Enhancing Benign User Cooperation in the Presence of Malicious Adversaries in Ad Hoc Networks

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

***Abstract*— Decentralized and unstructured networks are becoming more prevalent today (e.g. ad hoc networks). Like every network, they depend on the cooperation of their users to survive. However, each user does not necessarily know who the others are, or what their intentions are. Since there is no centralized infrastructure, the users can only base their decision on what they observe themselves. Ideally, they would like to cooperate only with users that have common interests with them.**

In this paper, we use a game theoretic model for the above situation. We assume there are only two kinds of users, Good (benign) and Bad (malicious). Good users receive a high game theoretic payoff when they cooperate with other Good users, but a low payoff when they cooperate with Bad users. We propose behavior rules (strategies) to achieve equilibria that enable as many Good users as possible to cooperate with each other, and at the same time minimize the number of Good-Bad cooperations.

I. Introduction

In ad-hoc networks all nodes are equivalent, in the sense that they play the role of both user and router: they create their own data traffic, but they also forward the traffic of other nodes. However, for the network to operate successfully (i.e. to deliver the traffic to its intended destination) the nodes need to cooperate. Seen from a selfish viewpoint, a node has no a priori incentive to cooperate. Actually, in a resource constrained network, as ad-hoc networks are envisioned to be, a node may have a serious counter incentive to cooperate. That is, by dropping other users' traffic, a node saves a considerable amount of energy.

This is the model that most of the literature has focused on, and many methods for providing incentives – or in any way enforcing cooperation – have been proposed. However, some users may actually be malicious. They are not interested in preserving their own resources. They are merely interested in destroying the operation of the network, by whatever means possible. They may be eavesdroppers who monitor traffic and try to learn private information, or they may want to inject malformed packets in the network (worms, etc.). In this case, a different approach is called for, since there is no incentive that would entice them to cooperate.

In this work, we are proposing a simple model for the interaction of Good (benign) and Bad (malicious) users in a network. Our model is based on game theory, and allows users (players in game theory) to choose one of two possible actions: Cooperate or Defect. The general scenario that we want to capture is the following: Good users want to cooperate with other Good users, but not with Bad users. Bad users, on the other hand, want to cooperate with Good users. The Good are unaware of who is Good and who is Bad, but since the game is played repeatedly they can gradually detect the Bad ones. We will explore strategies that the Good users can follow to detect and isolate the Bad ones.

Note that we will not be assuming collusion among the Bad users, although this can be an extension of our model. Also, our model for the Bad users means that they benefit from staying hidden and cooperating for as long as possible without getting caught. So, we do not cover situations where a single cooperation between a Bad and a Good user is enough, e.g., to destroy the whole network.

The rest of the paper is organized as follows: In the next section, we briefly present the work done on enforcing cooperation in ad-hoc networks. In section III, we give a detailed presentation of our model of the network and the specifics of the game played. After that, we analyze in section IV the strategies and corresponding outcomes that can appear in the game. Section V shows some indicative simulations and compares to the analytical approach of section IV, and section VII concludes.

II. Related Work

A growing body of literature, a comprehensive overview of which is in [1], deals with circumstances under which the cooperation between nodes can be sustained. A model used often in this literature (but also more generally in power control in ad hoc networks [2]) is a game theoretical representation of the users in the network (an exception is [3]). The players in game theory attempt to maximize an objective function which takes the form of a payoff. Users make choices and each user's payoff depends not only on his own choice, but also on those of the other users. Hence, in the wireless network context, a user's payoff depends not only on whether he decides to cooperate (by transmitting other users' data) or not, but also on whether his neighbors will decide to cooperate.

1-4244-0423-1/06/$20.00 ©2006 IEEE

The literature is not considering malicious users, only selfish, and there is no degree of selfishness that can approximate the payoffs of our Bad users. For example, Félegyházi, Hubaux and Buttyán [1] 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 [4] does not apply (as the authors themselves point out). Other relevant work is [5], [6], [7].

