of a wireless ad-hoc network, participation means forwarding other users packets. The incentive is the expectation the user has that other users will also participate, i.e., forward his packets. The disincentive is that the very action of transmitting data reduces a users available energy, which is scarce in such networks. Moreover, the user wastes his bandwidth, which he could be using to forward his own data.

Although wireless networks are our initial motivation and our running example in this paper, similar considerations apply to other types of unstructured networks. In peer-to-peer networks (P2P), file sharing protocols depend on the cooperation of the users to succeed: As an extreme example, if all users want to download files but no one wants to upload any, then the network collapses. The incentive for participation (uploading) is the increased total availability of files, from which everyone benefits. The disincentive is the increased usage of upload bandwidth, and also, potentially, the unwillingness to continue uploading after ones own download is complete (see also Section II). The salient features of our model are: A protocol that the users can follow or break, a benefit that comes from participating in the protocol, and a cost associated with that participation. The benefit increases with the number of other cooperating users, while the cost reflects the resource usage that participation requires.

Note that the benefit of cooperation is somewhat more abstract, global, and indirect than the cost. So, it could be argued that some of the Good users may behave selfishly, and as a result will not take very seriously the incentive that a globally desirable outcome presents. But we consider that users are either Good or Bad, not selfish or unselfish. In particular, all Good nodes behave equally unselfishly in the sense that, in principle, they value the network benefit more than their individual cost. This will be explained in greater detail in the discussion on the user model (Section II), but it should not be taken to mean that Good users unconditionally cooperate. If, for instance, none of a Good users neighbors cooperate (i.e., they do not forward his packets, or they do not upload anything to him), then the Good user will stop cooperating despite being Good.

We use a game theoretic model for the above situation, described in detail in Section II. We consider Malicious[1] users whose objective is to disrupt the operation of the network, and waste the resources of the Good users. In particular, they are defined as aiming to do the exact opposite of what the Good users want. We consider a repeated game framework. That is, the time is divided in slots and players choose in every slot whether to cooperate or not. In general, they can base their decision on what has already transpired. In our case, we will assume that the Good users aggregate the information about the past in a particular way, namely *fictitious play* [1].

In the literature, this situation has been studied for selfish users, and how to provide incentives to make them cooperate. To the best of our knowledge, there has been no game theoretic modeling of Malicious Users as we describe them here. The work by Blanc, Liu, and Vahdat [2] is an example of providing incentives for users to cooperate (another example is Buttyán and Hubaux [3]). However, they are modeling Malicious Users as "Never Cooperative", without any further sophistication,

[1] We will use the terms Malicious and Bad interchangeably.

since their main focus was discouraging selfish free-riders. There is no degree of selfishness that can approximate the behavior of our Malicious Users. For example, Félegyházi, Hubaux and Buttyán [4] assume that the payoff function of a user is non-decreasing in the throughput experienced by the user. Our Bad users do not care about their data being transmitted. For the same reason, the model proposed by Urpi, Bonuccelli, and Giordano [5] does not apply (as the authors themselves point out).

In other related work, Srinivasan, Nuggehalli, Chiasserini, and Rao [6] are using a modified version of Generous Tit For Tat (for an early famous paper in the history of Tit for Tat see [7]), but they have no notion of topology and, consequently, of neighborhoods. In their setting, each user is comparing his own frequency of cooperation to the aggregate frequency of cooperation of the rest of the network. Altman, Kherani, Michiardi, and Molva [8] proposed a scheme for punishing users whose frequency of cooperation is below the one dictated by a certain Nash equilibrium. Aimed particularly against free-riding in wireless networks is the work by Mahajan, Rodrig, Wetherall, and Zahorjan [9], and also the one by Feldman, Papadimitriou, Chuang, and Stoica [10].

Malicious users and attacks have been mostly considered from a system perspective for particular protocols or algorithms (e.g. securing Distributed Hash Tables [11], or from a cryptographic viewpoint: key exchanges, authentication, etc.). For a summary from the perspective of P2P networks, see Ref. [12].

II. Malicious and Legitimate User Model

The network is modeled as an undirected graph $G = (V, L)$, where each node in V corresponds to one user. An edge $(i, j) \in L$ means that there is a communication link between the users corresponding to nodes i and j. The set of neighbors of user i, denoted N_i, is the set of users j such that there exists an edge (i, j):

$$N_i = \{j \in V | (i, j) \in L\}. \quad (1)$$

The neighbors of user i are also called adjacent nodes to i. Since the graph is undirected, the neighbor relationship is symmetric: $j \in N_i \Leftrightarrow i \in N_j$. The assumption for an undirected graph can be dropped, in order to model asymmetric links, but we believe the extension to be straightforward. We denote the set of Bad users by V_B, and the set of Good users by V_G. It holds that $V_B \cap V_G = \emptyset$ and $V_B \cup V_G = V$. We will be using the term *type* of a user for the property of being Good or Bad.

