Secure Cooperation in Autonomous Mobile Ad-Hoc Networks Under Noise and Imperfect Monitoring: A Game-Theoretic Approach

Wei Yu and K. J. Ray Liu

***Abstract*—In autonomous mobile ad-hoc networks, one major challenge is to stimulate cooperation among selfish nodes, especially when some nodes may be malicious. In this paper, we address cooperation stimulation in realistic yet challenging contexts where the environment is noisy and the underlying monitoring is imperfect. We have first explored the underlying reasons why stimulating cooperation under such scenarios is difficult. Instead of trying to enforce all nodes to act fully cooperatively, our goal is to stimulate cooperation in a hostile environment as much as possible through playing conditional altruism. To formally address the problem, we have modeled the interactions among nodes as secure routing and packet forwarding games under noise and imperfect observation, and devised a set of reputation-based attack-resistant cooperation strategies without requiring any tamper-proof hardware or central banking service. The performance of the devised strategies has also been evaluated analytically. The limitations of the game-theoretic approaches and the practicability of the devised strategies have also been investigated through theoretical analysis and extensive simulation studies. The results have demonstrated that although sometimes a gap may exist between the ideal game model and the reality, game-theoretic analysis can still provide thoughtful insights and useful guidelines when designing cooperation strategies.**

***Index Terms*—Cooperation, game theory, mobile ad-hoc network, security.**

I. Introduction

In mobile ad-hoc networks, nodes communicate with others out of their direct transmission range through cooperatively forwarding packets for each other without requiring a fixed network infrastructure. However, in many applications, nodes may belong to different authorities and pursue different goals. Consequently, fully cooperative behavior, such as unconditionally forwarding packets for others, cannot be taken for granted. On the contrary, in order to save limited resources, nodes may tend to be "selfish." In this paper, we refer to such mobile ad-hoc networks as autonomous mobile ad-hoc networks.

Before ad-hoc networks can be successfully deployed in an autonomous way, the issue of cooperation must be resolved first. In the literature, many schemes have been proposed to enforce cooperation in ad-hoc networks [1]–[13], which can be roughly classified into two categories: 1) reputation based (e.g., [2]–[8] and [10]–[12]) and 2) pricing based (e.g., [1], [9], and [13]–[15]). One important observation is that without introducing any pricing mechanisms, in general, it is impossible to enforce all nodes to act fully cooperatively [12], [13]. However, pricing-based mechanisms have the drawback that they require either tamper-proof hardware or a central banking service, which may not always be available in autonomous ad-hoc networks.

In this paper, instead of trying to enforce all nodes to act fully cooperatively, our goal is to stimulate cooperation among selfish nodes as much as possible without relying on any tamper-proof hardware or central banking service. Further, instead of addressing this issue in ideal scenarios, we focus on realistic scenarios where communication channels are error prone, the underlying monitoring is imperfect, and some nodes may be malicious whose goal is to cause damage to the network, which make achieving the aforementioned goal an extremely challenging task.

Like most existing work, we also focus on the most basic networking mechanism in ad-hoc networks, namely packet forwarding. However, in our work, we have jointly considered routing and packet forwarding by modelling the interactions among nodes as multistage secure routing and packet forwarding game under noise and imperfect observation. We have explored the challenges to stimulate cooperation under such realistic settings, and identified the underlying reasons why in many situations cooperation cannot be enforced. Then, we devised a set of reputation-based attack-resistant cooperation strategies without requiring any tamper-proof hardware or central banking service, and evaluated the performance of the devised strategies. When devising cooperation strategies, besides the Nash equilibrium, the issues of fairness, cheat proofness, and robustness to attacks have also been considered. Furthermore, the limitation of the game-theoretic approaches and the practicability of the devised strategies in reality have also been investigated through theoretical analysis and extensive simulation studies. Meanwhile, although our focus is on mobile ad-hoc networks, networks with fixed topology have also been investigated when necessary.

The rest of this paper is organized as follows. In Section II, we provide an overview of the related work. In Section III, we describe the system model, pose the challenges for cooperation stimulation in realistic contexts, and model the interac-

Manuscript received April 8, 2007; revised March 9, 2008. This work was supported in part by the Army Research Office under URI Award No. DAAD19-01-1-0494. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Klara Nahrstedt.

The authors are with the Department of Electrical and Computer Engineering and The Institute for Systems Research University of Maryland, College Park, MD 20742 USA (e-mail: weiy@microsoft.com; kjrliu@isr.umd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2008.922453

tions among nodes as multistage secure routing and packet forwarding game under noise and imperfect observation. The set of devised attack-resistant cooperation stimulation strategies is described in Section IV, and the theoretical analysis of the devised strategy is presented in Section V. Extensive simulations have also been conducted to evaluate the effectiveness of the devised strategies under various scenarios, where the results are summarized in Section VI. Section VII compares our approaches with the existing approaches. Finally, Section VIII concludes this paper.

II. State of the Art

One way to stimulate cooperation among selfish nodes is to use payment-based methods, such as those proposed in [1], [9], and [13]–[15]. For example, a cooperation stimulation approach was proposed in [14] by using a virtual currency called nuglets as payment for packet forwarding, which was then improved in [15] by using credit counters. Both of these schemes require tamper-proof hardware in each node. Another payment-based system, Sprite [1], releases the requirement of tamper-proof hardware, but requires some central banking service trusted by all nodes. In [19], pricing-based truthful and cost-efficient routing protocols for mobile ad-hoc networks were proposed. A similar approach was also presented in [13]. Although these schemes can effectively stimulate cooperation among selfish nodes, the requirement of tamper-proof hardware or central billing service greatly limits their applications.

An alternative way to stimulate cooperation among selfish nodes is to use reputation-based methods with necessary monitoring [2]–[4], [11]. In [2], a reputation-based system was proposed to mitigate nodes' misbehavior, where each node launches a "watchdog" to monitor its neighbors' packet forwarding activities. Following that, Core was proposed to enforce cooperation among selfish nodes [3], and CONFIDANT was proposed to aim at detecting and isolating misbehaving node and thus making it unattractive to deny cooperation [4], and ARCS was proposed to simultaneously stimulate cooperation among selfish nodes and defend against attacks [11]. However, all of these schemes are heuristics. Further, the underlying monitoring mechanisms used by these schemes (e.g., watchdog) may not be robust to various attacks and cheating behavior.

Besides that, progress has also been made toward mathematically analyzing cooperation enforcement in autonomous ad-hoc networks by applying game theory, such as [5]–[8], [10], [12]. In [5], Srinivasan *et al.* provided a mathematical framework for cooperation in ad-hoc networks by focusing on the energy efficiency aspect of cooperation. In [12], Felegyhazi *et al.* defined a game model and identified the conditions under which cooperation strategies can form an equilibrium. In [8], Michiardi and Molva studied the cooperation among selfish nodes in a cooperative game-theoretic framework. In [10], Altman *et al.* studied the packet forwarding problem in a noncooperative game-theoretic framework and provided a simple punishing mechanism considering an end-to-end performance objective of the nodes. The study of selfish behavior in ad-hoc networks has also been addressed in [6] and [7]. All of these schemes focus on selfish behavior and most of them study cooperation enforcement under a repeated game framework.

Our work also falls in the category of reputation-based cooperation stimulation analysis for autonomous ad-hoc networks under a game-theoretic framework. However, there are several major differences which distinguish our work from the existing work. First, we study this problem under more realistic and more challenging scenarios, where the communication medium is error prone, the underlying monitoring mechanism is not perfect, and some nodes may be malicious. Second, instead of enforcing cooperation among nodes, which has been shown to not be achievable in most situations, our goal is to stimulate cooperation among selfish nodes as much as possible. Third, we have identified the reasons why in many situations cooperation cannot be enforced. Furthermore, we have also studied the limitation of game-theoretic approaches in reality.

Since the schemes presented in [5], [6], [12] directly relate to our work, next we briefly summarize their results. In [5], Srinivasan *et al.* focused on the energy efficiency aspect, where in their Tit for Tat (TFT)-based solution, the nodes are classified into different energy classes and the behavior of each node depends on the energy classes of the participants of each connection. They demonstrated that if two nodes belong to the same class, they should apply the same packet forwarding ratio. Similar TFT-based approaches were also considered by Felegyhazi *et al.* in [12]. In [6], Urpi *et al.* claimed that it is not possible to force a node to forward more packets than it sends on average, and then concluded that cooperation can be enforced in a mobile ad-hoc network provided that enough members of the network agree on it, and if no node has to forward more traffic that it generates.

In our previous work [16], [17], we proved that in order to maximize its own payoff and be robust to possible cheating behavior, a player should not forward more packets than its opponent does. We have also shown that this strategy can achieve Pareto optimality, cheat proofness, and absolute fairness. However, in [17], we have assumed perfect monitoring. In this paper, we focus on the scenario that the underlying monitoring is not perfect, which makes the task even more challenging. Meanwhile, instead of trying to identify the conditions under which the proposed strategy is optimal, as is done in [17] in this paper, we have also explored under what scenarios the proposed strategies cannot work well, through both analytical analysis and extensive simulations. In other words, this work can be regarded as a continuation of [17], but provides more thoughtful insights. Furthermore, in this paper, we have also studied the possible limitations of game-theoretic approaches to solve cooperation issues.

III. Design Challenges and Game Description

A. System Description and Design Challenges

In this paper, we investigate how to stimulate cooperation among selfish nodes under realistic scenarios. We consider an autonomous mobile ad-hoc network with a finite population of users, denoted by N. We do not assume the availability of any tamper-proof hardware or central banking service; therefore, the scheme should be completely reputation based. We focus on the

situation that each user will stay in the network for a relatively long time. But we do not require them to keep connected all of the time, and we allow users to leave and join the network when necessary. It is worth pointing out that our goal is not to enforce all of the users to act in a fully cooperative fashion, which has been shown in [12]and [13] to not be achievable in most situations. Instead, our goal is to stimulate cooperation among nodes as much as possible through playing conditional reciprocal altruism and, at the same time, take into consideration the possible cheating and malicious behavior as well as fairness concerns.