III. System Model

The network is modeled as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The nodes represent the users of the network. Each user can be of Good or Bad type, and the sets of Good and Bad users are, respectively, $V_G \subset V$ and $V_B \subset V$, $V_G \cup V_B = V$, $V_G \cap V_B = \emptyset$. An edge (i, j) means that users i and j can communicate; i is then said to be a neighbor of j ($i \in N_j$), and vice versa ($j \in N_i$). Also, the edges are weighted: The weight J_{ij} of an edge is $+1$ if both i and j are Good, and -1 if one is Good and the other is Bad. We assume there are no links between Bad users. Good users do not know who is Good and who is Bad, but Bad users do. Good users do not even know how many Bad users there are.

The network operates in rounds $t = 1, 2, \ldots$, and at each round t each user i chooses an action a_i^t: C (for Cooperate) or D (for Defect). Playing C corresponds to making oneself available for communication (e.g. sending/receiving data). Playing D corresponds to shutting down all communications to and from the user. After all users have chosen an action, each user learns his neighbors' actions (i.e. which neighbor played which action), and his own payoff for that round, which depends only on his own action and these of his neighbors. Note that a user's payoff is known only to him, and is never reported to others. If a Good user is able to tell that a particular neighbor of his is Bad, then he can sever the link that joins them, so as not to be affected by that neighbor's actions in the future. In Section IV, we will discuss how Good users can detect Bad ones.

At round t, the payoff v_i^t of a Good user i who played C equals the number of Good neighbors who played C minus the number of Bad neighbors who played C (for convenience, $a_i^t = 1$ for C, and 0 for D):

$$v_i^t = \sum_{j \in N_i} J_{ij} a_i^t a_j^t. \tag{1}$$

This reflects the preference of Good nodes to cooperate with other Good nodes and not with Bad ones. A Good user who played D receives a zero payoff regardless of the actions of the neighbors. This means that he risks no losses, but he has no gain, he learns nothing about his neighbors, and his neighbors learn nothing about him. The payoff of a Bad user who plays C is equal to the number of his Good neighbors who played C (remember that a Bad user has only Good neighbors). So, it is the negative of v_i^t.

IV. Analysis

To simplify the analysis, we will concentrate on a star topology network, where the central node is a Good user, and his neighbors are N Good users and 1 Bad (Fig. 1). We assume that the central node knows that he has exactly one Bad neighbor, but he does not know who that is. Note that the star topology assumption is not unrealistic, since this is exactly the situation (from the point of view of the central node) even in a general network. We will also see that the assumption that only one Bad user exists can be removed without significant conceptual change in the analysis.

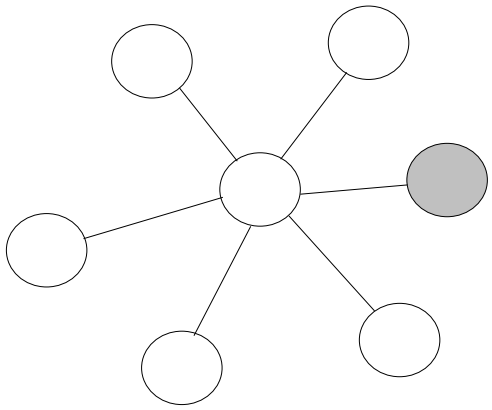

Fig. 1. Star topology. Bad user is shaded.

Note from the System Model discussion that Good users only learn the total payoff they receive after each round, and not the per-link payoffs they receive due to their interactions with individual neighbors. So, they cannot immediately tell which of their neighbors are Good and which are Bad, but they do get some information about the types of their neighbors. In what follows, we will describe strategies for the Good and Bad players that form a Nash equilibrium. For the most part, we will be seeing things either from the central Good user's point of view, or from the point of view of the Bad user. Since in a general network all Good nodes will see themselves in the role of a central node in a (local) star topology, we are looking for strategies that are symmetrical with respect to Good nodes. That is, we want all the Good nodes to follow the same rules when choosing what to play.

