Why does it pay to be selfish in a MANET?

Article *in* IEEE Wireless Communications · January 2007

DOI: 10.1109/MWC.2006.275203 · Source: IEEE Xplore

2 authors:

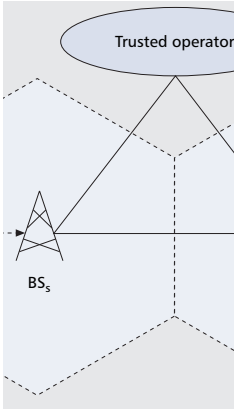
Younghwan Yoo
Pusan National University
61 PUBLICATIONS **482** CITATIONS
SEE PROFILE

Dharma Agrawal
University of Cincinnati
668 PUBLICATIONS **15,856** CITATIONS
SEE PROFILE

ACCEPTED FROM OPEN CALL

WHY DOES IT PAY TO BE SELFISH IN A MANET?

YOUNGHWAN YOO AND DHARMA P. AGRAWAL, UNIVERSITY OF CINCINNATI



Many studies have explored the use of both the carrot and the stick approaches by having reputation-based, credit-payment, and game theory schemes. The authors summarize existing schemes, identify their relative advantages, and project future directions.

ABSTRACT

Routing protocols for a mobile ad hoc network have assumed that all mobile nodes voluntarily participate in forwarding others' packets. This was a reasonable assumption because all MNs in a MANET belonged to a single authority. In the near future, however, a MANET may consist of MNs that belong to many different organizations since numerous civilian applications are expected to crop up. In this situation, some MNs may run independently and purposely decide not to forward packets so as to save their own energy. This could potentially lead to network partitioning and corresponding performance degradation. To minimize such situations in MANETs, many studies have explored the use of both the carrot and the stick approaches by having reputation-based, credit-payment, and game theory schemes. This article summarizes existing schemes, identifies their relative advantages, and projects future directions.

INTRODUCTION

Mobile ad hoc networks (MANETs) are basically peer-to-peer multihop mobile wireless networks that have neither fixed communication infrastructure nor any base stations (BSs) [1]. MANETs were originally introduced for use in dangerous situations such as rescue and battle field operations so that emergency personnel or soldiers may be aware of the location of chemical, biological, hazardous material, or tactical situations. In these networks, all mobile nodes (MNs) belong to a common authority (e.g., military or government agency) and are deployed to collaborate with each other for a common objective. This kind of MANET is termed a *closed* or *managed* ad hoc network. On the other hand, interest in commercialization of MANETs is recently growing at a much faster rate due to their portability and proliferation of mobile communication devices like laptops, PDAs, cell phones, and other intelligent radio devices [2]. A variety of MNs supplied by different manufacturers compose a MANET in a self-organizing manner and share their resources for global connectivity with their own goals. This type of MANET is called an *open* or a *pure* ad hoc network.

Unlike the typical Internet, which has dedicated nodes for basic network operations such as authorization, routing, packet forwarding, and network management, all these functions should be performed by all MNs themselves in MANETs. However, typical MNs cannot be trusted with these important network functions. Thus, security has become an essential consideration in MANETs, especially in open MANETs. The conventional Internet usually has a centralized authentication server and uses a symmetric or asymmetric cryptographic mechanism. However, MNs still have small storage, low bandwidth, high error rates, and limited battery power, in spite of recent appreciable advances in terms of power efficiency, flexibility, and robustness. For these reasons, common security algorithms designed for traditional networks are difficult to use in MANETs. Hence, new security approaches need to be developed for MANETs.

Security attacks in MANETs are divided into two categories: *active* and *passive* attacks [3]. Active attacks are performed by malicious nodes to harm the entire network operation intentionally, and include denial of service (DoS), tunneling (wormhole attack), black hole, and impersonation [3, 4]. On the other hand, passive attacks are done by selfish MNs whose goal is just to use their limited resources only for their own benefit.[1] That some MNs could be selfish is a reasonable assumption, especially in the open MANET environment, since MNs owned by different commercial entities always attempt to maximize their own interests. They do not want to use their resources to support global connectivity, even though all nodes benefit from such a commitment in the long run. Among various resources associated with MNs, energy is one of the most important, so it needs to be conserved as much as possible. In terms of energy consumption, data transmission is the most expensive function in the MANET environment. To send a bit over 10 or 100 m distance, MNs consume energy that can perform thousands to millions of arithmetic operations [5]. Thus, MNs may not forward others' packets and simply discard them on purpose. Or they may excessively reduce transmission power to save energy, resulting in network partitioning. Any such feature of MNs is called selfishness.