We assume that each user has a unique registered and verifiable identity, and may send information to the others or request information from the others. In other words, certain third-party authorities may be required to issue such identities. We focus on the information-push model, where it is the source's duty to guarantee the successful delivery of packets to their destinations. But the obtained results can be easily extended to the information-pull model. We assume that for each user $i \in N$, forwarding a packet will incur cost c_i and letting a packet be successfully delivered to its destination can bring it gain g_i. Here, the cost corresponds to the efforts spent by i, such as energy, and the gain is usually user specific and/or application specific.

In general, due to the multihop nature, when a node wants to send a packet to a certain destination, a sequence of nodes will usually be requested to help forward this packet. We refer to the sequence of the ordered nodes as a route, the set of intermediate nodes on a route as relays, and the procedure to discover a route as route discovery. In general, the route discovery can be partitioned into three stages. In the first stage, the requester notifies other nodes in the network that it wants to find a route to a certain destination. In the second stage, other nodes in the network will make their decisions on whether they will agree to be on the discovered route. In the third stage, the requester will determine which route should be used.

In general, not all packet forwarding decisions can be perfectly executed. For example, when a node has decided to help another node to forward a packet, the packet may still be dropped due to link breakage or the transmission may fail due to channel errors. In this paper, we refer to those factors that may cause decision execution error as noise, which include environmental unpredictability and system uncertainty, channel noise, mobility, etc. We use p_e to denote the average packet dropping probability due to noise. It is worth mentioning that the packet dropping probability may vary over time due to the varying channel conditions, mobility, etc. In this paper, for packet dropping due to noise, i.i.d. and non-i.i.d. cases will be studied.

We also assume that some underlying monitoring schemes have been employed (such as those proposed in [2], [18], and [19]) which can let the source know whether its packets have been successfully delivered to their destinations. Meanwhile, if a packet has been dropped by some relay, the underlying monitoring mechanism can let the source know who has dropped this packet. However, we do not assume any perfect monitoring; instead, we assume that even a node has successfully forwarded a packet, with a probability of no more than p_f, it can be observed as dropping a packet (i.e., false alarm). On the other hand, when a packet has been dropped by a certain relay, with a probability of no more than p_m, this can be observed as a forwarding event (i.e., misdetect). Here, p_f and p_m characterize the capability of the underlying monitoring mechanism. It is easy to understand that p_f and p_m may vary according to the underlying monitoring mechanism and the monitoring environment.

Before devising cooperation stimulation strategies for autonomous mobile ad-hoc networks, we first summarize some challenges that we may meet.

- Existence of noise: In many existing cooperation enforcement schemes, such as [5] and [12], each node decides its next step action based solely on the quality of service it has received in the current and/or previous stages, such as normalized throughput. However, if noise exists, some packets may be dropped unintentionally during the delivery. This can reduce the quality of service experienced by some nodes. As a consequence, these nodes will also lower the service quality provided by them. Such an avalanche effect may quickly propagate throughout the network and after some time, no nodes will forward packets for the others [12]. When designing cooperation stimulation strategies in realistic scenarios, the effect of noise has to be thoroughly considered.
- Imperfect monitoring: Since nodes usually base only on what they have observed to make their decisions, imperfect monitoring can always be taken advantage of by greedy or malicious nodes. For example, when the misdetect ratio p_m is high, a node can always drop other nodes' packets but still claim that it has forwarded. None of the existing approaches have been designed with the consideration of noise and imperfect monitoring, which greatly limits their potential applications in realistic scenarios.
- Presence of malicious users: If no malicious nodes exist and all nodes want to enjoy high-quality network service, stimulating cooperation may be less challenging according to the following logic: since misbehavior conducted by some nodes can lead to the decrease of service quality experienced by some other nodes, which may consequently reduce the service quality provided by them. After some time, such quality degradation will propagate back to those nodes that initially conducted such misbehavior [12]. Therefore, nodes have no incentive to intentionally behave maliciously. However, since an attackers' goal is usually to decrease the network service quality, they would like to see such misbehavior propagation. This makes cooperation stimulation extremely challenging. Further, it has been recognized that malicious behavior in autonomous ad-hoc networks will not be uncommon due to the loose access control [11], while security issues have been overlooked in the past when designing cooperation stimulation strategies.
- Topology dependency: It has been pointed out in [12] that network topology plays an important role when designing cooperation enforcement strategies, and usually it is impossible to find a strategy to enforce all nodes to play fully cooperatively in static ad-hoc networks. For example, if a user is in a bad location such that no users rely on him or her to forward packets, it is usually impossible for him or her to find other users to help him or her.

TABLE I
SUMMARY OF NOTATIONS

N_s	The set of selfish players.
N_m	The set of attackers
N	The set of network users, where $N = N_m + N_s$.
t_f	The lifetime of this network.
p_e	Packet dropping probability due to noise.
p_f	Probability that a successful packet forwarding is observed as dropping.
p_m	Probability that a packet dropping is observed as forwarding.
θ_i	The type of player i, which can be either malicious or selfish.
c_i	The cost incurred to player i by transmitting a packet either for itself or for the others.
g_i	The gain player i can get g_i for any successfully delivered packet originating from it.
$T_i(t)$	The number of packets that player i needs to send by time t.
$S_i(t)$	The number of packets that have successfully reached their destinations by time t with i being the source.
$F_i(j,t)$	The number of packets that i has forwarded for j by time t.
$F_i(t)$	The summation of $F_i(j,t)$ over all $j \in N$, that is $F_i(t) = \sum_{j \in N} F_i(j,t)$.
$W_i(j,t)$	The total number of useless packet transmissions that player i has caused to player j by time t due to i dropping those packets transmitted by j.
$R_i(j,t)$	The number of packets that node j has agreed to forward for node i by time t.
$H_i(j,t)$	The times that i has observed j forwarding a packet for it.
α	An acceptable false alarm ratio when doing attacker detection.
$\beta(i,j)$	Node i's confidence on whether node j is malicious.
$D_i^{max}(j,t)$	The cooperation level node i set for node j.

- Changing topology and opponents: In ad-hoc networks, at each time instance, each node may request different nodes to forward packets for it due to the topology change or other reasons, and/or be requested by different nodes. This also poses a big challenge to cooperation stimulation: since nodes are selfish, unless a relay node is sure with high confidence that those requesters will return the favor later, it has no incentive to forward packets for them.
- Variable service request rates: Similar to changing opponents, we have identified that the variable request rate also plays an important role. For example, if a node has too many packets to send, it is usually impossible to let the other nodes forward all of the packets for it, unless it can return enough favors to the others. Further, due to the topology change, a node that is requested may not need the requester' help immediately, though it may need it later.
- Nonrepeated model: Most previous work addresses cooperation enforcement under a repeated game model, such as [5], [6], [10], [12], and [16], which assume either random connection or fixed setup. However, the repeated model rarely holds in reality. This leads to a new challenge that the favor cannot be returned immediately, which is one major hurdle for effective cooperation stimulation.

In [20], Dawkins demonstrated that reciprocal altruism is beneficial for every ecological system when favors are granted simultaneously. However, when favors cannot be granted simultaneously, altruism may not guarantee satisfactory future payback, especially when the future is not predictable. The situation will be further deteriorated when the observation is imperfect with a high false alarm ratio and misdetect ratio. In this paper, one critical goal is to design attack-resistant cooperation stimulation strategies for autonomous mobile ad-hoc networks which can even work well under a noisy and hostile environment with imperfect monitoring.

B. Multistage Secure Routing and Packet Forwarding Game

Similar to [17], in this paper, we to model the dynamic interactions among nodes in autonomous mobile ad-hoc networks as a multistage secure routing and packet forwarding game, where the notations are summarized in Table I:

- Players: A finite set of network users, denoted by N.
- Types: Each player $i \in N$ has a type $\theta_i \in \Theta$ where $\Theta = \{\text{sel fish}, \text{ malicious}\}$. Meanwhile, no player knows the others' types *a priori*.
- Strategy space:
  1) Route participation stage: For each player, after receiving a request asking it to be on a certain route, it can either accept or refuse this request.
  2) Route selection stage: For each player who has a packet to send, after discovering a valid route, it can either use or not use this route to send the packet.
  3) Packet forwarding stage: For each relay, once it has received a packet requiring it to forward, its decision can be to either forward or drop this packet.
- Utility functions: Based on the notations in Table I, we model the players' utility as follows:
  1) For any selfish player i, its objective is to maximize

$$U_i^s(t_f) = \frac{S_i(t_f)g_i - F_i(t_f)c_i}{T_i(t_f)}. \quad (1)$$

2) For any attacker j, its objective is to maximize

$$U_j^m(t_f) = \frac{1}{t_f}\sum_{i\in N}(W_j(i,t_f) + F_i(j,t_f))c_i - \eta F_j(t_f)c_j. \tag{2}$$

Here, η is introduced to determine the relative importance of the attackers' cost compared to the other nodes' cost. That is, it is worth spending cost c to cause damage c' to other nodes only if $\eta < \frac{c'}{c}$. If the game will be played for an infinite duration, their utilities will become $\lim_{t_f\to\infty} U_i^s(t_f)$ and $\lim_{t_f\to\infty} U_j^m(t_f)$, respectively.

On the right-hand side of (1), the numerator denotes the net profit (i.e., total gain minus total cost) that the selfish node i obtained, and the denominator denotes the total number of packets that i needs to send. This utility represents the average net profit that i can obtain per packet. We can see that maximizing (1) is equivalent to maximizing the total number of successfully delivered packets subject to the total cost constraint. If $c_i = 0$, this is equal to maximizing the throughput.