Assume that the central Good user i has memory of the past history (own and neighbor moves, as well as received payoffs). Let CN_i^t (resp. DN_i^t) be the subset of i's neighbors that play C (resp. D) at round t. We assume that i plays C at round t, so i's payoff at round t is $|CN_i^t|$ if the Bad user played D, or $|CN_i^t| - 2$ if the Bad user played C (Remember that a C from a Good user gives $+1$, whereas from a Bad user it gives -1.). So, just by looking at his payoff, the central Good user i can deduce whether the Bad user played C or D at round t. The Bad user is then known to be either in the set CN_i^t or

in DN_i^t. Without loss of generality, let's assume that the Bad user played C.

In the next round $(t+1)$, if the Bad user plays C again, then i can deduce that he is in the intersection $CN_i^t \cap CN_i^{t+1}$. If he plays D, then he is in $CN_i^t \cap DN_i^{t+1}$. This sequence of sets (the sequence of *hiding sets*, the initial of which is N_i) is non-increasing, but the Bad user will only be detected if it converges to a singleton set. If the behavior of the Good users is deterministic, then the Bad user can imitate a Good user, and he will never be discovered. However, if the Good nodes choose their actions in a randomized manner, they are no longer predictable.

We will look at the simplest possible randomization: each Good user plays C with probability p independently at each round. We want to compute the probability p that maximizes the central Good user's payoff. Given an infinite sequence of round payoffs $\{v_i^t, t = 1, 2, \ldots\}$, the game payoff for user i is $v_i = (1-\delta)\sum_{t=1}^{\infty} \delta^{t-1} v_i^t$. The parameter $\delta, 0 < \delta < 1$, signifies the relative importance of current payoffs compared to later payoffs. In game theoretic literature (e.g. [8]) it is called *discount factor* and is often associated with the patience of a player. For example, if $\delta \to 0$, then only a few initial round payoffs practically dominate the game payoff. On the contrary, when $\delta \to 1$ all rounds are equivalent. In our case, δ could correspond to how long the Good user thinks that the network will keep operating. To be precise, δ could be seen as the probability that the network will collapse at time $t+1$, given that it has been operating up to and including time t.

To compute the payoff, we split the network evolution into two stages: pre-detection and post-detection of the Bad user. The Good users start by playing C with probability p, and we assume that the Bad user always plays C. We will see that this is the best that the Bad user can do. The expected per round payoff for the central Good user i is

$$\mathrm{E}[v_i^t] = \Pr\{a_i^t = C\} \sum_{j \in N_i} J_{ij} \Pr\{a_j^t = C\} = p(pN-1). \quad (2)$$

After the Bad user has been detected, the link to him is severed and the Good nodes are free to play C forever. So, the central Good user's payoff is N from then on ($+1$ from each one of the N Good neighbors). Assuming that the Bad user is caught after the actions of round t_0, the total game payoff for i is

$$v_i(\delta, p, N) = (1-\delta)\{\sum_{t=1}^{t_0} \delta^{t-1} p(pN-1) + \sum_{t=t_0+1}^{\infty} \delta^{t-1} N\} \quad (3a)$$

$$= (1-\delta^{t_0})p(pN-1) + \delta^{t_0} N \quad (3b)$$

$$= \delta^{t_0}(N - p(pN-1)) + p(pN-1) \quad (3c)$$

We will now calculate the expected value of t_0 with the following argument. Since the Bad user is always playing C, his hiding set after t rounds is $N_i \cap CN_i^1 \cap CN_i^2 \cap \cdots \cap CN_i^t$. Since the Good users are playing C with probability p at every round, the size of the hiding set is reduced by a factor of p every round. Since t_0 is the detection time, it has to be that