Users have a choice between two actions: C (for Cooperate), and D (for Defect). When all users choose their actions, each user receives a payoff that depends on three things: his own action, his neighbors'[2] actions, and his own type (but not his neighbors' types). The payoff is decomposed as a sum of payoffs, one for each link. Each term of the sum depends on the user's own action, and the action and type of his neighbor along that link. Observe that the user is playing the same action against all neighbors. As an extension, a user's actions could be different for different links. The whole issue is about how much granularity of control each user has. If all the user can (or wants to) do is turn a switch ON or OFF, then the only model that can be used is a single action for all neighbors.

[2]In-neighbors', if the graph is directed.

Bad

Good	C	D
C	$N-E, E-N$	$-E, E$
D	$0, 0$	$0, 0$

Good

Good	C	D
C	$N-E, N-E$	$-E, 0$
D	$0, -E$	$0, 0$

Fig. 1. The two games that can take place on a link: Good versus Bad and Good versus Good.

The payoff of user i is denoted by $R_i(a_i|t_i)$, when i's action is a_i and i's type is t_i. We extend and slightly abuse this notation to denote by $R_i(a_i a_j|t_i)$ the payoff for i when j is a neighbor of i and j's action is a_j. So, the decomposition of i's payoff can be written as:

$$R_i(a_i|t_i) = \sum_{j \in N_i} R_i(a_i a_j|t_i) \quad (2)$$

We assume there are no links between any two Bad users. The Bad users are supposed to be able to communicate and coordinate perfectly; hence, there is no need to restrict their interaction by modeling it in these terms. Moreover, the Bad users know exactly both the topology and the type of each user in the network. Good users only know their local topology, e.g., how many neighbors they have and what each one of them plays, but not their types.

We explain the payoffs as follows: In the example of a wireless network, a C means that a user makes himself available for communication, that is, forwarding traffic of other nodes through himself. A link becomes active (i.e., data is exchanged over it) only when the users on both endpoints of the link cooperate, that is, play C. Playing C is in line with what Good users want to achieve – good network operation – but it costs energy, since it means receiving and forwarding data. So, when both players on a link play C, the Good player (or both players, if they are both Good) receives N (for Network) minus E (for Energy) for a total of $N - E$. We assume $N > E > 0$, otherwise no player would have an incentive to play C. On the other hand, when a Good player plays C and the other player D, then the Good player only wastes his energy since the other endpoint is not receiving or forwarding any data. For this reason, the payoff is only $-E$. The Bad user's payoff is always the opposite of the Good user's payoff. In particular, we do not assume any energy expenditure when the Bad users play C.

The payoffs are shown in table form in Fig. 1 for the two pairs of types that can arise (Good versus Good, and Good versus Bad). Using the R-notation, the payoffs for a Good

player would be:

$$R_i(CC|G) = N - E \quad (3)$$
$$R_i(CD|G) = -E \quad (4)$$
$$R_i(DC|G) = 0 \quad (5)$$
$$R_i(DD|G) = 0, \quad (6)$$

while for a Bad player they would be:

$$R_i(CC|B) = E - N \quad (7)$$
$$R_i(CD|B) = 0 \quad (8)$$
$$R_i(DC|B) = E \quad (9)$$
$$R_i(DD|B) = 0. \quad (10)$$

In a peer-to-peer network, a C would mean uploading high quality content (as well as, of course, downloading), and a D would be the opposite (e.g. only downloading). The benefit of cooperation is the increased total availability of files. The cost of cooperation E could be, for instance, the hassle and possible expense associated with continuing to upload after one's download is over. In a general social network, edges would correspond to social interactions, a C would mean cooperating with one's neighbors toward a socially desirable objective (like cleaning the snow from the sidewalk in front of your house), and a D would mean the opposite of a C. The cost E and benefit N are also obvious here.

We consider that the game is played repeatedly with an infinite horizon, and time is divided in rounds $t = 1, 2, 3, \ldots$. Actions and payoffs of round t are denoted with a superscript t: a_i^t and R_i^t. The objective of the players in a repeated game is to maximize a function of the sequence of payoffs that they accumulate over the infinite course of the game. In this paper, we consider the average of the payoffs to be the payoff for the whole game:

$$R_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} R_i^t. \quad (11)$$

For other payoff functions, and for repeated games in general, see [13] or [14].