[1] *Some articles in the literature define passive attacks as the case where an attacker just eavesdrops communication to get important information unfairly, in contrast with active attacks to harm the entire network.*

1536-1284/06/$20.00 © 2006 IEEE

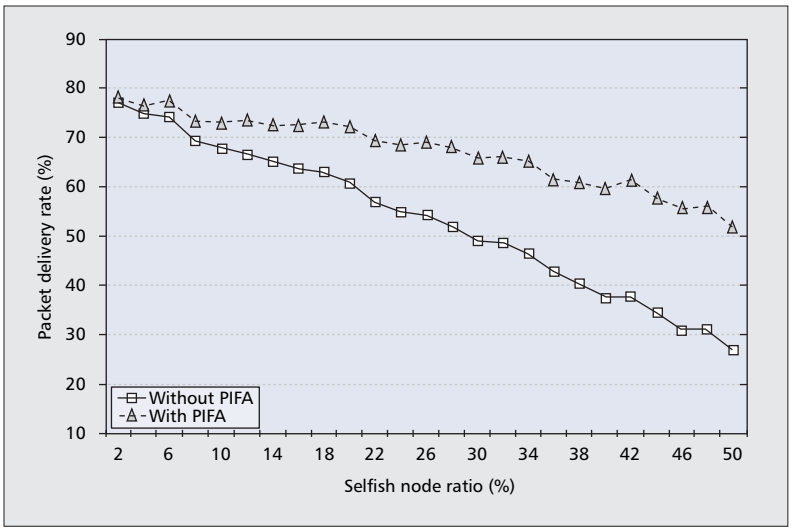


Figure 1. *Packet delivery rate against selfish MN ratio [7].*

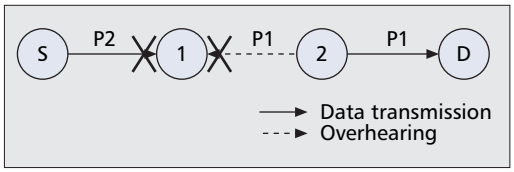


Figure 2. *While MN 1 overhears MN 2 transmitting P1, P2 arrives at MN1.*

MANETs are easily exposed to these attacks because most routing protocols like DSDV, TBRPF, AODV, and DSR [1] operate on the assumption that all MNs follows the protocols completely. To overcome this problem, several secure routing protocols have been proposed, such as SRP, ARIADNE, SEAD, ARAN, and SPINS [3]. However, they take care of only active attacks; thus, new schemes are needed for the selfishness problem. Actually, several selfishness prevention schemes have been reported since Marti et al. [6] proposed a method to detect misbehaving nodes in August 2000. The proposed strategies can be divided into the following three groups: *reputation-based schemes*, *credit-payment schemes*, and *game theory schemes*. In reputation schemes, each MN observes others' behavior and uses the information in the routing process. On the other hand, credit-payment schemes give credits (which is real or virtual money) to MNs as a reward for packet forwarding. All MNs need the credit in order to send their own packets. Finally, game theory based schemes model the forwarding process as a game whereby all rational MNs gradually determine their own optimal strategies.

Evaluation of existing methods shows that even a small percentage of selfish MNs can disrupt all communication and severely degrade network performance. Figure 1 compares the two cases with and without the selfishness prevention scheme PIFA [7], which will be explained later in detail. The figure illustrates the packet delivery rate against the ratio of selfish MNs out of all MNs. As the ratio of selfish MN increases, the packet delivery rates for the case without any selfishness prevention scheme becomes seriously degraded. On the other hand, when PIFA is adopted, the rate is maintained at an acceptable level, even with 50 percent selfish MNs. This result substantiates the necessity for a selfishness prevention scheme. The following three sections introduce existing reputation-based methods, credit-based methods, and game theory methods in order. Then, after describing other work and a summary, we propose possible future directions.

Reputation-based Method

Most secure routing algorithms provide prevention mechanisms for a variety of security attacks. However, they cannot work correctly unless it is 100 percent perfect as someone can always find a way out to bypass/fool the prevention mechanism [8]. Hence, a reactive detection scheme for misbehaving MNs is absolutely essential.

Marti *et al.* [6] propose a reputation-based scheme to mitigate bad effects of misbehaving MNs that are selfish, malicious, broken, or overloaded. Each MN runs two extensions on top of DSR: *watchdog* and *pathrater*. The watchdog overhears neighbor MNs' transmission *promiscuously* to check if neighbors are forwarding the packets correctly or not. In the promiscuous listening mode, MNs capture all packets they can receive, not just packets addressed directly to them. If a neighbor repeats any misbehavior more times than a predefined threshold value, the observer notifies the source node of this by sending a message. This information is collected by the pathrater located at each MN, which maintains a rating for every other MN. This rating is used in calculating the reliability of paths to avoid using misbehaving MNs when selecting a route. As indicated by the authors, this scheme has several weaknesses. First, since the focus is not on cooperation but on network throughput, detected selfish or malicious nodes are just bypassed by detouring in forwarding paths and remain unpunished. As a result, being selfish becomes a blessing to MNs themselves. Second, the promiscuous listening mode may not work for all cases. For promiscuous mode operation, wireless links need to be bidirectional, while recent topology control techniques allow some unidirectional links to be present. Also, the development of directional antennas makes it difficult for the watchdog to overhear neighbors' traffic. Third, if any collision occurs during overhearing as in Fig. 2, the watchdog cannot know if the collision is due to its neighbor's misbehavior or because another neighbor transmits a packet at the same time. Lastly, each MN requires large storage to buffer packets until proper forwarding by its neighbor is confirmed. These stored packets are used for a comparison with packets forwarded by its neighboring MN to check and ensure if the neighbor transmits correct data.

Whereas the work of Marti *et al.* [6] is focused on data forwarding, a context-aware inference mechanism in [9] provides a way to detect misbehavior in the DSR route discovery process as well. After route discovery, it also keeps selfish MNs from falsely reporting link breakage errors to avoid packet forwarding. For misbehaving

MNs, special security messages (SECMs) are sent to the source. Meanwhile, to prevent undue punishment by false accusation, source MNs do not convict the accused MNs until the same accusation comes from more than three neighbors.