The summation in the right-hand side of (2) represents the net damage of the other nodes by j. Since, in general, this value may increase monotonically, we normalize it by using the network lifetime t_f. Now, this utility represents the average net damage that j caused to the other nodes per time unit. From (2), we can see that in this game setting, the attackers' goal is to waste the other nodes' cost (or energy) as much as possible. Other possible alternatives, such as minimizing the others' payoff, will also be discussed later.

The aforementioned game can be divided into many subgames as will be explained. Once a player wants to send a packet to a certain destination, a subgame will be initiated which consists of, at most, three stages: in the first stage, the source will request some players to be on a certain route to the destination; in the second stage, the source will decide whether it should use this route to send the packet; in the third stage, each relay player will decide whether it should help the source to forward this packet once a packet is received. We refer to each subgame as a single routing and packet forwarding subgame.

IV. Attack-Resistant Cooperation Stimulation

A. Statistical Drop Packet Attack Detection

Before devising attack-resistant cooperation stimulation strategies, we first study how to handle possible malicious behavior. We focus on two classes of attacks: dropping packet attack and injecting traffic attack. Next, we show how to detect a dropping packet attack under noise with imperfect monitoring.

Let $R_i(j,t)$ denote the number of packets that node j has agreed to forward for node i by time t and let $H_i(j,t)$ denote the times that i has observed j forwarding a packet for it. If j has never intentionally dropped i's packets, given p_e, p_f, and p_m, in average we should have

$$p_o R_i(j,t) \le H_i(j,t) \le (1 - p_e + p_e p_m)R_i(j,t) \tag{3}$$

with $p_o = (1-p_e)(1-p_f)$. Then, a simple detection rule can be as follows: node i will mark node j as intentionally dropping packets if the following holds:

$$H_i(j,t) < R_i(j,t)p_o - \Delta(R_i(j,t), p_e, p_f, p_m) \tag{4}$$

where $\Delta(R_i(j,t), p_e, p_f, p_m)$ is a function of p_e, p_f, p_m, and $R_i(j,t)$. In general, there is a tradeoff when selecting $\Delta(R_i(j,t), p_e, p_f, p_m)$. A large $\Delta(R_i(j,t), p_e, p_f, p_m)$ may incur a high misdetect ratio, while a small $\Delta(R_i(j,t), p_e, p_f, p_m)$ may result in high false alarm ratio. One way to find a good $\Delta(R_i(j,t), p_e, p_f, p_m)$ is to apply the Neyman–Pearson hypothesis testing theory [21]. Let $P_F(\Delta)$ denote the false alarm probability resulting from using a certain Δ in (4), and let $P_M(\Delta)$ denote the miss probability resulting from using a certain Δ in (4). Given a certain acceptable false alarm probability α, we say that Δ^* is optimal if

$$\Delta^* \in \min_{\Delta} P_M(\Delta) \text{ subject to } P_F(\Delta) < \alpha. \tag{5}$$

If packet dropping due to noise can be modeled as an independent identically distributed (i.i.d.) random process with drop probability p_e, and the observation errors are also independent identically distributed random processes and independent of each other, then according to the central limit theorem [22], for any $x \in \mathcal{R}$, we have

$$\lim_{R_i(j,t)\to\infty} \text{Prob}\left(\frac{H_i(j,t) - R_i(j,t)p_o}{\sqrt{R_i(j,t)p_o(1-p_o)}} \ge x\right) \ge 1 - \Phi(x) \tag{6}$$

where $\Phi(x)$ is the cumulative distribution function of the normal distribution with mean 0 and variance 1. Then, we can let

$$\Delta(R_i(j,t), p_e, p_f, p_m) = x\sqrt{R_i(j,t)p_o(1-p_o)}. \tag{7}$$

In this case, the false alarm ratio will be no more than $1-\Phi(x)$ when $R_i(j,t)$ is large, and the obtained detector (4) with Δ being defined in (7) is an optimal Neyman–Pearson detector subject to the false alarm probability $\alpha = 1 - \Phi(x)$. Since, in general, $\Phi(x)$ can still approach 1 even for a small positive x, $\Delta(R_i(j,t), p_e, p_f, p_m)$ will be a very small value compared to np_o for a large $R_i(j,t)$. However, in general, neither packet dropping nor observation error is i.i.d. Under such circumstances, if the aforementioned detection rule is used, the false alarm ratio will usually be larger than $1 - \Phi(x)$. In order to maintain the same false alarm probability as in i.i.d. cases, in non-i.i.d. cases, the threshold $\Delta(R_i(j,t), p_e, p_f, p_m)$ should also be increased.

Let $\beta = 1-\alpha$, which can be explained as i's confidence on its detection decision. The value of β lies in the range of [0,1], with 0 indicating that i has not marked j as malicious and 1 indicating that i is sure that j is malicious. Then, we have $\beta = \Phi(x)$ for the i.i.d. scenarios and $\beta < \Phi(x)$ for the non-i.i.d. scenarios.

Once node i has marked node j as intentionally dropping packets, one possible rule is that it should not work with j again. However, such a rule has a drawback that if j has been mistakenly marked as malicious, it can never recover, since i will not

give it any chance. To overcome this drawback, we modify this decision rule such that j will be given a chance to recover, which will be described in the following subsection.

B. Cooperation Strategy With Attacker Detection

The strategies for nonmalicious players involve decision making in the following three stages: route participation stage, route selection stage, and packet forwarding stage.

1) Route Participation Stage: We first study what decision a selfish node i should make when it receives a route participation request from node j. First, if i has detected j as malicious with confidence β, with probability $1-\beta$, it should immediately refuse this request. Second, even if j has not been marked as malicious by i, i should accept this request only if it believes that it can get help from j later. However, whether i can get help from j depends on a lot of uncertain factors, such as i's and j's future requests, the changing network topology, j's strategy, and so on. Due to the unpredictability of future and favors not being granted simultaneously, stimulating i to act cooperatively is difficult.

In this paper, we focus on the scenario that nodes will stay in the network for a relatively long time. We consider the following strategy: a node may first forward some packets for other nodes without getting instantaneous payback. However, in order to be robust to possible malicious behavior (e.g., injecting traffic attack) or greedy behavior (e.g., request more but return less), a node should not be too generous. Before formalizing the aforementioned strategy, we first introduce a simple procedure: let β be i's confidence on whether j is malicious, i then randomly picks a value r between 0 and 1, and will give j another chance if $r < 1-\beta$. We refer to this procedure as the recovery check procedure. Let $\tilde{F}_j(i,t)$ be i's estimate of $F_j(i,t)$. Then, the aforementioned strategy can be translated as follows: i will accept j's route participation request only if j has passed the recovery check and the following holds:

$$F_i(j,t) - \tilde{F}_j(i,t) < D_i^{\max}(j,t). \tag{8}$$

Similar to [11], we refer to $F_i(j,t) - \tilde{F}_j(i,t)$ as i's estimated balance with j, and refer to $D_i^{\max}(j,t)$ as the cooperation level. Setting $D_i^{\max}(j,t)$ to be ∞ means that i will always help j, setting $D_i^{\max}(j,t)$ to be $-\infty$ means that i will never help j, and setting $D_i^{\max}(j,t)$ to be a finite value means that i will conditionally help j. Meanwhile, $D_i^{\max}(j,t)$ can be either constant or variable depending on i's past interactions with j. It is easy to see that a good choice of $D_i^{\max}(j,t)$ is a key to optimizing i's performance.

In order for the aforementioned strategy to work well, node i needs to have a good estimate of $F_j(i,t)$ for any other node j and needs to select a good cooperation level. We first study how to get a good estimate of $F_j(i,t)$. If i can have accurate knowledge of monitoring errors experienced by j, denoted by $\tilde{p}_f$ and $\tilde{p}_m$, then we should have

$$F_j(i,t)((1-p_e)(1-\tilde{p}_f) + p_e\tilde{p}_m) \simeq H_i(j,t). \tag{9}$$

Then, a good estimate of $F_j(i,t)$ can be

$$\tilde{F}_j(i,t) = \frac{H_i(j,t)}{(1-p_e)(1-\tilde{p}_f) + p_e\tilde{p}_m}. \tag{10}$$

However, in general, i cannot accurately estimate $\tilde{p}_f$ and $\tilde{p}_m$. In such scenarios, a more conservative estimate can be

$$\tilde{F}_j(i,t) = \frac{H_i(j,t)}{(1-p_e)(1-p_f)}. \tag{11}$$

Consequently, j can take advantage of such inaccuracy to forward less packets for i, or ask i to forward more packets for it. This will be further investigated in Section V.

Now we study how to select a good cooperation level. First, finding an optimal cooperation level is usually impossible unless nodes can accurately predict the future. In general, cooperation level $D_i^{\max}(j,t)$ is related to both i's and j's request rate. For example, if i has a relatively low request rate compared to the others, a relatively small $D_i^{\max}(j,t)$ should be able to work well. However, if i's request rate is much higher than the other nodes in the network or exhibits too high of a bursty pattern, a larger $D_i^{\max}(j,t)$ may be needed. Meanwhile, $D_i^{\max}(j,t)$ may also change according to i's interactions with j. For example, if i and j have helped each other many times, slightly increasing their cooperation levels may be a good choice from both nodes' point of view. Extensive simulations have been conducted to study the effect of the cooperation level, and the results suggest that when all nodes almost have equal request rates, a relatively small cooperation level can work well.

2) Route Selection Stage: Next, we study the strategy in the route selection stage. Once a set of routes have been discovered by node i with all relays on these routes having agreed to forward packets for it, the following strategy will be taken by i: first, i will not further consider this route if any relay cannot pass the recovery check; second, among all of those routes with all nodes having passed recovery check, i will pick the one with the minimum number of hops.