$$p^{t_0}(N+1) = 1 \Rightarrow t_0 = -\log_p(N+1). \quad (4)$$

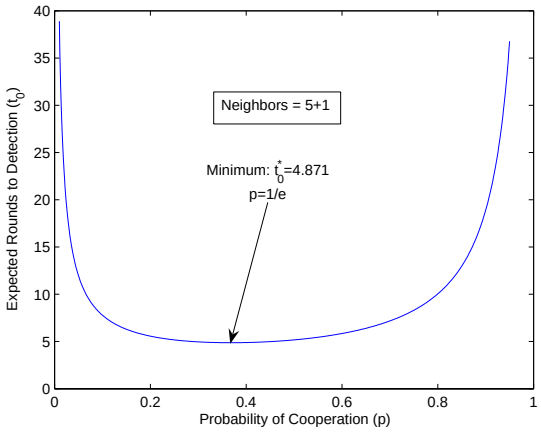

Fig. 2. Expected number of rounds until detection.

In this calculation, however, we have neglected the fact that the central Good user also plays C with probability p. When he plays D he will not be able to make any observations about his neighboring nodes, so the D-rounds will be wasted as far as detection is concerned. The above calculated t_0 is the number of *observations* that are needed to detect the Bad user. A C will be played once every $\frac{1}{p}$ rounds, so the correct expected time of detection will be

$$t_0(p) = \frac{1}{p} \cdot (-\log_p(N+1)) = -\frac{\log_p(N+1)}{p}, \quad (5)$$

shown in Fig. 2 for $N = 5$. We can calculate that the expected number of rounds to detection is minimized for $p = \frac{1}{e}$. It is then equal to $t_0^* = e\ln(N+1)$.

Looking at Eq. (3b), we observe that $p(pN-1)$ (the pre-detection payoff) is always smaller than N (the post-detection payoff). This seems to imply that the Good user's payoff is maximized when δ^{t_0} is maximized, i.e. when t_0 is minimized. But this is not the case, although it is not easy to see analytically. The idea is that when δ is small enough, δ^{t_0} is very close to zero even for $t_0 = t_0^*$. As a result, v_i is practically equal to $p(pN-1)$, and it is maximized when p is maximized, i.e. for $p \to 1$.

To study the behavior of $v_i(p)$ for different values of the discount factor δ we perform numerical computations. First of all, the payoff increases monotonically with δ. This is because, regardless of the value of p, being more patient gives more weight to the post-detection payoff, which is higher (N versus $p(pN-1)$). We can also see that the Good users can guarantee a payoff of $N-1$ (4 in this case) by choosing p very close to 1. Even though the pre-detection period can become very large, the pre-detection payoff is close to $N-1$. But we will see that in some cases, the Good can do better than that.

For a fixed value of δ the effect of an increase of p is twofold: First, the payoff during the pre-detection period is increased. Second, if p increases from 0 to $\frac{1}{e}$, then the number of rounds until detection (the length of the pre-detection period) decreases. But if p increases beyond $\frac{1}{e}$, the length of

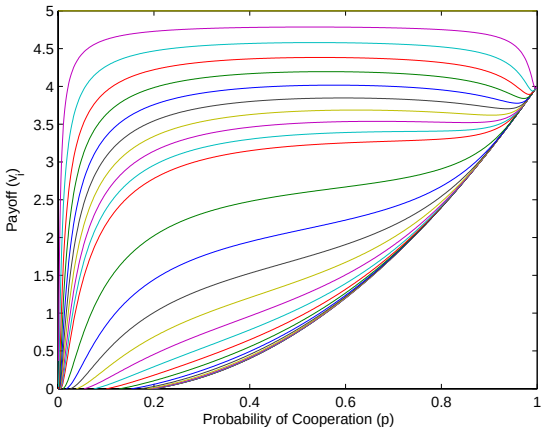

Fig. 3. All values of δ: $0 \leq \delta \leq 0.9$, step 0.05; $0.9 \leq \delta \leq 1$, step 0.01.