Cooperation of Nodes: Fairness in Dynamic Ad Hoc Networks (CONFIDANT) [10] is also a reputation-based scheme. Each MN adopting CONFIDANT has four components: *monitor*, *trust manager*, *reputation system*, and *path manager*. The monitor is similar to the watchdog in [6], which not only promiscuously listens to the transmission of neighboring MNs but also observes route protocol behavior. If any misbehavior is detected, the trust manager sends ALARM messages to all its friend MNs, and trust managers receiving this ALARM determine the trustworthiness of the message based on the trust level of the sender. Using this information, the reputation system maintains a local rating list and a black list, and exchanges these lists with friends. The local rating list has the trust level of each MN, and the black list contains a list of MNs that should be avoided for routing. Also, each MN does not forward a route request originated by a node in its black list. This is an important difference from the watchdog system [6] whereby selfish MNs are just circumvented and not punished. Finally, the path manager, similar to the pathrater in [6], ranks paths according to the reputation of MNs along the path and deletes paths containing malicious MNs. Like the watchdog system, the monitor cannot exactly tell malicious behavior from mere coincidence such as collision. Besides, scalability is another problem due to key validation and certification in the trust manager.

CORE [11] enforces cooperation by a collaborative monitoring technique: it utilizes positive reports by other MNs (indirect reputation) as well as its own observation (subjective reputation) for neighbors, as shown in Fig. 3. Each MN also monitors task-specific behavior (functional reputation) of neighbors; for example, how to handle route request messages. These three types of reputation are used to determine a combined reputation, and MNs with reputation below a threshold value are isolated from networking. They, however, can rejoin a MANET if they increase reputation by cooperating well for a period of time. It is noteworthy that only positive reports for other MNs can be spread as indirect reputation. Hence, malicious nodes cannot falsely accuse other MNs. Also, to mitigate the effect of possible wrong detection by each MN itself, an aging factor is adopted while determining the combined reputation. More weight is given to past observations so that reputation of MNs may not be changed frequently. This protects those MNs that temporarily suffer from bad environmental conditions from being punished severely.

Yau *et al.* [12] identify many problems that make CONFIDANT and CORE impractical to use in actual MANETs. First, quantifying reputation is difficult because a single behavior may be regarded as good or bad depending on the role of the MN. Second, it takes much overhead to reliably distribute reputation in a MANET. Besides, the trustworthiness of distributed information cannot be guaranteed due to false accusation of malicious MNs or collusion between them. Third, most reputation systems cannot work against a *spoofing attack* whereby an MN changes its ID continuously. Finally, the promiscuous listening mode is never a reliable way to observe misbehavior of neighboring MNs. A simple reputation system is also proposed to enhance the robustness of a MANET by an MN utilizing only its own experience about all neighboring MNs. Whenever sending service requests to neighbors, each MN modifies the reputation of neighbor MNs based on their responses. Through prudent evaluation, this scheme makes malicious MNs difficult to build up good reputation in a short period of cooperation.

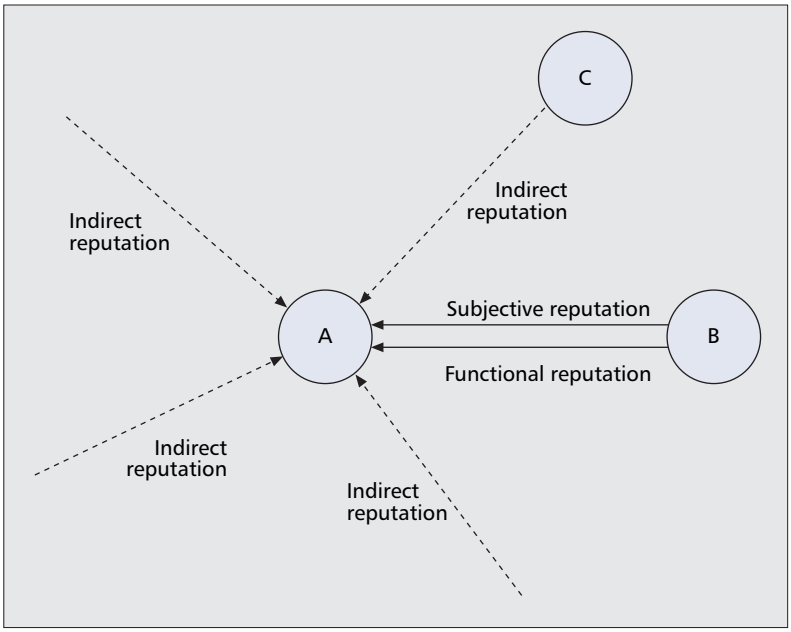


Figure 3. *Reputation of MN B combined in MN A using CORE.*

Miranda *et al.* [13] suggest an approach allowing MNs to publicly declare lists of MNs for which they will not provide service. Using a control message, each MN, say *i*, advertises three sets of MN IDs:

- *Friends* to which MN *i* is willing to provide service
- *Foes* for which MN *i* refuses to serve
- *Selfish* that regard MN *i* as a foe