In repeated games, the players are allowed to have full or partial memory of the past actions. Here, we allow the Bad users to have all information about the past (their own moves, as well as everybody else's moves since the first round). On the other hand, the Good users follow a *fictitious play* process, that is, they assume that each of their neighbors chooses his actions independently and identically distributed according to a probability distribution with unknown parameters (Bernoulli in this case, since there are only two actions available: C and D). So, at each round they are choosing the action that maximizes their payoff given the estimates they have for each of their neighbors' strategies. For example, if player i has observed that player j has played c Cs and d Ds in the first $c + d$ rounds, then i assumes that in round $t = c + d + 1$, j will play C with probability $\frac{c}{c+d}$ and D with probability $\frac{d}{c+d}$. We denote by q_j^t the estimated probability that j will play C in round $t + 1$, which is based on j's actions in rounds $1, \ldots, t$.

Let us now calculate the expected payoff for each of the two actions of a Good user. We assume that t rounds have been completed, and Good user i is contemplating his move in round $t + 1$.

$$\begin{aligned} R_i(C|G) &= \sum_{j\in N_i} \left\{ q_j^t R_i(CC|G) + (1 - q_j^t) R_i(CD|G) \right\} \\ &= \sum_{j\in N_i} \left\{ q_j^t N - E \right\} \\ &= N \cdot \sum_{j\in N_i} q_j^t - |N_i| \cdot E \\ R_i(D|G) &= 0. \end{aligned} \quad (12)$$

So, in order to decide what to play, user i has to compare the expected payoff that each action will bring. Action C will be chosen if and only if $R_i(C|G) \geq R_i(D|G)$, that is, iff

$$\sum_{j\in N_i} q_j^t \geq |N_i| \frac{E}{N}. \quad (13)$$

Therefore, the Good user will add the estimates for his neighbors and compare to the quantity $|N_i|\frac{E}{N}$ to decide whether to play C or D. This makes the implementation of the Good user behavior particularly simple, since they only need to keep track of one number for each neighbor, as opposed to the whole history of actions observed. Our main aim in this paper is to see what the Bad users can do against this strategy[3]. On the one hand, perhaps they can exploit its simplicity to gain the upper hand against the Good users. On the other hand, the fact that the Bad users have full knowledge of the history and can achieve perfect coordination in their actions may not be very useful here. The reason is that Good users only care about the frequencies of the actions that they observe (see Eq. (13)), and the Bad users will achieve nothing by more elaborate strategies. This point is worth repeating: The only relevant decision that the Bad users can make is to choose the frequency of their Cs and Ds. Nothing more elaborate than that will be noticed by the Good users. So, we will be assuming that the Bad players can only choose a fixed probability with which they will be playing C.

The choice of payoff that we made (Eq. (11)) implies that only the steady state matters when discussing the sequences of actions of the users. In particular, it allows us to talk about frequencies instead of probabilities, since frequencies will have converged in the steady state. Initial (a priori) estimates will not matter. What does matter, however, is our assumption that the Good users start out by playing C. If that is not the case, the game will in general converge to an equilibrium where all users are playing D.

As a justification for the choice of fictitious play as the process that the Good users follow, we note that very similar behavior assumptions have been done when computing trust and reputation values. In our paper, a user's reputation corresponds to the probability that he plays C. An example is

[3] A strategy is, in the case of a repeated game, a function that maps histories up to round t to probabilities of actions for round $t + 1$.

the work by Buchegger, and Jean-Yves Le Boudec [15], who, in summary, are assuming that a user can misbehave with an unknown probability θ. This probability is then estimated by gathering observations and updating a prior distribution (a Beta distribution) through Bayes' law. The Beta distribution function has also been used in Ismail and Jøsang's work [16]. It is particularly useful because it can count positive and negative events, which are often used to represent a user's behavior in a network, and then compute his reputation.

III. Searching for a Nash Equilibrium

In game theory, the solution concept we are dealing with most frequently is the Nash Equilibrium. The Nash Equilibrium is a tuple of strategies, one strategy for each player, with the property that no single player would benefit from changing his own equilibrium strategy, given that everybody else follows theirs. In our case, we have already restricted the Good players' strategies to fictitious play, as shown in Eq. (13), so the Nash Equilibrium will be restricted in this sense.

More formally, the Nash Equilibrium in our case would be a vector $\vec{q} \in [0,1]^{|V|}$, where q_i is the frequency with which user i plays C. The subvector $\vec{q}_G$ corresponding to the Good users will contain only 0s and 1s, according to whether Eq. (13) is false or true, respectively. For example, if Eq. (13) is false for user i, then the i^{th} element of $\vec{q}$ will be zero. The subvector $\vec{q}_B$ corresponding to the Bad users will contain values in $[0,1]$ such that no other value of q_i would increase the payoff of Bad user i. The payoff of Bad user i when he is playing C with frequency q_i is computed as follows:

$$\begin{aligned} R_i(q_i|B) = & \sum_{j \in N_i : q_j = 1} \{q_i R_i(CC|B) + (1-q_i) R_i(DC|B)\} \\ & + \sum_{j \in N_i : q_j = 0} \{q_i R_i(CD|B) + (1-q_i) R_i(DD|B)\} \\ = & \sum_{j \in N_i : q_j = 1} \{q_i (E-N) + (1-q_i) E\} \\ & + \sum_{j \in N_i : q_j = 0} \{q_i \cdot 0 + (1-q_i) \cdot 0\} \\ = & (E - N q_i) |\{j \in N_i : q_j = 1\}| \end{aligned} \tag{14}$$

Note that since a Bad user only has Good neighbors, and Good users only play always C or always D, the q_j ($j \in N_i$) will all be either 0 or 1.