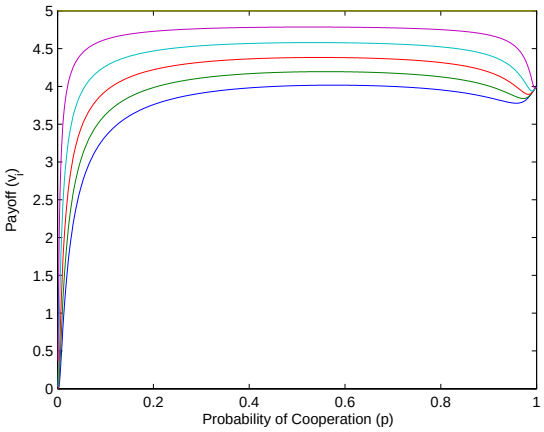

Fig. 5. Large values of δ: $0.95 \leq \delta \leq 1$, step 0.01.

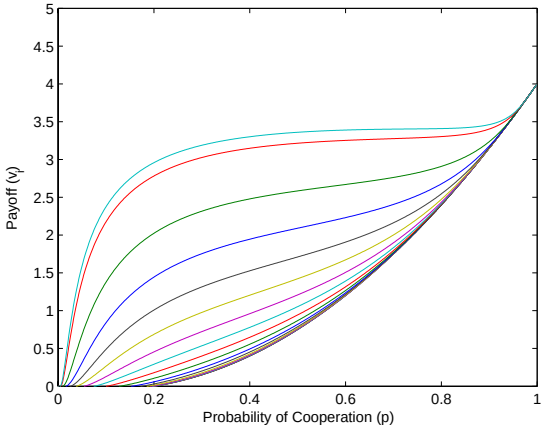

Fig. 4. Small values of δ: $0 \leq \delta \leq 0.9$, step 0.05; 0.91.

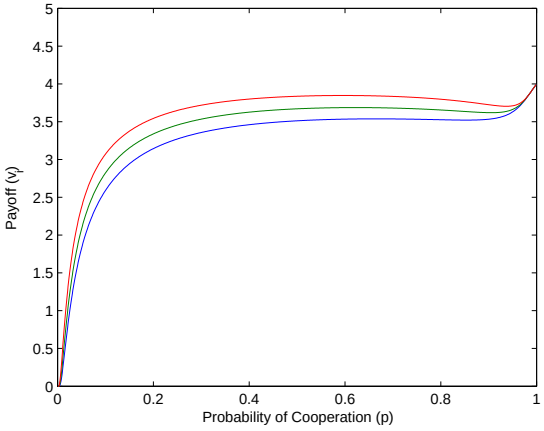

Fig. 6. Medium values of δ: 0.92, 0.93, 0.94.

the pre-detection period increases.

It seems there are three distinct types of behavior for different intervals of δ (Fig. 3). When δ is small, the payoff increases monotonically with p (Fig. 4). This happens because a small δ implies that $\delta^{t_0} \approx 0$, and $v_i \approx p(pN-1)$ which increases monotonically with p. So, if the Good user is very impatient, then the post-detection payoff is too heavily discounted to make any difference, even if the Bad user is detected in the shortest number of rounds ($t_0^* = e\ln(N+1)$). All the Good can do in this case is increase the pre-detection payoff as much as possible by setting p to 1.

When δ is large (Fig. 5), δ^{t_0} is approximately equal to 1, so $v_i \approx N$ (the post-detection payoff, which is the maximum possible value of the total payoff) regardless of p. A value of δ close to 1 means that the Good users don't mind waiting; so, provided that the detection happens in finite time, the payoff is very close to N. This is reflected in Fig. 5 where the payoff is almost constant for a wide range of values of p, and the maximum is attained for values of p that are around 0.5 and definitely away from 1. However, when p is too close to 0 then the detection time is too large, and the pre-detection payoff is too small, so the total payoff is close to 0. When p approaches 1, the detection time is again too large (hence the observed drop in post-detection payoff). But now the pre-detection payoff is close to its maximum value of $N-1$, so the total payoff goes up again and becomes equal to $N-1$ for $p=1$.

When δ is medium (Fig. 6), the maximum payoff is again attained for $p=1$, as in the small δ case. However, unlike the small δ case, the payoff does not increase monotonically with p. There is a local maximum for values of p around 0.5, which is caused by the same reasons as in the large δ case, although not as pronounced.