MNs forward packets only from their friends, not from foes, and they do not send packets to selfish MNs. At first, all other MNs are in the friends set, but they are gradually moved into other sets according to their behavior. In order for MN *i* to determine if any MN should be inserted into its foes or not, it maintains *credits* for each MN in a network, which indicates the difference between the numbers of packets forwarded for each other. If the credits for any MN reach a predefined maximum value (i.e., if an MN does not provide service to MN *i* but receives unilaterally), the MN is put into the foes of MN *i*. For managing the selfish set, MN *i* should also promiscuously watch how its neighbors treat packets forwarded by MN *i*. If a neighbor does not process the packets fairly, MN *i* adds the neighbor MN into the selfish set. This

CORE places more weight on past observation to protect some MNs from wrong punishment based on unintentional errors. Due to this conservative punishment policy, however, some MNs can behave selfishly for a long time after building up a good reputation.

scheme requires a large memory space to maintain a variety of status variables such as credits, friends and foes of other MNs, and recently forwarded packets when neighbors drop the packets. It also has a constraint that every pair of nodes should have at least one route consisting of only well behaved nodes.

Most previous approaches utilize the promiscuous listening mode to watch neighbor MNs' behavior. This is called *passive acknowledgment* as opposed to *active acknowledgment* using an ACK message. As stated earlier, passive acknowledgment may not work correctly in the presence of ambiguous collisions, unidirectional links, and partial dropping. In order to overcome these problems, TWOACK and S-TWOACK [14] schemes are proposed. Used only with a source routing protocol, they require an MN to send an acknowledgment packet named TWOACK to the MN through which a data packet passed two hops before. Whereas an acknowledgment is required for every data packet in the TWOACK scheme, S-TWOACK allows one acknowledgment for several consecutive packets, similar to go-back-N and selective repeat automatic repeat request (ARQ) in TCP. However, there are many issues that need to be addressed. First of all, they cause large message overheads. Although S-TWOACK may reduce message overheads, any sequence number for every session has to be maintained for S-TWOACK to work correctly, which means all intermediate nodes have to perform the transport layer operation during forwarding. Besides, if an MN does not receive a TWOACK within a timeout period, it cannot determine which is the misbehaving MN, the next hop MN or the next-to-next MN. Thus, the proposed approaches maintain the number of times of misbehavior for every link, resulting in larger storage overheads than where the number is maintained for every MN. Although some MNs are convicted of misbehaving, they are just detoured in route selection and re-enter the network after a certain period of time. Thus, there is no reason for MNs not to behave selfishly.

As mentioned before, CORE places more weight on past observation to protect some MNs from wrong punishment based on unintentional errors. Due to this conservative punishment policy, however, some MNs can behave selfishly for a long time after building up a good reputation. On the contrary, reputation indexing window (RIW) [15] emphasizes current behavior feedback rather than old values. The easiest way to do this is to use an aging factor α as

$$Repu_{new} = \alpha \times CurrFeedback + (1 - \alpha)\, Repu_{old}. \quad (1)$$

This enables isolated MNs to easily recover their reputation within a short period of cooperation. This drawback can be mitigated by assorting feedback items (FIs) into three windows, RIW_1, RIW_2, and RIW_3. RIW_1 has the latest ones and RIW_3 the oldest ones; and the ratio of size of RIWs is $RIW_1 : RIW_2 : RIW_3 = 10 : 30 : 60$. If the latest FI arrives, all FIs in RIWs are pushed one step toward the oldest FI's position, and the oldest is discarded. The overall reputation is computed as

$$Repu = \lambda RIW_1 + \mu RIW_2 + \nu RIW_3. \quad (2)$$

It is claimed that this equation responds to the latest changes in a better way when $\lambda = 0.66$, $\mu = 0.22$, and $\nu = 0.11$.

The aforementioned reputation-based methods assume that reputation computed by each MN is propagated in a secure manner, and reputation agreement is easily achieved between all MNs. However, this assumption is not always true in a wireless mobile environment, and the speed of reputation propagation has an important effect on the convergence speed of the reputation agreement. Liu *et al*. [16] analyze the methods for calculating and updating the reputation of other MNs and show that MNs attain reputation agreement through localized propagation, if the frequency of propagation is often enough and MNs use their own experience as part of the update.

Credit-Payment Scheme

Buttyán *et al*. [17] first introduced the commercial transaction concept into a selfishness prevention model. MNs providing service to other MNs receive virtual currency or credit, and MNs benefiting from the service are charged for it. In their two approaches, the packet purse model (PPM) and packet trade model (PTM), MNs forwarding packets for others are given virtual currency called *nuglets*,[2] which is used for those MNs to send and receive their own packets later. When any MN gains nuglets, which MN is charged for the nuglets is different depending on the type of model. In PPM, nuglets are loaded into packets by the source MN and deducted by intermediate MNs as a reward for packet forwarding service. Since packets without sufficient nuglets will be discarded at intermediate MNs, the source MN has to know how many nuglets are needed until those packets arrive at the destination. Thus, the use of PPM is limited to a source routing protocol like DSR. Also, forwarding MNs may take out more nuglets from packets than they are supposed to, and possibly may not forward packets after taking out nuglets. On the other hand, in PTM each intermediate MN buys packets from the previous MN and sells them to the next-hop MN at a higher price. After all, the destination MN should pay the final price to its predecessor. While PTM is not limited to use in source routing protocols, the MANET may be an easy target for DoS attack because MNs can freely originate packets without paying any currency. Both PPM and PTM have another limitation: the question of how to trust the validity of nuglets. MNs may not only reuse nuglets already used one time but also increase their nuglets at their own will. It is assumed that each MN has an arbitrary tamper resistant security module like a special chip or a smart card, but this makes it difficult for them to be widely accepted. A recent work [18] employs a public key algorithm in the security module and illustrates how each MN can maximize its benefits by using PPM or PTM based on credit counters in a network with selfish MNs.