3) Packet Forwarding Stage: Now we consider the strategy in the packet forwarding stage. For any selfish node, once it has agreed to forward a packet for a certain node, it should not intentionally drop this packet unless the following can hold:

$$(1-p_e)(1-\tilde{p}_f) + p_e\tilde{p}_m \le \tilde{p}_m. \tag{12}$$

That is, $\tilde{p}_f + \tilde{p}_m \ge 1$, where $\tilde{p}_f$ and $\tilde{p}_m$ are the actual false alarm ratio and misdetect ratio experienced by the node. If (12) holds, this means that the chance that it will be marked as malicious even after dropping all of the packets will still be no more than forwarding all packets due to high monitoring inaccuracy. However, if (12) cannot hold, intentionally dropping packets will not be a good strategy if it still needs others' help, since such dropping may cause it to be detected as malicious and, consequently, cannot get help from other nodes in the future.

Let $\beta(i,j)$ denote i's confidence on whether j is malicious. By combining the attacker detection strategy and the routing and packet forwarding strategies described before, we devise the following attack-resistant cooperation stimulation strategy:

Attack-resistant cooperation stimulation strategy: For each single routing and packet forwarding subgame, assuming that

P_1 is the initiator who wants to send a packet to P_n at time t, and a route "$P_1 \rightarrow P_2 \rightarrow \ldots \rightarrow P_n$" has been discovered by P_1. After P_1 has sent requests to all of the relays on this route asking them to participate, for each nonmalicious player on this route, the following strategies should be taken.

1) In the route participation stage: For any relay P_i, it will accept this request if and only if P_1 can pass the recovery check and $F_{P_i}(P_1,t) - \tilde{F}_{P_1}(P_i,t) < D_{P_i}^{\max}(P_1,t)$; otherwise, it should refuse.
2) In the route selection stage: P_1 will use this route if and only if all relays on this route have passed the recovery check and this route has the minimum number of hops among all of those routes with all relays having passed recovery check; otherwise, P_1 should not use this route.
3) In the packet forwarding stage: For any relay P_i, it will forward this packet if and only if it has agreed to be on this route and (12) does not hold; otherwise, it should drop.
4) Attacker detection: Let α be an acceptable false alarm ratio from P_1's point of view. Then, it will mark a relay P_j as malicious if (4) holds with $i = P_1$, $j = P_i$ with Δ being calculated as in (7). Consequently, P_1 updates $\beta(P_1,P_j)$ as $1 - alpha$.

V. Game-Theoretic Analysis and Limitations

A. Strategy Analysis Under No Attacks

We first consider the decisions made by the relays in the packet forwarding stage. As long as (12) does not hold and the source i can get an accurate estimate of $F_j(i,t)$, from any selfish node's point of view, the only gain after intentionally dropping a packet is saving cost c_j, while the penalty includes the increase of the probability being marked as malicious by i and the decrease of the number of packets that i will forward for j in the future. Therefore, j has no incentive to intentionally drop packets in such scenarios.

What is the consequence of an inaccurate estimate of $F_j(i,t)$? Let us assume that $\tilde{p}_f$ and $\tilde{p}_m$ are the actual false alarm and misdetect ratios experienced by j, and i does not know it. In this case, i may use (10) to estimate $F_j(i,t)$, and we have

$$\frac{F_j(i,t)}{\tilde{F}_j(i,t)} \simeq \frac{(1-p_e)(1-p_f)}{(1-p_e)(1-\tilde{p}_f)+p_e\tilde{p}_m}. \tag{13}$$

If $\tilde{p}_f < p_f$, then we have $\tilde{F}_j(i,t) > F_j(i,t)$, and consequently

$$\lim_{F_i(j,t)\rightarrow\infty}\left(\frac{F_j(i,t)}{F_i(j,t)}\right) = \frac{(1-p_e)(1-p_f)}{(1-p_e)(1-\tilde{p}_f)+p_e\tilde{p}_m}. \tag{14}$$

In other words, node j can take advantage of imperfect monitoring to increase its performance by forwarding less packets for node i. However, if the underlying monitoring mechanism can guarantee p_f and p_m to be small enough, the damage caused to node i will be very limited. Further, if node i also experiences a lower false alarm ratio, the damage will be further reduced, since the aforementioned analysis is also applicable to i. We can also check that if the false alarm ratio and misdetect ratio experienced by node i and j are the same, then we can still have $\lim_{F_i(j,t)\rightarrow\infty}((F_j(i,t)/F_i(j,t))) = 1$.

Next, we consider the source's decision in the route selection stage. If no relays on the selected route have been marked as malicious by the source, it is easy to see that this is an optimal selection. What is the consequence if some relays have been marked as malicious? First, with very small probability, those nodes can pass the recovery check, so even if they are malicious, the long-term average damage is still negligible. Second, since these nodes may have been mistakenly marked as malicious, such chance can allow them to recover their reputation, and may consequently increase the source's future payoff, since it may have more resources to select and use.

Finally, we analyze the relay's decision in the route participation stage. The optimality of the proposed strategy in this stage depends on a lot of uncertain factors, such as the nodes' future request pattern, the changing topology, the nodes' future staying time, the selection of good cooperation level, etc. Since most of these factors cannot be known *a priori*, the optimality of the proposed strategies cannot be guaranteed. It is usually impossible to find an optimal strategy without being able to accurately predict the future. However, our simulation results show that when the nodes' request rates do not vary a lot, a relatively small cooperation level can work well.

If the future is predictable or at least partially predictable, such as the network being kept alive for a long time, all nodes staying in the network will keep generating and sending packets, and any pair of nodes will meet and request each other's help again and again, and then each node can set its cooperation level to be a very large positive constant without affecting its overall performance (any extra constant cost will not affect the overall payoff as long as $\lim_{t\rightarrow\infty} T_i(t) = \infty$). Then, the proposed strategies can form a Nash equilibrium, and are Pareto optimal, are subgame perfect, and achieve absolute fairness (in cost), provided that each node i can accurately estimate $F_j(i,t)$ for any other node j, and $D_i^{\max}(j,t)$ is large enough to accommodate possible variable and bursty requests between them. The proof is easy by following the aforementioned analysis, which is not used here due to space limitations (In [17], we have provided a detailed proof of similar statements.) Unfortunately, such ideal scenarios usually do not exist in reality. That is, a gap exists between the ideal game model and the reality. Accordingly, the devised strategy cannot maintain its optimality in reality. However, our simulation results demonstrate that the devised strategy can still work well in most scenarios, which suggests that game-theoretic approaches can still provide thoughtful insights and useful guidelines when devising cooperation strategies even when some gap exists between the ideal model and the reality.

B. Attacking Strategy and Damage Analysis

Thus far, we mainly focus on the scenarios that no nodes are malicious. Next, we analyze the possible damage that can be caused by the attackers. Specifically, we focus on the following two important attacks: dropping packets and injecting traffic. That is, to damage the network, the attackers can either drop other nodes' packet, or inject a lot of traffic to consume other nodes' resources. We first consider dropping a packet attack. According to the devised strategy, for attacker j to avoid being marked as malicious by node i, the highest packet drop ratio

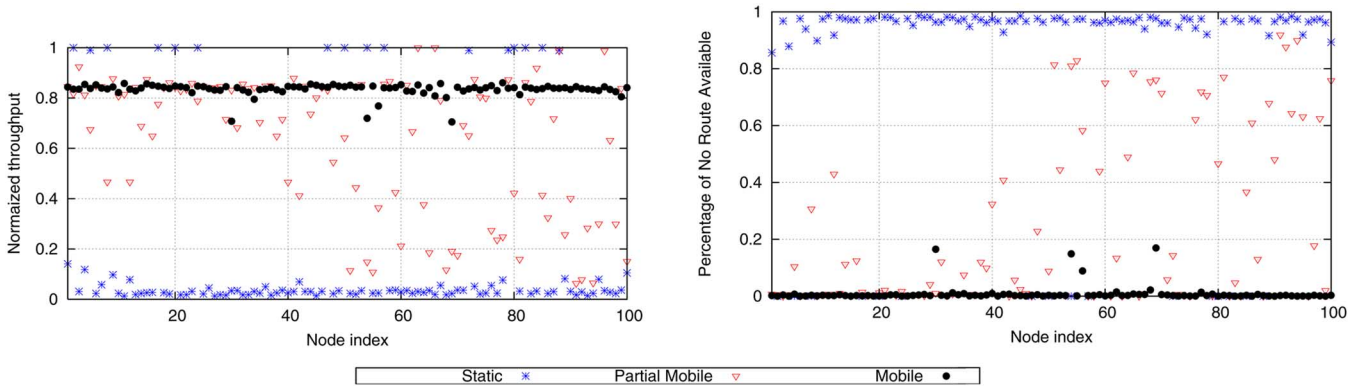


Fig. 1. Effects of mobility on cooperation stimulation.