The conclusion is that, for a given value of δ, the maximum payoff is achieved either for a probability of cooperation p equal to 1, or for some value around 0.5. The first case happens when the Good users are impatient (small δ), and so the post-detection period is too far into the future for them to care. Which means that all their gains are going to come from the pre-detection period, therefore the best policy for them is to maximize the pre-detection period payoff $p(pN-1)$. The second case happens when δ is large enough

for the post-detection payoff to be weighted more heavily than the pre-detection payoff. The maximum around 0.5 is happening because the post-detection payoff is maximized for $p = \frac{1}{e} \approx 0.37$ (the value of p that maximizes δ^{t_0}), but does not drop significantly for somewhat larger values of p. The pre-detection payoff is, however, monotonically increasing with p, and until around $p = 0.5$ it more than compensates for the slight drop in the post-detection payoff. After that, the post-detection payoff starts dropping faster than the pre-detection can increase.

Why does the Bad user have to play C all the time? If he plays what most Good play, he prolongs the time of detection. If he plays C, he gains payoff. If p was chosen by the Good to be larger than $\frac{1}{2}$, then these two considerations of the Bad user would both concur to playing C. However, the maximizing p may be less than $\frac{1}{2}$ for some values of δ, so it might make sense for the Bad user to play D once in a while so as to hide a bit longer.

We will see that playing D never increases the Bad user's payoff, and it can even decrease it. Suppose the current hiding set is X, and the Bad plays D. If the central Good plays D, nothing changes. If the central Good plays C, he observes who plays D and who plays C, so by looking at the payoff he can tell what the Bad played. The new hiding set is $X \cap Y \subseteq X$, i.e. smaller than X, the Bad has gained nothing in the current period, and because of the discount factor δ the payoff of a C has become smaller. So, in effect, the Bad player is facing the exact same situation he was facing before, only he is in a smaller hiding set, and the benefit of a C is smaller. This allows us to conclude that the best thing the Bad can do is play C from the first round until he is detected. Hence, the strategies "always C" for the Bad and "C with probability p" (for the maximizing value of p) for the Good form a Nash equilibrium. Noone can do better by unilateral deviation.

V. Simulation for the star topology

Another observation is that we have assumed that the detection is happening at exactly the expected time. This translates to an incorrect computation of the expected payoff v_i. Using equation (3c) to calculate the payoff ($v_i(t)$) for a time of detection equal to t, the expected payoff over all possible times of detection (but for a fixed value of p) is

$$\mathrm{E}[v_i(t)] = \mathrm{E}[\delta^t](N - p(pN-1)) + p(pN-1). \quad (6)$$

What we have computed, however, is

$$\delta^{\mathrm{E}[t]}(N - p(pN-1)) + p(pN-1) = \quad (7)$$
$$\delta^{t_0}(N - p(pN-1)) + p(pN-1). \quad (8)$$

To estimate how far from the truth we are, we performed some indicative simulations. Shown in figure 7 are the comparisons of the computations to the simulations for three values of $\delta : \delta = 0.2, 0.91, 0.97$. These were chosen as representative values from each one of the three intervals of δ where the behavior of the payoff function changes. Each simulation was repeated 100 times and the mean and variance are depicted in the diagram for each value of p that the simulation was done. We see that, except for the $\delta = 0.2$ case, there is a disparity not accounted for by the variance interval. On the other hand, qualitatively the simulations are similar to the computations. For $\delta = 0.2$ they are almost identical, for $\delta = 0.91$ the simulation curve is monotonically increasing achieving its maximum at $p = 1$ (similarly to the computation), and for $\delta = 0.97$ the maximum is attained long before $p = 1$, after which the payoff drops, just as the computation predicts.