The PPM method is analyzed and adopted with a little modification by the ad hoc-VCG

[2] *The name of nugget was originally used in their first article, and it was changed into nuglet afterward.*

scheme [19], which is a DSR-like reactive routing protocol that pursues cost efficiency and truthfulness of paths. It consists of two phases: *route discovery* and *data transmission*. In the route discovery phase, destination MNs compute the amount of payments for intermediate MNs and notify the source MN or central bank of this information according to who the payer is, and the payment is actually performed during data transmission. Ad hoc-VCG, however, completely depends on the destination MN for how much credit should be given to intermediate MNs. Thus, if the destination does not exactly compute payment or does not report it to the source or central bank, there is no way for intermediate MNs to get their fair share. Since all MNs are assumed to be selfish, destination MNs have two motives to not report the amount of payments in the central bank model. First, transmission energy is required to report to the central bank. Second, the central bank compensates for the premiums it has given to the intermediate MNs by debiting accounts of all MNs evenly. Thus, if a destination MN does not report that the central bank should pay a premium to intermediate MNs, the amount of premium the bank has to collect is reduced as much, resulting in an advantage for the destination MN itself.

Sprite (a simple, cheat-proof, credit-based system) [20] also utilizes credit to give incentive to the MNs that forward packets. Not relying on any tamper-proof hardware, Sprite needs a central authority server called a credit clearance service (CCS) to maintain credit balance, which is a fixed system outside a MANET. Every MN keeps a *receipt* whenever it receives a packet and reports to CCS when it has a fast connection to CCS. After collecting reports, CCS rewards both the last MN of each path for report transmission and all intermediate MNs for packet forwarding, instead of charging the source. However, the amount of credit charged to the source is not always equal to that given to other MNs. In order to prevent cheating with false receipts and encourage MNs to cooperate, CCS deducts more credits from the source than it should give to others and uses them to induce MNs to make a true report. Under the Sprite algorithm, neither false reporting nor collusion between more than one MN gives more credits to MNs. Scalability can be a problem because MNs report a receipt for every message to CCS, and messages are encrypted using public/private key pairs. As CCS has to know the complete path between two MNs, Sprite can be used only with a source routing protocol.

Salem *et al.* [21] extend the credit payment scheme to multihop cellular networks that combine cellular and mobile ad hoc networks. Similar to a typical cellular network, many BSs cover an area, and are connected via a backbone between them. However, all MNs need not have a direct link to a BS for communication; instead, the communication between an MN and a BS is generally relayed by other MNs. This scheme requires all packets to pass through a BS, not allowing direct communication between any two MNs. This requirement reduces routing overheads of each MN, because it is enough for each MN to have only one route to a BS. In the proposed model, actually, a BS maintains a route to every MN in its own cell. This research also assumes the presence of a trusted operator that manages the billing accounts of all the MNs. Figure 4 illustrates the packet transmission from the source *S* to the destination *D*. Once a packet reaches BS_s from source *S*, the trusted operator deducts as many credits from the account of *S* as it should give to all forwarding MNs. As a BS has exact route information to every MN, the required amount of credits can be accurately computed; besides, collusion between MNs cannot work. When a packet arrives at BS_s, part of the credits are practically given to N_s. Similarly, when the packet reaches destination *D* and *D* responds with an acknowledgment to BS_d, credits are paid to N_d. As provision against the case where *D* does not send an acknowledgment to conserve its power, the trusted operator takes away some credits from the account of *D* before sending a packet to *D*. These credits are returned to *D* when its acknowledgment arrives. Using symmetric cryptography, communication between MN and BS is protected.