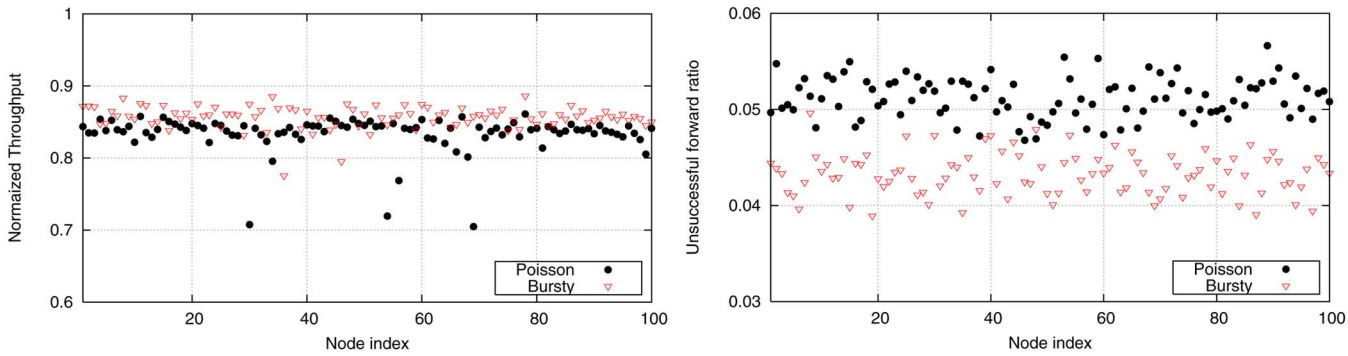


Fig. 2. Effects of traffic pattern on cooperation stimulation.

p'_e that it can employ should satisfy the following inequality to avoid being detected:

$$(1-p_e)(1-p_f) \le (1-p'_e)(1-\tilde{p}_f) + p'_e\tilde{p}_m \tag{15}$$

where $\tilde{p}_f$ and $\tilde{p}_m$ are the actual false alarm ratio and misdetect ratio experienced by j. That is, the observed times of packet forwarding are no less than the value corresponding to the normal behaviors. Since, in general, we can

$$p_e(1-p_f) + (p_f - \tilde{p}_f) \ge 0 \tag{16}$$

the maximum possible p'_e that the attacker can use without being detected is

$$p'_e = \begin{cases} \min\left\{1, \frac{p_e(1-p_f)+(p_f-\tilde{p}_f)}{1-\tilde{p}_f-\tilde{p}_m}\right\}, & \text{if } 1-\tilde{p}_f-\tilde{p}_m > 0 \\ 1, & \text{if } 1-\tilde{p}_f-\tilde{p}_m \le 0. \end{cases} \tag{17}$$

These results tell us that if the attackers can make the misdetection ratio large enough (i.e., $\tilde{p}_m \ge 1-\tilde{p}_f$), it can arbitrarily drop the packet without being detected.

Now we study the case for $1-\tilde{p}_f-\tilde{p}_m > 0$. In this case, the attacker can set the drop ratio to be

$$p'_e = \min\left\{\frac{p_e(1-p_f)+(p_f-\tilde{p}_f)}{1-\tilde{p}_f-\tilde{p}_m}, 1\right\}. \tag{18}$$

Then we have

$$\min\left\{\frac{\tilde{p}_m}{1-p_f-\tilde{p}_m}p_e, 1-p_e\right\} < p'_e - p_e < \min\left\{\frac{p_e\tilde{p}_m+(1-p_e)p_f}{1-\tilde{p}_m}, 1-p_e\right\} \tag{19}$$

where $p'_e - p_e$ can be regarded as the extra damage caused by the attackers without being detected.

If an attacker can successfully exploit the underlying monitoring to avoid being detected, such as experiencing a high $\tilde{p}_m$, then the extra number of packets it can drop without being detected can increase dramatically. According to (19), the extra damage may increase nonlinearly with the increase of $\tilde{p}_m$. This suggests that it is critical to have a robust monitoring scheme to ensure that the monitoring error will not be too large. Actually, from (19), we can also see that even for $\tilde{p}_m = 0.5$, $p'_e - p_e$ is still upperbounded by $p_e + 2p_f$, which is still small as long as p_e and p_f are small.

For an injecting traffic attack, since each selfish node i will try to maintain $\lim_{F_i(j,t)\to\infty}((F_j(i,t)/F_i(j,t))) = 1$, for any node j, the extra number of packets that node j can request node i to forward is always bounded. According to (14), the maximum possible ratio between $F_i(j,t)$ and $F_j(i,t)$ is upperbounded by $((1-p_e)(1-\tilde{p}_f)+p_e\tilde{p}_m/(1-p_e)(1-p_f))$ provided $\tilde{p}_f + \tilde{p}_m < 1$. Meanwhile, if the underlying monitoring mechanism can ensure that p_m and p_f are small, the ratio will be small. However, if j can successfully manage to let $\tilde{p}_f + \tilde{p}_m \ge 1$, such as making the misdetect ratio approach 1,

it can always request i to forward the packet without returning any favors.

It is worth noting that under the proposed strategies, regardless of what goal the attackers may have, the selfish nodes' payoff can always be guaranteed as long as p_e, p_m, and p_f are small. Meanwhile, if η [defined in (2)] is small enough, from an attacker's point of view, maximizing (2) is almost equivalent to minimizing the selfish nodes' payoff. Otherwise, maximizing (2) may not cause as much damage as minimizing the selfish nodes' payoff since, in this case, the attackers may not be willing to continuously drop packets without being detected due to the reason that this also requires the attackers to forward a lot of packets for other nodes and may not be in their best interest.

VI. Simulation Studies

In this section, we conduct extensive simulations to evaluate the effectiveness of the devised strategy and identify when and why in some situations these strategies cannot work well.

In our simulations, both static and mobile ad-hoc networks have been studied, with mobile ad-hoc networks being our focus. In these simulations, nodes are randomly deployed inside a rectangular area of 1000 $\times$ 1000 m, and each mobile node moves according to the random waypoint model [23], which can be characterized by the following three parameters: the pause time, the minimum velocity $v_{\min}$ and the maximum velocity $v_{\max}$. We set $v_{\min} = 10$ m/s, $v_{\max} = 30$ m/s, and the average pause time as 100 s. The medium-access control (MAC) layer protocol implements the IEEE 802.11 DCF with a four-way handshaking mechanism [24]. The link bandwidth is 2 Mb/s, and the data packet size is 512 B. DSR [25] is used as the underlying route discovery protocol. The maximum transmission range is 250 m. Inside the transmission range, the channel errors are characterized in terms of outage probability. Outage is defined as the event that the received signal-to-noise ratio (SNR) falls below a certain threshold δ. Here, for the transmission distance d, the probability of outage P_O is defined as

$$\mathcal{P}_O(d) = \mathcal{P}\left(\mathrm{SNR(d)} \leq \delta\right) = 1 - \exp\left(-\frac{\delta}{\mathrm{SNR}}\right). \quad (20)$$

The transmission power has been adjusted in such a way that $(1 - \mathcal{P}_O(d = 250))^{512} = 3\%$.

In these simulations, each node randomly picks another node as the destination to send packets. The total number of selfish nodes is 100. Both p_m and p_f are set to be 5%, and α is set to be 0.1%. Each packet has a delay constraint, which is set to be 10 s. If a packet is dropped by some relay, no retransmission will be applied. For each node i, we set $g_i = 1$ and $c_i = 0.1$. The nodes are indexed from 1 to N, where N is the total number of nodes.

To conduct performance evaluation and comparison, the following are measured for each selfish node in the simulations:

- Normalized throughput: this is defined as the ratio between the total number of successfully delivered packets and the total number of packets scheduled to be sent;
- probability of no route available: this is defined as the percentage of packets dropped due to no available valid route;
- cost per successful packet delivery: this is the ratio between the total number of forwarded packets and the total number of successfully delivered packets originating from it;
- balance: this is the difference between the total number of packets that this node forwarded for the others and the total number of packets that the others forwarded for it.

According to (1), it is easy to see that a selfish node's payoff can be calculated based on its normalized throughput and the cost per successful packet delivery.

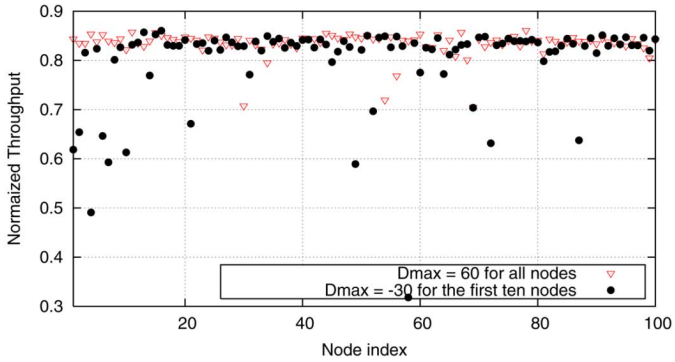


Fig. 3. Effect of negative cooperation level on cooperation stimulation.

A. Mobile Ad-Hoc Networks versus Static Ad-Hoc Networks

We first study the effect of mobility on cooperation stimulation. In this set of simulations, three types of networks are generated: mobile, partial mobile, and static. In the partial mobile ad-hoc network, the nodes with indices ranging from 1 to 50 are mobile, and the other half are static. All nodes employ the same traffic pattern: the packet interarrival time follows exponential distribution with the mean being 2 s. All nodes set their cooperation level to be 60. The simulation results are illustrated in Fig. 1.

First, from the throughput comparison, we can see that for the static case except for several nodes, the majority of nodes (85%) experience extremely bad throughput. This is due to the reason that, at most times, they cannot find a route with all relays willing to help it (shown in the second figure). For those several nodes with high normalized throughput, the reason is that the destinations are in the transmission range of the sources. These results suggest that the devised strategies cannot be used in static ad-hoc networks. Actually, in [12] and [13], the authors have demonstrated that in networks with fixed topology, cooperation enforcement is impossible to achieve by relying solely on reputation. The most basic reason is that the service that a node can provide is usually not needed by its neighbors; therefore, its neighbors have no incentive to help it.

From these results, we can also see that when all nodes are mobile, the normalized throughput can be fairly high. For example, except for four nodes, all of the other nodes have normalized throughput that is more than 80%. Even for those four nodes, their normalized throughput is still more than 70%. We can also see that for the majority of the nodes (96%), almost none of their packets are dropped due to no available routes, that is, cooperation among nodes has been effectively stimulated.