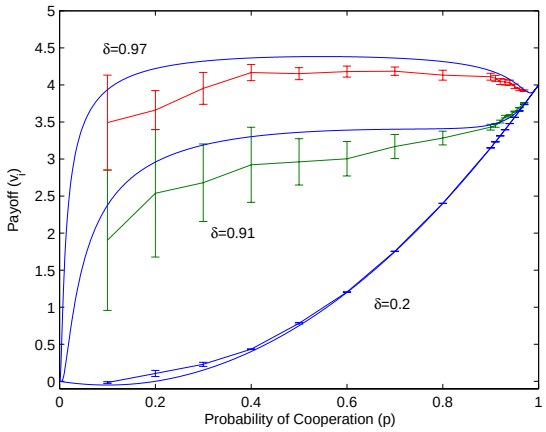

Fig. 7. Comparison: Simulation against computation results for $\delta = 0.2, 0.91, 0.97$

VI. Extensions

The case of multiple Bad users in the neighborhood of a Good user is not much different in essence (there is still the concept of a hiding set), although the calculations grow longer. Even if the Good user does not exactly know how many Bad neighbors he has, he can discover after each round how many Bad users were among the cooperators. Since the set of cooperators changes in every round, the Good user can utilize the information he gathers after each round to eventually pinpoint who the Bad users are. For example, if a set of cooperators has just one extra member compared to another set, then the type of that extra member (Good or Bad) can be immediately determined by comparing the payoffs in the two cases. Of course, this process requires extra memory capabilities on the part of the Good user (to "remember" what happened in the past rounds).

The δ-discounted payoff function is not the only possible choice to calculate the payoff of the repeated game. Two other popular choices in game theoretic literature (for precise definitions, see [8]) are the average payoff ($\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} v_i^t$), and the plain sum of payoffs ($\lim_{T\to\infty} \sum_{t=1}^{T} v_i^t$). Any one of the three could be used for our game. We chose the δ-discounted because the parameter δ can be interpreted as the probability that the network will continue existing for (at least) one more round, given that it has existed so far. More precisely, δ is the subjective estimate of the Good users for that probability. More emphasis is given to upcoming gains than

longer term ones. The average payoff can be approximated in the limit $\delta \to 1$. The plain sum of payoffs would give an infinite post-detection payoff, so, as long as detection happened in finite time, the total payoff would be infinite. In this last case, perhaps it would make more sense for the Good users to try to minimize the payoff of the Bad user, rather than increase their own (since theirs would be infinite in any case).

VII. Conclusion and Future Work

Ad hoc networks depend on the cooperation of their members to operate successfully. In this paper, we considered what happens if some of the network users are outright malicious, so cooperation with them is specifically undesirable. The malicious users are not immediately known for what they are, but they are gradually exposed. We considered different strategies (probabilities p of cooperation), and we discussed which are more desirable (leading to larger payoff), and under which circumstances (values of the discount factor δ).

Our main conclusion is that a lot depends on the patience of the Good users (represented by δ). It may be the case that only continual cooperation is the best solution, thus effectively ignoring the presence of the Bad user. Or, a more refined way of choosing how often to cooperate may be more appropriate.

In the future, we want to precisely compute the expected payoffs of the Good users, to avoid the discrepancy with the simulation results as shown in section V. Also, it is necessary to consider in more detail the case of multiple Bad users, not just one, and additionally allow collusion among them. Finally, more general topologies should be studied having as a basis the simple star topology we considered here.

Acknowledgments

This material is based upon work 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 Office.

The authors would also like to thank the anonymous reviewers for their comments that helped improve the presentation of the material.

References

[1] 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.

[2] Y. Xing and R. Chandramouli, "Distributed discrete power control for bursty transmissions over wireless data networks," in *Proc. of ICC 2004 - IEEE International Conference on Communications*, Paris, France, June 2004, pp. 139–143.

[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, 2003.

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

[5] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," in *Proc. of IEEE Infocom 2003*, San Francisco, March 30-April 3 2003.

[6] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring, "Modelling Incentives for Collaboration in Mobile Ad Hoc Networks," in *Proc. of WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, INRIA Sophia-Antipolis, France, Mar. 2003.

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

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