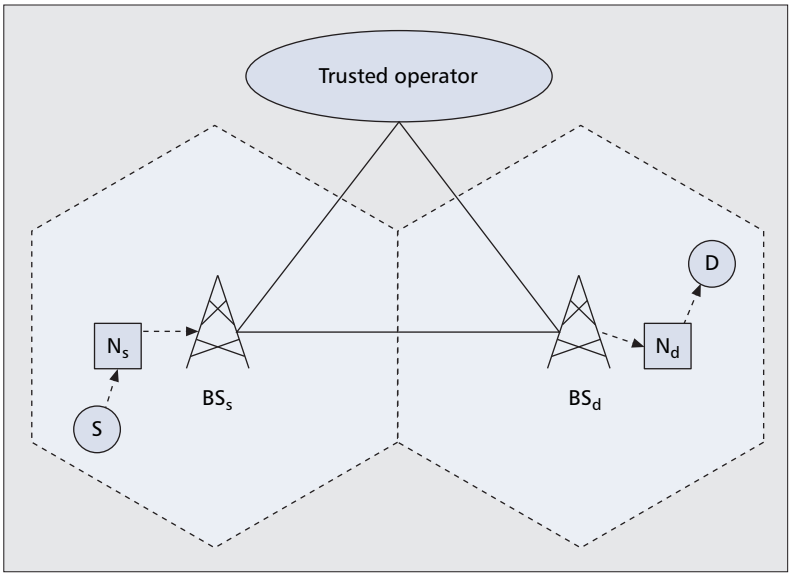


Figure 4. *Incentive scheme in a multihop cellular network.*

The aforementioned credit-payment approaches require global participation; that is, all MNs should follow the selfishness prevention protocol in order for the MANET to work well. In a practical MANET, however, a variety of heterogeneous MNs may exist, and some of them may not know of the presence of any credit-payment method and how to handle those credits. Furthermore, some MNs may not earn enough credits, not because they are selfish but because they are just badly positioned to be used as forwarders. To solve this problem, Raghavan *et al.* [22] propose two layered forwarding services: *priced priority forwarding* and *free best effort forwarding*. Whereas MNs should pay some credits to intermediate MNs for priority forwarding, they do not need any credit to use the best effort forwarding service. Intermediate MNs forward priority packets before any best effort traffic, and this forwarding behavior is watched by their neighbors in a promiscuous way. If they do not

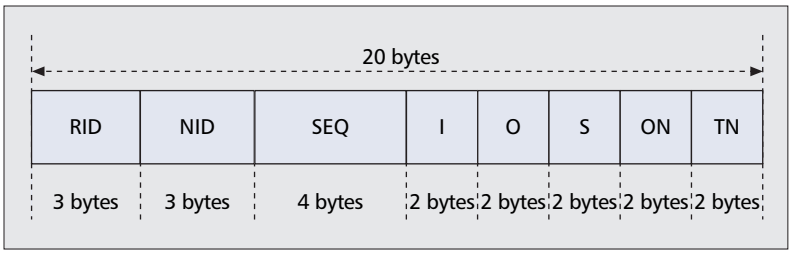


Figure 5. *Report message in PIFA.*

support priority forwarding properly, the payment for the priority service is void. In order to manage the credits for all MNs, the concept of CCS is borrowed from [20]. A minor difference is that the CCS is a general MN here, while it is a fixed authorization in the Sprite system. However, it is not clearly stated how an MN can keep track of credits for all other MNs.

Crowcroft *et al.* [23] first assume a set of MNs that are equipped with directional antennas. Route selection and flow allocation are decided based on current congestion price indicated by relevant MNs. The price takes both bandwidth and power into account at the same time. Each MN s first determines a willingness to pay at time t, $w_s(t)$, which means how many credits it is willing to pay for its traffic. Afterward, it finds the minimum cost path by summing up the cost advertised by all MNs along each path. Using willingness to pay and cost of the minimum cost path, MNs adjust their resource usage accordingly. Although the price is defined to be adapted depending on available bandwidth and power, there is no practical way to prevent MNs from declaring arbitrarily higher prices.

Wang *et al.* [24] argue that it is too naive for each MN to determine price independently, and a truthful multicast routing protocol has been designed with each MN truthfully reporting its cost in multicast routing structures. However, this has some strong assumptions. First, the network should be *bi-connected*, which means the network should not be partitioned by removing one MN. Second, all receiver MNs should relay packets for peer receivers for free whenever they are asked. Finally, all MNs always try to maximize profits by forwarding packets. In a practical world, however, some MNs that no longer need any credits may not participate in packet forwarding.

Most previous credit-payment approaches are compatible only with source routing protocols like DSR, as they require the complete path information from the source to the destination. Although PTM [17] and research in [21] are not limited to source routing, they need the help of tamper-proof hardware or BS instead, which is usually not provided in a MANET environment. On the other hand, Protocol-Independent Fairness Algorithm (PIFA) [7] is compatible with any type of routing protocol, being implemented as a simple add-on to them. PIFA needs a server node called a credit manager (CM), which manages all MNs' credits. Other MNs periodically send report messages (shown in Fig. 5) to the CM on the number of packets they forwarded in each time interval; the CM verifies the credibility of the reports and rewards forwarding MNs according to the results. An MN with either the maximum amount of power or an AP to the wired network, like a BS in sensor networks, can serve as the CM node. The CM node is assumed to be specially managed by an administrator and is considered trusted. The credibility test by the CM is to check whether or not reports from two neighbor MNs match each other. If a pair of MNs send inconsistent reports, both of them are given one penalty point called numbers of alleged manipulation (NAM). When the NAM of an MN crosses a threshold value, the MN is isolated from other MNs. The NAM of an MN is reduced when its peer MN makes another inconsistent report with another MN, based on the concept that a selfish MN will repeatedly attempt to deceive others. Due to the NAM, the credit-payment scheme PIFA is said to have a feature of reputation-based approaches as well.

Game Theory

Game theory is a branch of economics related to deriving the optimal strategy for every rational competitive player. The objective is to look for the *Nash equilibrium* point where a player cannot increase his or her payoff by changing strategies while other players' strategies remain fixed. Primarily, it has been used in many fields of social economy, including policy on taxation, design of roads, and development of transportation networks. In recent days the use of game theory has been extended to a variety of areas in communication networks including packet forwarding. It is worth noting that game theory assumes all players are rational. In other words, if some MNs do not have any interest in increasing their profits but just concentrate on power saving, the game theory may not be able to model a selfish user network exactly. Assuming rational MNs, all the following methods model packet forwarding as a strategic game so that the found forwarding rate may be a Nash equilibrium for every MN. Each algorithm in this section is based on different complicated equations, but there is a common notion behind them: Under the assumption that all MNs can change their strategies freely, if selfish behavior of an MN is detected, all other MNs will change their strategies to punish it. Thus, the game will be gradually stable at one point where all MNs are satisfied.