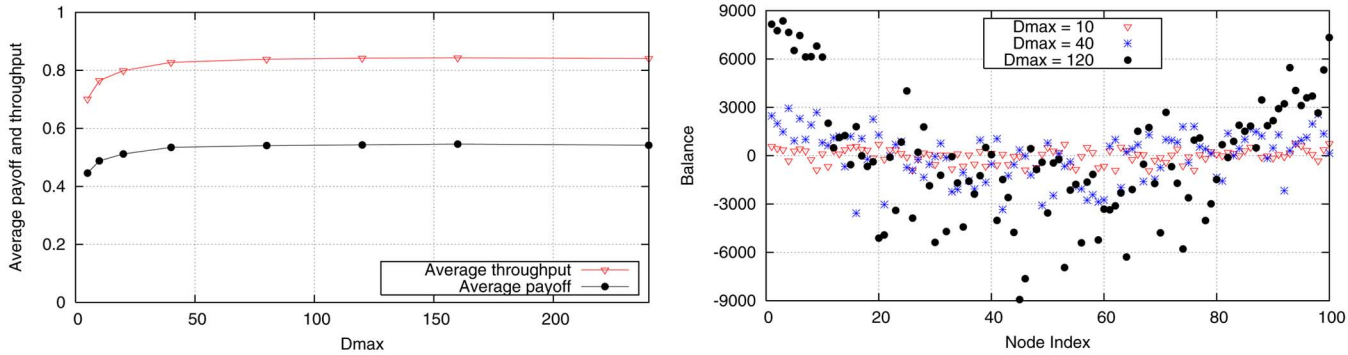


Fig. 4. Effect of cooperation level on cooperation stimulation.

Now we study the partial mobile case. From the throughput comparison, we can see that for those mobile nodes, no one has normalized a throughput of less than 40%, and the majority (33 out of 50) have a normalized throughput of higher than 80%. However, for those static nodes, the situation is totally reversed: half of them have a normalized throughput of less than 40%. This suggests that mobility can help stimulating cooperation. The underlying reason is that mobility can make the service exchange more effectively. An analogy to this is the effect of businessperson: without them, we can only exchange service locally, the service we can get will be very limited; while with the help of businesspeople, service can be exchanged globally. From now on, we will mainly focus on mobile ad-hoc networks with all nodes being mobile.

B. Bursty Traffic Pattern versus a Nonbursty Traffic Pattern

Next, we investigate the effect of traffic pattern on cooperation stimulation. In these simulations, two traffic patterns are considered: bursty and nonbursty. In the bursty case, packets are generated in a bursty pattern with an average bursty length 10, while in a nonbursty pattern, the packet arrival follows a Poisson process. In both cases, the average packet arrival rate is 0.5 packet/s. The simulation results are illustrated in Fig. 2.

It is surprising to see that the bursty case has slightly better normalized throughput than the nonbursty case. This can be explained using the unsuccessful forward ratio experienced by each node (shown in the second figure): in the bursty case, the unsuccessful forward ratio experienced by each other is 1% lower than the nonbursty case. This is because in the nonbursty case, when a packet needs to be sent, with a high probability, the existing route may have broken since this route may have been discovered a long time ago, while in the bursty case, though link breakages also occur frequently, as long as the current route is good, almost all of the packets can be delivered successfully. However, if nodes with the bursty pattern have much higher rates or if the burst length is much longer, the performance of the bursty case may be decreased, as will be shown later.

C. Effect of Negative Cooperation Level

In this set of simulations, some nodes set their cooperation level to be negative. Specifically, the first ten nodes set $D^{\max}$ to be -30, and all of the others set $D^{\max}$ to be 60. The results are illustrated in Fig. 3. From these results, we can see that the majority of nodes (six out of ten) who set $D^{\max}$ to be negative have a normalized throughput of less than 65%. Meanwhile, they also cause some other nodes to experience lower normalized throughput (six out of 90 have a normalized throughput of no more than 70%). These results suggest that as long as a node wants to stay in the network for a long time and needs to send packets continuously, they should not set their cooperation level to be negative.

D. Effect of Cooperation Level on Cooperation Stimulation

In this set of simulations, each node sets its traffic rate to be 0.5 packet/s following the Poisson arrival. In each simulation, a different $D^{\max}$ value is used, ranging from 10 to 240. The results are illustrated in Fig. 4. From the first figure, we can see that once $D^{\max} \geq 80$, both the average normalized throughput and the average payoff experienced by selfish nodes do not increase further, which suggests that in this case, setting $D^{\max} = 80$ can almost approach the optimal solution in terms of normalized throughput. However, from the second figure, we can see that with the increase of $D^{\max} \geq 80$, the balance variation experienced by nodes also increases, which leads to high unfairness. That explains why we have set $D^{\max} = 60$ in our simulations: a good tradeoff between payoff and fairness.

E. Effect of Inhomogeneous Request Rates

In this set of simulations, each node's traffic rate is determined as follows: let i be a node's index ranging from 1 to 100, then its traffic rate will be set as $(i/20) + 1)/2$ packet/s. Based on the configuration of $D^{\max}$ and traffic pattern, three cases are studied: in case 1 and 3, for each node, its traffic follows the Poisson arrival, while in case 2, each node's traffic follows a bursty arrival. Meanwhile, in case 1 and 2, all nodes set $D^{\max}$ to be 60, while in case 3, each node with index i set $D^{\max}$ to be $60 + (i/2)$. The results are shown in Fig. 5.

We first study the throughput comparison. From these results, we can see that case 3 has the highest normalized throughput while case 2 has the lowest normalized throughput. This suggests that bursty traffic may decrease the performance, while if a node has too much traffic to send, increasing their cooperation level can increase their performance. From these results, we can also see that with an increase of the traffic rate, the throughput decreases too. Although increasing $D^{\max}$ can slightly increase the performance, it cannot completely solve the problem. The reason is that the service provided by those nodes with a high

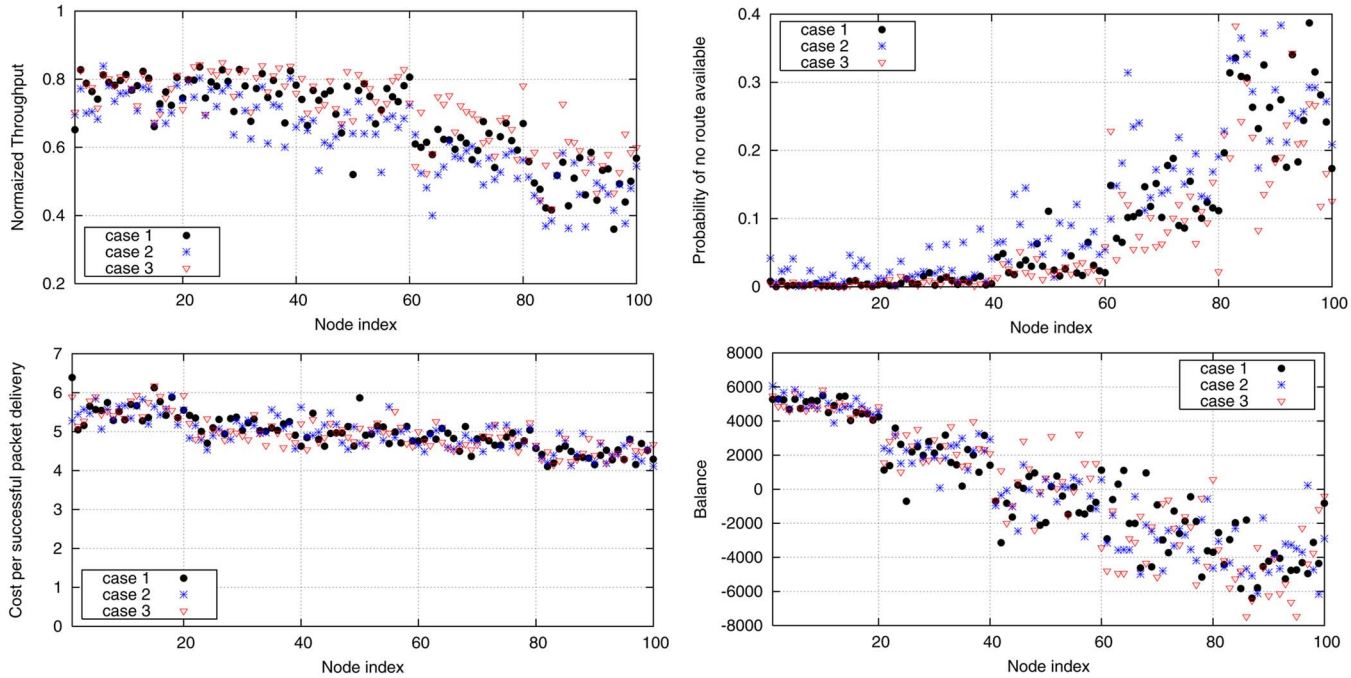


Fig. 5. Effect of inhomogeneous request rates on cooperation stimulation.

traffic rate is not needed by those nodes with lower rates. This can be shown more clearly in the following simulations.

By checking the second figure (probability of no route available) in Fig. 5, we can see that in case 2 (bursty case), a lot of packets will be dropped due to no available routes, especially when the node's traffic rate is high, which explains why they have the lowest throughput. From the third figure (cost per successful delivery) in Fig. 5, we can see that with an increase of the traffic rate, the hop number per route may decrease slightly, which is a little bit surprising, but makes sense: when a node with a high traffic rate has used up the quota assigned by those nodes with a lower rate, they are forced to use short routes, such as a one-hop route. This is also confirmed by the results in the fourth figure, which indicates that for the first 20 nodes, their overall balance almost reaches the maximum.