The problem with the equation we just wrote is that it is not immediately obvious what the cardinality of the set $\{j \in N_i : q_j = 1\}$ is. This set contains the neighbors of a Bad user that play C. In general, what a Good user plays will be affected by the choice of frequencies by all the Bad users. Actually, it will not just be affected: it will be completely determined. Given a choice of frequencies $\vec{q}_B$ by the Bad users, we describe an algorithm (see Fig. 2) to compute the optimal responses of the Good users, which will in turn determine the payoffs for every user (Good or Bad). Our purpose in presenting this algorithm is to show how Good users' actions would affect one another and Ds would propagate through the network. It is not claimed to be the most efficient to compute what the equilibrium is.

In a few words, the algorithm starts by assuming that all Good users play C. Then, they check the validity of Eq. (13), and some start playing D. Then, the ones who started playing D may cause their Good neighbors who are still playing C to also start playing D. The set S contains these users who still need to (re)check Eq. (13), since its validity may have changed.

```
EQUILIBRIUM(G, q_B)
    ▷ All Good users are initialized to playing C,
    ▷ i.e., q_G contains only 1s.
    ▷ S contains users that may switch from C to D.
1   S ← V_G
2   while S ≠ ∅
3         do i ← REMOVE(S)
4            if SUM(i) < (E/N)|N_i|
5               then
6                    q_i ← 0
7                    S ← S ∪ {j | j ∈ N_i ∧ j ∈ V_G ∧ q_j = 1}
```

Fig. 2. The algorithm EQUILIBRIUM computes the optimal actions (C or D) for the Good users, given the subvector $\vec{q}_B$ of the Bad user frequencies. The procedure SUM(i) returns the sum of the frequencies of i's neighbors. Since a Good user switching from C to D could cause other Good users to switch from C to D, the while-loop needs to run until there are no more candidates for switching.

In general, a change in the frequency q_i of a Bad player i will affect the payoffs of other Bad players, too, since it will change the optimal responses of Good users that could, e.g., be common neighbors of i and other Bad users. So, the definition of the Nash Equilibrium we gave earlier could be expanded to regard the Bad players as a team that aims to maximize the total payoff, rather than each Bad user trying to maximize his own individual payoff. In Section III-A, we will see a case when local maximization by each Bad user is equivalent to maximization of the sum of all the Bad users' payoffs. In Section III-B we will examine more closely why the two objectives are in general different and will describe a heuristic for the general case.

A. The Uncoupled Case

Let us look at the case of a single Bad player in the whole network. Since no other Bad players exist, the choice of the Bad user will only affect his own payoff. We will see what he has to do (i.e., with what frequency to play C) in order to maximize his payoff, in a tree topology where the Bad player is at the root of the tree. In Figure 3 we see the root (Bad user) and the one-hop neighbors only.

Assume that the Bad user – labeled user 0 – has k neighbors, labeled $1, \ldots, k$. We also assume that all the Good users will start by playing C, and will only change to D if they are forced by the Bad user. Applying Eq. (13) for each neighbor, we see that each expects to see a different sum of frequencies from his own neighbors in order to keep playing C. User i expects to see a sum of frequencies that is at least $\frac{E}{N}|N_i|$. Since all of

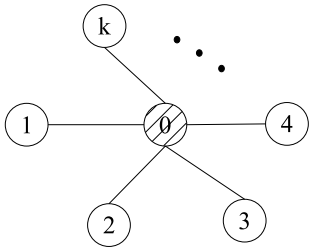

Fig. 3. Single Bad user maximizes his own local payoff.

i's neighbors except user 0 are Good, they will at least start by playing C, so user i will see a sum of frequencies equal to $|N_i| - 1 + q_0$ (q_0 is the frequency with which the Bad user 0 is playing C). So, in order to make user i continue playing C, the Bad user should play C with frequency

$$q_0 \geq \frac{E}{N}|N_i| - (|N_i| - 1) = 1 - |N_i|(1 - \frac{E}{N}) \equiv t_i, \quad (15)$$

which is decreasing with $|N_i|$, since $E < N$. We call this quantity the *threshold* t_i:

Definition 1: The *threshold* t_i of a Good user i with g Good neighbors is the sum of frequencies of Cs that his $|N_i| - g$ Bad neighbors need to play, so that he keeps playing C. We assume that all his Good neighbors play C.

$$t_i = \max\left\{\frac{E}{N}|N_i| - g, 0\right\} \quad (16)$$

Without loss of generality, we assume that the Bad user's neighbors are labeled in increasing order of t_i. So, $t_1 < t_2 < \ldots < t_k$. By choosing $q_0 = t_k$, the Bad user guarantees that all his neighbors will be playing C. The payoff that he receives by playing $q_0 = t_k$ can be calculated using Eq. (14) to be equal to $R_0(t_k|B) = k(E - Nt_k)$. In general, the following holds:

$$R_0(t_j|B) = j(E - Nt_j). \quad (17)$$

It does not make sense for the Bad user to choose any value for q_0 other than one of the t_j, since it would not make any difference to the Good users; it would only reduce the payoff of the Bad user. So, the aim of the Bad user is to find the value of t_j that maximizes the payoff. Note that as j increases, the first term of the product increases, but the second term decreases. This gives a bound on the acceptable values that q_0 can take. It cannot be higher than $\frac{E}{N}$, since that would make the payoff negative, when the Bad user can always guarantee a payoff of at least 0 by choosing $q_0 = 0$. In order to find the optimal t_j we compare the payoffs for all values of j, and pick the maximum. For two values of q_0, say t_l and t_m with $l < m$, the comparison boils down to:

$$\begin{aligned} R_0(t_l|B) > R_0(t_m|B) &\Leftrightarrow l(E - Nt_l) > m(E - Nt_m) \\ &\Leftrightarrow t_m > \frac{l}{m}t_l + (1 - \frac{l}{m})\frac{E}{N} \end{aligned} \quad (18)$$

When the Bad user chooses a value for q_0, some of his neighbors will play C and some D. The ones who play D may cause their own neighbors to start playing D, and so on, in a spirit similar to the algorithm described in the previous section. However, the Ds cannot, by propagating, influence other neighbors of the Bad user: That is a consequence of the tree topology that we have assumed, since the only path that goes between two one-hop neighbors of the Bad user, goes through the Bad user.

What happens when there are multiple Bad users in a general topology? We will examine the circumstances under which the maximization of the total sum of Bad users payoff is achieved through the local maximization of each Bad user's payoff. This local maximization is done as we have just described in Eq. (18). We call "Uncoupled Case" the situation described by these circumstances.

We will find it useful to define the notion of the *tolerance* of a Good user.

Definition 2: The *tolerance* of a Good user is the largest number of his one-hop neighbors that can play D, before he starts playing D himself.

The tolerance of a Good users is a function of $\frac{E}{N}$. To compute the tolerance of user i, assume that n of his neighbors play D, and $|N_i| - n$ play C. From Eq. (13), for user i to play C the following needs to hold:

$$|N_i| - n \geq |N_i|\frac{E}{N} \Leftrightarrow n \leq |N_i|(1 - \frac{E}{N}) \quad (19)$$

The tolerance is the largest integer n for which this equation holds, i.e., $n_{\max} = \lfloor |N_i|(1 - \frac{E}{N}) \rfloor$.

To show that local maximization is equivalent to global maximization, we need to make sure that Good players who start playing D do not cause, recursively, "too many" other Good users to play D so that the payoffs of other Bad users are affected. Going back to the algorithm EQUILIBRIUM, we can see that this will happen if and only if the nodes that play D because of a Bad user B_1 are separated by at least two nodes (three hops) from the nodes that play D because of any other Bad user. In other words, there needs to be a layer of nodes at least two nodes deep that have large enough tolerances so that they will not start playing D themselves. Since, for a fixed $\frac{E}{N}$, the tolerance of a user depends only on the number of his neighbors, nodes with a high degree that are connected to each other would provide the highest resistance to playing D. In graph theoretical terms, the greatest "aggregate" tolerance is achieved, for a given number of nodes, when the nodes are connected in a clique (maximum connectivity).

We now apply these considerations to a specific example, shown in Fig. 4. There are two Bad users, labeled B_1 and B_2, and eleven Good users. Shown in the picture are the thresholds

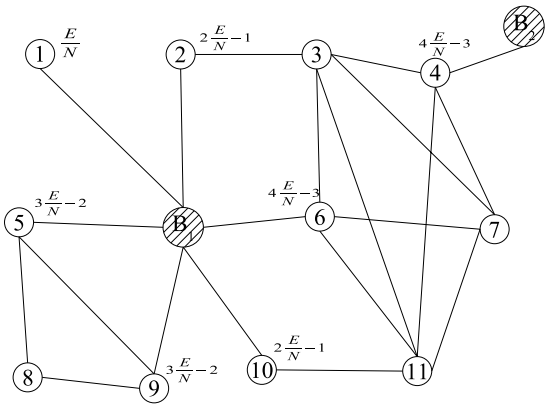

Fig. 4. The Uncoupled Case. Shown next to each node is its threshold: The frequency of Cs it expects to see from the neighboring Bad user. It is equal to $\frac{E}{N}|N_i| - (|N_i| - 1)$, since each Good user happens to have at most one Bad neighbor.