Generous Tit-For-Tat (GTFT) and multiple-GTFT (m-GTFT) [25] are the first relay acceptance algorithms to use game theory in MANETs. GTFT is for the case where all requests are relayed by just one MN until they reach the destination, and m-GTFT is for when multiple relays exist between the source and the destination. These algorithms are for a node to balance the energy consumed for other MNs with the energy used by others for itself; and to find an optimal trade-off between blocking probability and power consumption. Also, they pursue energy consumption fairness for a session as a unit; thus, computation complexity is remarkably reduced from previous algorithms with a packet as a unit. In order to reflect the practicality of self-organization, GTFT and m-GTFT

P1 \ P2	Confess	Not confess
Confess	(5, 5)	(0, 10)
Not confess	(10, 0)	(1, 1)

Table 1. *Prisoners dilemma.*

assume the presence of heterogeneous MNs: each MN has different amount of resources depending on its type (e.g., laptop, PDA, or cell phone). For MN h to decide whether or not it accepts a relay request, it maintains two variables $\phi_h^j(k)$ and $\psi_h^j(k)$: $\phi_h^j(k)$ is the ratio of the requests successfully relayed by others out of total requests generated by MN h for session type j till time k, and $\psi_h^j(k)$ is the ratio of the requests relayed by MN h out of total requests reaching MN h for session type j. MN h rejects a relay request only if

$$\psi_h^j(k) > \tau_j \text{ or } \phi_h^j(k) < \psi_h^j(k) - \varepsilon, \quad (3)$$

where ε is a small positive number. In the first part, τ_j denotes the maximum relay ratio for session type j to fit with the amount of traffic allowed for j. The second part of the equation means that the number of requests relayed by MN h should be greater than the number of requests relayed by others for h. The value ε depicts the generosity of each MN, which indicates that an MN relays more requests than they should at some degree without any compensation. This research shows that a Nash equilibrium exists in a MANET using the GTFT algorithm. However, each MN needs sufficient information about an entire system such as the number of MNs, energy constraints of them, and requests for each session in order to derive values of $\phi_h^j(k)$, $\psi_h^j(k)$, τ_j, and an optimal ε. For this a distributed mechanism is required, but no algorithm is provided that can prevent MNs from disseminating incorrect information for their own interests.

Urpi *et al.* [26] consider a constraint on energy consumed by sending their own packets as well as forwarding others' packets. Although an MN has a packet to send, if it presumes that its neighbor will probably not forward its packet for some reason, it selects the strategy not to send the packet. However, some strong assumptions and a naive definition of payoff for each MN are drawbacks that are difficult to overcome.

Michiardi *et al.* [27] propose two methods to analyze the CORE algorithm [11] from the perspective of a cooperative game approach and a non-cooperative game approach. In a cooperative game, players reach an agreement through communication, while all players in a non-cooperative game chase their own profits independently. The most famous example of a non-cooperative game is the Prisoners Dilemma (PD). Table 1 shows prison terms for prisoners P1 and P2 if each of them confesses or not. While each prisoner is separated in a different room and not aware of the other's strategy, confession is the best strategy for him/herself. Thus, "confess-confess" is the Nash equilibrium. On the other hand, if they can cooperate, both of them can get the best profit. The proposed cooperative game approach extends the PD game to N-prisoners dilemma. It is assumed that the payoff increased by one more MN's cooperation is greater than the increased cost and that all MNs share the payoff. In the non-cooperative game approach, behavior of each MN is represented as the ratio of energy E_{self} consumed for itself to the sum of E_R and E_{PF} consumed for routing and forwarding for others. The utility function is based on the difference between E_{self} and (E_R + E_{PF}), and it is adjusted depending on the importance of the power. All MNs try to maximize their own benefit while maintaining reputation.

Catch [28] attacks both the connectivity and forwarding avoidance problems of selfish MNs at the same time. Catch uses anonymous messages in which sender identity is hidden to check true connectivity with neighboring MNs. If an MN does not send an acknowledgment after receiving the anonymous message, connectivity is dropped. Although selfish MNs want to hide as many connectivities as possible in order to avoid being used as relays, they should have at least one connectivity to send their own packets. However, since they cannot grasp the sender of the anonymous message, they have no choice but to acknowledge connectivities to all neighbors. In addition, forwarding avoidance of an MN is prevented by the watchdog systems in neighbor MNs. Cheaters are isolated from all other MNs. Using game theory, the authors attempt to prove that Catch induces cooperation to be an evolutionarily stable strategy (ESS). The concept behind this proof is as follows: After setting up all strategies MNs may adopt, a single round game is designed. In this game payoff for each MN depends on currently determined strategies of all MNs; and they cannot change strategies anymore. However, if it is defined as a repeated game whereby they can change their strategies every round of the game while not knowing when the entire game will end, they would consider the consequence of their actions and try to increase their interests by changing strategies. A cheater in a round will be punished by all other MNs in the next round. Similarly, prisoners in the PD game will be evolutionarily stable at the "not confess-not confess" strategy. This research applies game theory to develop of mechanisms that enable inter-ISP (Internet service provider) coordination as well.