Next, we study an extremely asymmetric case, where in this set of simulations, except the first ten nodes which have a packet arrival rate of 5 packet/s, all of the other nodes have a packet arrival rate 0.5 packet/s. According to the first ten nodes' $D^{\max}$ values, three cases are studied: in case 1, they let $D^{\max} = 60$, in case 2 they set $D^{\max} = 120$, and in case 3, they set $D^{\max} = 180$. For the other nodes in all of the three cases, $D^{\max} = 60$. The results are illustrated in Fig. 6. From these results, we can see that by increasing $D^{\max}$ from 60 to 120, a lot of gain can be obtained (normalized throughput increases from 8% to 22%), while increasing $D^{\max}$ from 120 to 180 introduces almost no gain, and the normalized throughput is still only about 22%. This suggests that although increasing $D^{\max}$ can provide some gain, they cannot change the inherent problem.

F. Effects of Different Dropping Packet Attacks

In this set of simulations, we study the effect of different dropping packet attacks. Four dropping packet attack strategies are studied: not participating in any route discovery, dropping all packets passing through it, dropping half of the packets passing through it, and selectively dropping packets passing through it, and, at the same time, keep from being detected. Fig. 7 illustrates the evolution of the normalized throughput and payoff averaged among all selfish nodes over time. From these results, first we can see that dropping all packets can cause the maximum damage. The reason is that we have set $D^{\max}$ to be a large value (200), so each attacker can drop up to 199 of any other node's packets without being marked as malicious. However, we can also see that with time increasing, the selfish nodes' performance will also increase. From these results, we can also see that adaptive dropping can even increase the selfish nodes' performance. This is because the damage it can cause is very limited in order to avoid being detected, while keeping forwarding packets for selfish nodes can reduce the selfish nodes' average hop number per selected route. Although intuitively adaptive dropping may cause a lot of damage, in reality, this may not be the case.

G. Effect of Attacker Number

In this set of simulations, we study the selfish nodes' average performance in the presence of a different number of attackers, with the number of attackers ranging from 5 to 30. All attackers launch an injecting traffic attack, and will not forward any packets for selfish nodes. The results are illustrated in Fig. 8. From these results, we can see that with the increase of attacker number, the average normalized throughput among all selfish nodes is kept almost unchanged, and the average payoff only decreases very slightly. This can be explained using the second figure, where the total damage is defined as the total number of packets that selfish nodes have forwarded for each attacker. From this figure, we can see that after some time, no more

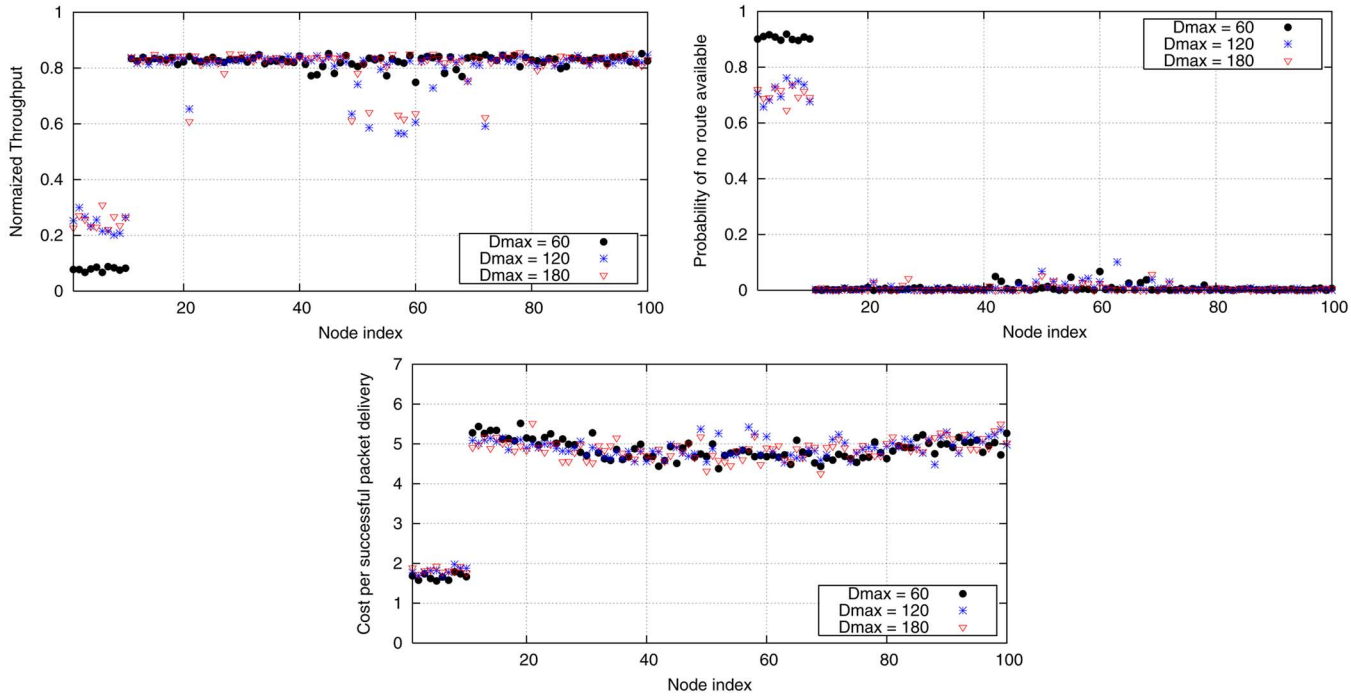


Fig. 6. Effect of inhomogeneous request rates, an extreme case.

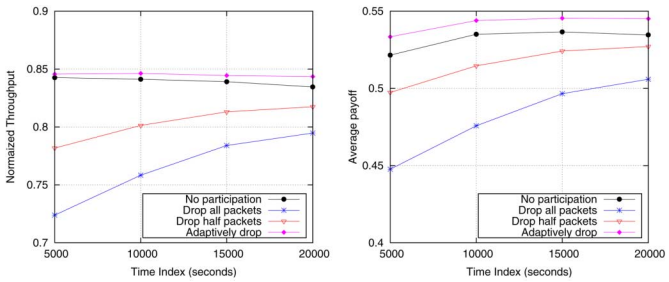


Fig. 7. Comparison of different dropping packet attacks.

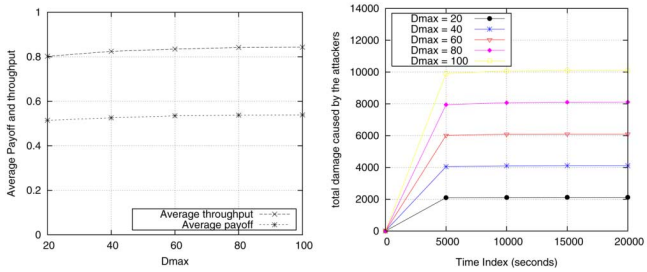


Fig. 9. Effect of cooperation level on damage.

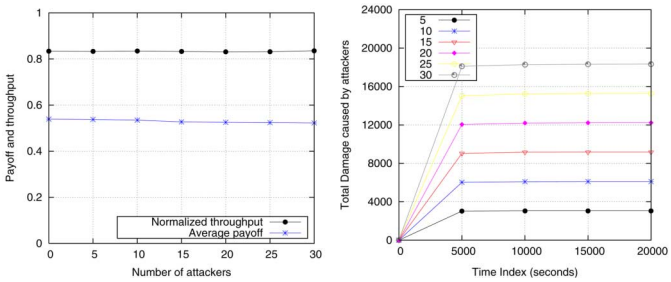


Fig. 8. Performance comparison under a different number of attackers.

damage can be caused to selfish nodes due to the reason that they have used up all of the quota assigned to them. This suggests that the proposed strategy is robust to injecting a traffic attack.

H. Cooperation Level Versus Damage

In this final set of simulations, the effect of $D^{\max}$ on selfish nodes' performance under the injecting traffic attack is studied, with the selfish nodes' $D^{\max}$ varying from 20 to 100. The results are illustrated in Fig. 9. From these results, we can see that after $D^{\max}$ passes 60, the selfish nodes' average performance (normalized throughput and payoff) were kept almost unchanged. Similar to the results illustrated in Fig. 8, for each given $D^{\max}$, the damage caused by the attackers will not change after some time due to using up all of the assigned quota. Meanwhile, the damage will increase linearly with the increase of $D^{\max}$. By also taking the fairness issue into consideration, these results also suggest $D^{\max} = 60$ can be a good choice. However, we need to keep in mind that the selection of $D^{\max}$ also depends on the underlying traffic rate. It is easy to understand that with the increase of the traffic rate, we should also increase $D^{\max}$, especially when mobility is low and traffic may exhibit strong bursty pattern and/or variable rates.

VII. Discussion and Future Work

Comparing to the pricing-based schemes, such as those in [1], [9], [13]–[15], the major drawback of reputation-based schemes is that some nodes may not get enough help to send out all of their packets. As we have demonstrated in Section III, the reason lies in the combining effect of 1) favors cannot be granted simultaneously and 2) the future is unpredictable. The pricing-

based schemes do not suffer such problems in that a node can get immediate monetary payback after providing services. The drawback of pricing-based schemes lies in the requirement of tamper-proof hardware or a central banking service. If such a requirement can be effectively satisfied with low overhead, pricing-based schemes can be a better choice than reputation-based schemes. However, it is worth pointing out that pricing-based schemes also suffer from noise and imperfect monitoring and possible malicious behavior. The study of robust pricing-based schemes has been put in our future calender.

The differences between our work and the existing reputation-based work (e.g., [5]–[8], [10], [12]) are as follows. First, we address this issue under a very realistic scenario: noisy environment, imperfect monitoring, existence of attackers, mobile nodes, an inhomogeneous traffic rate, future unpredictability, and so on. This makes our task extremely challenging, and optimal solutions may not be always available. Second, our goal is not to enforce all nodes to act fully cooperatively, but to stimulate cooperation among nodes as much as possible. The simulation results have demonstrated that our solution can work well under various scenarios and the damage caused by the attackers is limited as long as the underlying monitoring mechanism will not introduce too much uncertainty.