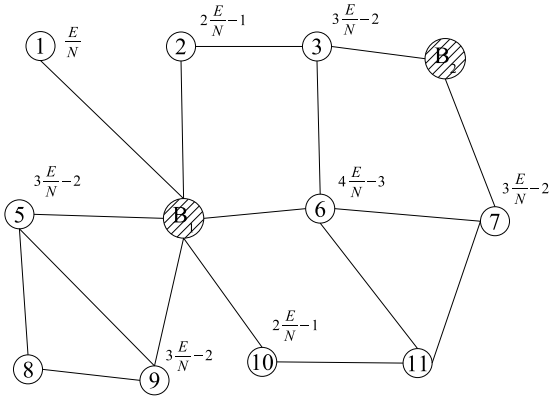

Fig. 5. The General Case. Shown next to each node is its threshold: The frequency of Cs it expects to see from the neighboring Bad user. It is equal to $\frac{E}{N}|N_i| - (|N_i| - 1)$, since each Good user happens to have at most one Bad neighbor.

of the Good users who are neighbors of a Bad user. Remember that the threshold of a Good user is the frequency of Cs that the Bad neighbor should play if the Good user is to play C, assuming that all the other neighbors of the Good user play C.

The thresholds of B_1's neighbors are, in ascending order: $4\frac{E}{N}-3, 3\frac{E}{N}-2, 2\frac{E}{N}-1, \frac{E}{N}$. These, along with the value 0, are the options that B_1 has for choosing his q_{B_1}. To keep things simple, we assume that $\frac{E}{N}$ is such that all the above thresholds are positive. In particular, this translates to $\frac{E}{N} > \frac{3}{4}$.

The table below shows the choices that B_1 can make and the associated payoffs. The thresholds are computed using Eq. (16), and the payoffs using Eq. (14).

frequency	payoff
0	0
$4\frac{E}{N}-3$	$3(N-E)$
$3\frac{E}{N}-2$	$6(N-E)$
$2\frac{E}{N}-1$	$5(N-E)$
$\frac{E}{N}$	0

So, to maximize his own payoff, B_1 should play C with frequency $3\frac{E}{N}-2$. On the other hand, the maximizing frequency for B_2 is $4\frac{E}{N}-3$, which brings him a payoff of $3(N-E)$.

How will the Good users react to these choices? In effect, we will now go through what the algorithm EQUILIBRIUM in Fig. 2 would do. Users 5,6, and 9 will play C, and users 1,2, and 10 will play D because B_1's choice is above or below their thresholds, respectively. User 4 will play C because of B_2's choice.

For the two-hop neighbors of the Bad users, we will have the following: User 8 will obviously play C, since both his neighbors (5 and 9) play C. For users 3 and 11, the situation is identical: They have five neighbors each (hence they need to see a sum of frequencies equal to $5\frac{E}{N}$ to play C), and exactly one of their neighbors is playing D (2 and 10, respectively). So, each sees a total sum of frequencies equal to 4, and they will play C if $4 > 5\frac{E}{N} \Leftrightarrow \frac{E}{N} < \frac{4}{5}$. Although we have constrained $\frac{E}{N}$ to be greater than $\frac{3}{4}$, there is still a range of values for $\frac{E}{N}$, namely $\frac{3}{4} < \frac{E}{N} \leq \frac{4}{5}$, for which users 3 and 11 will play C. The only remaining user is user 7, who will obviously play C, since all his neighbors are playing C.

So, to recapitulate, we have verified that the individual maximization of each Bad user's payoff results, in this case, in the global maximization of the sum of the payoffs of the Bad users.

B. The General Case and a Heuristic Solution

In this section, with the help of Fig. 5, we will see that local maximization of Bad user payoffs does not always correspond to maximization of the sum of their payoffs. This case particularly applies when a Good user has two or more Bad neighbors.