In most existing works, such as in [5], [6], [10], [12], each node makes its decision based solely on its own experienced quality of service, such as throughput. One advantage of such a scheme is that only end-to-end acknowledge is required, which introduces very little monitoring overhead. Another advantage is that each node only needs to keep its own past state, which introduces very little storage overhead. In our solution, we require the underlying monitoring mechanism to provide per-node monitoring, and each node needs to keep track of its balance with other nodes. Although this can introduce higher overhead, such extra overhead is necessary to stimulate cooperation under noise and imperfect monitoring and in the presence of malicious behavior, as we have demonstrated through Sections III and V. Otherwise, attackers can easily break down the network and greedy users can easily increase their payoff by taking advantage of noise and monitoring inaccuracy.

From the analysis in Section V, we can see that the underlying monitoring plays an extremely critical role in successfully stimulating cooperation among nodes. If the monitoring error is too high (i.e., high p_e and p_f), then this can be easily taken advantage of by malicious and selfish nodes. A robust and effective monitoring system will be key to the successful deployment of autonomous mobile ad-hoc networks in hostile environments, which also poses new research challenges. Further, the overhead associated with the underlying monitoring has not been included in our analysis, which may be crucial in practical implementation. In general, the higher accuracy of the monitoring scheme, the larger overhead it may incur. Due to space limitations, these issues will be addressed in a future work.

It is also worth mentioning that the security of the proposed strategy also relies on the existing secure protocols to achieve secure access control and secure authentication, and to defend those attacks launched during the route discovery procedure, such as those in [11], [18], [26]–[34]. In general, besides drop packet and inject traffic, a variety of other types of attacks exist, such as jamming, slander, etc. In this paper, our focus is not to address all of these attacks, but to provide insight on stimulating cooperation in a hostile environment under noise and imperfect monitoring. To the best of our knowledge, we are the first one to formally address this issue under such realistic scenarios. However, since the security of a system is determined by its weakest link, exploiting the possible system vulnerability has also been put in our future calender.

VIII. Conclusion

In this paper, we have investigated the issues of cooperation stimulation for autonomous mobile ad-hoc networks in a realistic context, where the communication channels are error prone, the underlying monitoring is imperfect, and the environment is hostile with possible malicious behavior. We have identified the underlying reasons why stimulating cooperation among nodes under scenarios is extremely challenging. Unlike most existing work whose goal is to enforce all nodes to act fully cooperatively, our goal is to stimulate cooperation among selfish nodes as much as possible through reciprocal altruism. We have devised a set of reputation-based attack-resistant cooperation stimulation strategies, which are completely self-organizing and fully distributed, and do not require any tamper-proof hardware or central banking or billing services. Both theoretical analysis and extensive simulation studies have demonstrated that although a gap may exist between the game model and reality, the game-theoretic approach can still provide thoughtful insights and useful guidelines when devising cooperation strategies, and the devised strategies can effectively stimulate cooperation among selfish nodes under various scenarios and meanwhile be robust to attacks.

References

[1] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. INFOCOM: 22nd Annu. Joint Conf. IEEE Computer Communications Societies*, San Francisco, CA, 2003, vol. 3, pp. 1987–1997.

[2] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annual Int. Conf. Mobile Computing and Networking*, New York, 2000, pp. 255–265.

[3] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. 6th IFIP TC6/TC11 Joint Working Conf. Communications Multimedia Security*, Deventer, The Netherlands, 2002, pp. 107–121.

[4] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Networking Computing*, New York, 2002, pp. 226–236.

[5] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad hoc networks," in *Proc. 22nd Annu. Joint Conf. IEEE Computer Communications Societies*, San Francisco, CA, 2003, pp. 808–817.

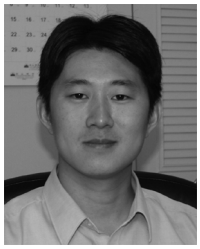
[6] A. Urpi, M. Bonuccelli, and S. Giordano, "Modeling cooperation in mobile ad hoc networks: A formal description of selfishness," presented at the Modeling Optimization Mobile, Ad Hoc Wireless Networks, Sophia-Antipolis, France, 2003.

[7] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring, "Modelling incentives for collaboration in mobile ad hoc networks," *Performance Eval.*, vol. 57, no. 4, pp. 427–439, Aug. 2004.

[8] P. Michiardi and R. Molva, "A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad hoc networks," presented at the Modeling Optimization Mobile, Ad Hoc Wireless Network, Sophia-Antipolis, France, 2003.

[9] 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. 9th Annu. Int. Conf. Mobile Computing Networking*, New York, 2003, pp. 245–259.

[10] E. Altman, A. A. Kherani, P. Michiardi, and R. Molva, "Non-cooperative forwarding in ad-hoc networks," in *Proc. 4th Int. Networking Conf.*, Waterloo, ON, Canada, May 2005, pp. 486–498.
[11] W. Yu and K. J. R. Liu, "Attack-resistant cooperation stimulation in autonomous ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 12, pp. 2260–2271, Dec. 2005, special issue.
[12] M. Felegyhazi, J.-P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 463–476, May 2006.
[13] S. Zhong, L. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks," *Wireless Netw.*, vol. 13, no. 6, pp. 799–816, 2007.
[14] L. Buttyán and J.-P. Hubaux, "Enforcing service availability in mobile Ad-Hoc wans," in *Proc. 1st ACM Int. Symp. Mobile Ad Hoc Networking Computing*, 2000, pp. 87–96.
[15] L. Buttyán and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mobile Netw. Appl.*, vol. 8, no. 5, pp. 579–592, Oct. 2003.
[16] W. Yu and K. J. R. Liu, "On optimal and cheat-proof packets forwarding strategies in autonomous ad hoc networks," in *Proc. 40th Annu. Conf. Information Sciences Systems*, 2006, pp. 1455–1460.
[17] W. Yu and K. J. R. Liu, "Game theoretic analysis of cooperation and security in autonomous mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 5, pp. 459–473, May 2007.
[18] W. Yu, Y. Sun, and K. J. R. Liu, "HADOF: Defense against routing disruptions in mobile ad hoc networks," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 1252–1261.
[19] W. Yu and K. J. R. Liu, "Secure cooperative mobile ad hoc networks against injecting traffic attacks," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 2, pp. 227–239, Jun. 2007.
[20] R. Dawkins, *The Selfish Gene*, 2nd ed. Oxford, U.K.: Oxford Univ. Press, 1990.
[21] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer, 1994.
[22] O. Kallenberg, *Foundations of Modern Probability*. New York: Springer-Verlag, 1977.
[23] J. Yoon, M. Liu, and B. Noble, "Sound Mobility Models," in *Proc. 9th Annu. Int. Conf. Mobile Computing Networking*, New York, 2003, pp. 205–216.
[24] "IEEE computer society lan man standards committee," Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11–1007, Inst. Elect. Elect. Eng.
[25] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," in *Mobile Computing, Edited by Tomasz Imielinski and Hank Korth*. Norwell, MA: Kluwer, 1996, ch. 5, pp. 153–181.
[26] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Netw. Mag.*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.
[27] J. P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Networking Computing*, New York, 2001, pp. 146–155.
[28] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," *Wireless Netw.*, vol. 11, no. 1–2, pp. 21–38, 2005.
[29] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," presented at the SCS Communication Networks and Distributed Systems Modeling Simulation Conf., San Antonio, TX, Jan. 2002.
[30] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Proc. 10th IEEE Int. Conf. Network Protocols*, Washington, DC, 2002, pp. 78–89.
[31] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. 1st ACM Workshop Wireless Security*, New York, 2002, pp. 1–10.
[32] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proc. 2nd ACM Workshop Wireless Security*, New York, 2003, pp. 30–40.
[33] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," in *Proc. 22nd Annu. Joint Conf. IEEE Computer Communications Societies*, 2003, vol. 3, pp. 1976–1986.
[34] Y.-C. Hu, A. Perrig, and D. B. Johnson, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Netw. J.*, vol. 1, pp. 175–192, 2003.



Wei Yu received the B.S. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2000, the M.S. degree in computer science from Washington University, St. Louis, MO, in 2002, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2006.

From 2000 to 2002, he was a Graduate Research Assistant at Washington University. From 2002 to 2006, he was a Graduate Research Assistant with the Communications and Signal Processing Laboratory and the Institute for Systems Research, University of Maryland. He joined Microsoft Corporation, Redmond, WA, in 2006. His research interests include network security, wireless communications and networking, game theory, wireless multimedia, handwriting recognition, and pattern recognition.

K. J. Ray Liu (F'03) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., and the Ph.D. degree from the University of California, Los Angeles, both in electrical engineering.

He is Professor and Associate Chair, Graduate Studies and Research, of the Electrical and Computer Engineering Department, University of Maryland, College Park, where he is Director of Communications and the Signal Processing Laboratory. He leads the Maryland Signals and Information Group, conducting research that encompasses broad aspects of information technology, including signal processing, communications, networking, information forensics and security, and biomedical and bioinformatics imaging.

Dr. Liu is the recipient of best paper awards from the IEEE Signal Processing Society (twice), IEEE Vehicular Technology Society, and EURASIP, IEEE Signal Processing Society Distinguished Lecturer, EURASIP Meritorious Service Award, and the National Science Foundation Young Investigator Award. He also received various teaching and research recognitions from the University of Maryland, including university-level Distinguished Scholar–Teacher Award, Invention of the Year Award, and college-level Poole and Kent Company Senior Faculty Teaching Award. He is Vice President—Publications and on the Board of Governors of the IEEE Signal Processing Society. He was the Editor-in-Chief of *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of the *EURASIP Journal on Applied Signal Processing*.