The situation for Bad user B_1 has not changed compared to Fig. 4. The maximizing frequency is still $3\frac{E}{N}-2$, and the payoff is still $6(N-E)$. As far as B_2 is concerned, we can see that the maximizing frequency is also $3\frac{E}{N}-2$, which will bring him a payoff of $4(N-E)$. However, if we apply the algorithm of Fig. 2, we will find that if B_1 sets $q_{B_1} = 3\frac{E}{N}-2$ and B_2 sets $q_{B_2} = 3\frac{E}{N}-2$, then things will not go "as planned" by the Bad users. Namely, user 2 will start playing D, because his threshold is not satisfied. As a result, user 3 will see a total sum of frequencies equal to $0+1+3\frac{E}{N}-2 = 3\frac{E}{N}-1$, which is lower than the total sum of $3\frac{E}{N}$ that he expects, and so he will start playing D. Therefore, B_2's payoff will be diminished. If we carry the calculations further, we will see that, after 3 starts playing D, neither user 6 sees the total sum of frequencies he needs to keep playing C – he only sees $0+1+1+3\frac{E}{N}-2 = 3\frac{E}{N} < 4\frac{E}{N}$. So user 6 will also start playing D, which will diminish B_1's payoff, too. Eventually, all Good users except 5,8, and 9 will play D.

One could argue that a Malicious User would be happy with a situation like that, i.e., when most of the Good users are

playing D. In that case, the network ceases to operate, which is supposed to be what a Malicious User wants. However, by playing D, the Good users do not use up their energy. Remember that another aim of the Bad users is to deplete the energy of the Good ones. Although we have not incorporated this in our model, we could set a finite limit on the amount of energy that the Good users can use. Then, it makes more direct sense for the Bad users to try and deplete the Good users energy, which seems to be infinite in our current model. Moreover, we could constrain the life of the Bad users to some finite (or probabilistic) length of time (after which they get caught, for example, and the network operates smoothly). Then, again it would make more direct sense for the Bad users to "hurry up" and waste the energy of the Good users, so that the network will not be able to operate for very long, even after the Bad users are caught. Even in a non-wireless network, where energy is not a concern, Bad users would still try to exploit the cost of cooperation. Remember that the cost of cooperation is a fundamental characteristic of the situations we are modeling.

We now look at what the Bad users should do, in order to maximize the sum of their payoffs. We keep referring to Fig. 5. In general, either one of two options is likely to achieve the optimal sum: Either one of the two nodes sacrifices his own payoff to a small or large extent, so that the other one can keep playing his maximizing frequency, or they both sacrifice a part of their individually maximum payoffs, in such a way that the sum of payoffs is maximized.

We observe that the Bad user B_2 has two choices for his frequency: either 0, or $3\frac{E}{N}-2$, which do not coincide since $3\frac{E}{N}-2 > 3\frac{3}{4}-2 > 0$ (we keep assuming, as previously, that $\frac{E}{N} > \frac{3}{4}$). If B_2 plays 0, then both his neighbors (3 and 7) will start playing D, and so B_2 will get a zero payoff. As a result, user 2 will expect a total sum of frequencies equal to $2\frac{E}{N}$ from B_1, which is more than the absolute maximum frequency of $\frac{E}{N}$ that a Bad user needs to play in order to keep his payoff positive. So, user 2 would also play D. Moreover, because of users 3 and 7 playing D, user 6 would need a larger frequency of Cs from B_1. We could continue this analysis, but it has become clear that the Bad users are worse off with this choice of B_2.

If B_2 chooses $3\frac{E}{N}-2$, then both 3 and 7 will play C, if they are not affected by the choice of B_1. The payoff for B_2 would then be $4(N-E)$. We have already seen that if B_1 chooses his maximizing frequency $3\frac{E}{N}-2$, then many users will end up playing D, thus ruining the payoffs of the Bad users. So, we turn to the next best local option for B_1, the frequency $2\frac{E}{N}-1$, which gives him a payoff of $5(N-E)$. We see that this keeps all neighbors of B_1 at C except user 1. Therefore, the rest of the network is not affected, and everybody keeps playing C. The sum of payoffs for the two Bad users is now $5(N-E)+4(N-E) = 9(N-E)$, which is the maximum sum that can be achieved (remember that the sum of the two individual maximum payoffs is $6(N-E)+4(N-E) = 10(N-E)$, and we have seen that this payoff cannot be achieved in a stable equilibrium).

We now propose a heuristic for finding a payoff as close as possible to the maximum sum of payoffs. The heuristic maximizes the payoff of the "most promising" Bad users first. In other words, the idea is to find the node whose maximum individual payoff is the largest among all individual payoffs and let him play his maximizing frequency. Then, re-evaluate the individual payoffs attainable by all other Bad users in the new situation, and again choose the "most promising" one. Finally, repeat this process until no other Bad users are left. In effect, this sacrifices the payoffs of the "least promising" Bad users for the sake of the "most promising" ones. Note that in the uncoupled case, this heuristic performs optimally.

In the topology depicted in Fig. 5, our heuristic would choose B_1 as the most promising Bad user, since his maximum individual payoff is $6(N-E)$, for a maximizing frequency of $3\frac{E}{N}-2$. That would cause Good users 1, 2, and 10 to play D because their thresholds would not be satisfied. Now, we come to what B_2 can do in the new situation. as we have already seen, he cannot play his individual maximizing frequency. The heuristic would make B_2 play whatever frequency is necessary to prevent the Ds from spreading to other Good users. For instance, to prevent user 3 from playing D, B_2 will need to play the maximum possible frequency ($q_{B_2} = 1$), despite the fact that B_2 would incur a negative payoff by doing that. Then, user 3 would see a total sum of frequencies of 2, and, if his expected sum of frequencies, $3\frac{E}{N}$, is lower than that, then he would play C. So, we see that the performance of the heuristic depends on the value of $\frac{E}{N}$. It can be shown that the lower that value is, the better the heuristic performs. If $\frac{E}{N}$ is low enough, we get the uncoupled case, for which we know that the heuristic coincides with the optimal solution.

IV. Conclusion and Future Work

Unstructured networks, like wireless ad-hoc or peer-to-peer networks, depend on the cooperation of their users to operate successfully. However, users have incentives and disincentives to cooperate, both of which we model in a game theoretic fashion, with appropriate payoffs (N and E). The main contribution in this paper is the modeling of Malicious Users in the same game theoretic framework by assigning payoffs to them, and not just modeling them as, e.g., "Always Defect". We consider a repeated game to emphasize the duration of the network operation, and we model the Good user strategy to follow fictitious play, versions of which have appeared in previous work on trust and reputation calculations.

In the future, we plan to elaborate on how the topology and the relative values of the parameters $\frac{E}{N}$ affect the optimal equilibrium payoffs for the users. We also want to make a more rigorous and general evaluation of the performance of the heuristic, compared to the optimal solution. Two more general extensions worth pursuing are: First, enable the users to play different actions against different neighbors. Second, associate an actual cost with the Bad users' attempt to attack the network. This second extension, suggested by one of the reviewers, seems to require a genuine change in the modeling of the Bad users, since in the current model the Bad users do

not incur any direct cost. More important, any combination of actions could be called an attack, so it is not clear when the Bad users would have to pay the extra cost and when not.

ACKNOWLEDGMENTS

This work is prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. Research is also supported by the U.S. Army Research Office under Award No DAAD 190110494. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Army Research Laboratory or the U.S. Army Research Office.

The authors would like to thank the anonymous reviewers for their constructive comments and useful suggestions.

REFERENCES

[1] G. W. Brown, "Iterative solution of games by fictitious play," in *Activity Analysis of Production and Allocation*, T. Koopmans, Ed. New York: John Wiley and Sons, 1951, pp. 374–376.

[2] A. Blanc, Y.-K. Liu, and A. Vahdat, "Designing incentives for peer-to-peer routing," in *Proc. of IEEE Infocom*, Miami, FL, March 2005.

[3] L. Buttyan and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 8, no. 5, October 2003.

[4] M. Félegyházi, J.-P. Hubaux, and L. Buttyán, "Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 463–476, May 2006.

[5] A. Urpi, M. Bonuccelli, and S. Giordano, "Modelling Cooperation in Mobile Ad Hoc Networks: A Formal Description of Selfishness," in *Proc. of WiOpt'03:Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, INRIA Sophia-Antipolis, France, March 2003.

[6] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," in *Proc. of IEEE Infocom*, San Francisco, CA, April 2003.

[7] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, no. 4489, pp. 1390–1396, March 2002.

[8] E. Altman, A. Kherani, P. Michiardi, and R. Molva, "Non-cooperative Forwarding in Ad Hoc Networks," INRIA, Tech. Rep. RR-5116, 2004.

[9] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining cooperation in multi-hop wireless networks," in *Proc. of the 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, May 2005.

[10] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," in *Proc. of the 3rd Annual Workshop on Economics and Information Security (WEIS)*, Minneapolis, MN, May 2004.

[11] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Proc. of the 1st International Workshop on Peer-To-Peer Systems, (IPTPS)*, Cambridge, MA, March 2002.

[12] N. Daswani, H. Garcia-Molina, and B. Yang, "Open problems in data-sharing peer-to-peer systems," in *Proc. of the 9th International Conference on Database Theory (ICDT)*, Siena, Italy, January 2003.

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

[14] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[15] S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for peer-to-peer and mobile ad hoc networks," in *Proc. of the 2nd Workshop on the Economics of Peer-to-Peer Systems (P2PEcon)*, Cambridge, MA, June 2004.

[16] R. Ismail and A. Jøsang, "The beta reputation system," in *Proc. of the 15th Bled Electronic Commerce Conference*, Bled, Slovenia, June 2002.