Selfish Link and Behavior Adaptation to Produce Cooperation (SLAC) [29] for peer-to-peer networks is based on the repeated PD game. Since this research assumes that every peer MN can freely change its strategy and select its partner, cooperative MNs try to find another cooperative one as their peer, resulting in isolation of selfish MNs. However, SLAC does not address how each MN fairly compares its performance against another node.

Game theory is also used to design incentive scheduling for cooperative relay MNs in WWANs/WLANs [30]. The idea is very simple: a BS allocates more time slots or more power to relay MNs than non-relay MNs in order to encourage cooperative relay. If an MN wants to serve as a relay, it sends relay advertisement

Catch attacks both the connectivity problem and the forwarding avoidance problem of selfish MNs at the same time. Catch uses anonymous messages of which sender identity is hidden, to check true connectivity with neighboring MNs.

Name	Type	Manage	Feature	Limitation
Watchdog [6]	R	Distributed	Detouring selfish MNs, not punishing them	Dependence on promiscuous listening
Context-aware [9]	R	Distributed	Misbehavior detection in the route discovery process as well	Offline agreement on a secret number
CONFIDANT [10]	R	Distributed	Selfish MNs isolated	Dependence on promiscuous listening
CORE [11]	R	Distributed	Collaboratively monitoring neighbor MNs	Slow reaction to MNs' behavior
Local reputation [12]	R	Distributed	Utilization of only self-experience to evaluate reputation	Ignorance of non-neighboring MNs
Friends and foes [13]	R	Distributed	Individual relation between two MNs in reputation management	Large memory overhead
TWOACK [14]	R	Distributed	Acknowledgment for transmission between MNs two hops away	Large message and memory overhead
RIW [15]	R	Centralized	Three-window weighted average for reputation to smooth change of MN status	Arbitrary weight without a theoretical base
PPM/PTM [17, 18]	C	Distributed	First source- and destination-charge model for packet transmission	Tamper-proof hardware for security
Ad hoc-VCG [19]	C	Centralized	Two phases of cost calculation and payment for relays	Dependence on destination's report
Sprite [20]	C, G	Centralized	Collusion prevention as well	Scalability issue w/ message overhead
Multihop cellular [21]	C	Centralized	Combined architecture of cellular network and MANET	Indirect communication between MNs
Priority forwarding [22]	C	Centralized	Two-layered service: free best-effort forwarding and priced priority forwarding	Dependence on an MN as a credit server
Willingness to pay [23]	C	Distributed	Adaptive price depending on the status of resources	Naive trust in each MN on the cost
Truthful multicast [24]	C, G	Distributed	Encouragement for truthful reporting in multicast routing tree	Only bi-connected networks
PIFA [7]	C, R	Centralized	Full compatibility to any types of routing	Dependence on an MN as credit server protocol
GTFT [25]	G	Distributed	Generous MNs for others' selfishness to some degree	Need for much system information
Catch [28]	G	Distributed	Sender ID of packets hidden	No proof of evolutionary stability
SLAC [29]	G	Distributed	Prisoners dilemma in P2P network	Need to compare MNs' performance fairly
Incentive scheduling [30]	G	Centralized	More time slots and power for relay MNs than non-relay MNs	Relay MNs actually not relaying packets
Token-based [31]	—	Distributed	A partial of the system secret shared by each MN	Not on a sparse or high-mobility network
AD-MIX [32]	—	Distributed	Destination ID of packets hidden	Longer path by deliberate loopback
SMT [33]	—	Distributed	Redundant data via multiple paths	Increased amount of traffic

(R: Reputation-based, C: Credit-based, G: Game theory)

Table 2. *Selfishness prevention schemes.*

messages to other MNs and registers itself on the BS. The proposed method just concentrates on improving throughput, not considering energy consumption.

Sprite [20] and the work done by Wang *et al.* [24] can be classified into this category also as they determine the amount of charge and credit so that truth telling may be the optimal strategy for all MNs.

Other Schemes

In this section a variety of selfishness prevention schemes are considered that are not included in the earlier three major classes.

Yang *et al.* [31] propose a token-based protocol whereby an MN should hold a valid token to participate in networking. The token of each MN has a period of validity; thus, the token has to be refreshed from neighboring MNs before its expiration. Each MN in the network has a portion of the system secret. If they get other portions from enough neighbor MNs, they can renew their tokens. The proposed algorithm consists of four parts: two proactive mechanisms, *Neighbor Verification* and *Security Enhanced Routing Protocol*, for secure routing; and two reactive mechanisms, *Neighbor Monitoring* and *Intrusion Reaction*, for cooperative packet forwarding. In Neighbor Verification the expiration time of each MN's token is determined. The time is proportional to the duration of an MN behaving well in the network; thus, a well behaved MN does not have to renew its token frequently. For the Security Enhanced Routing Protocol, the authors implement AODV-S in which all MNs maintain a list of all their verified neighbors with valid tokens. For secure routing, each MN also maintains the route entries announced by its neighbors as well as its own routing table. Meanwhile, the Neighbor Monitoring part in each MN promiscuously observes its neighbors all the time and records the headers of the packets it has overheard recently. If a cheater is detected, the detector sends a single intrusion detection (SID) to other MNs. Finally, Intrusion Reaction revokes the token of a cheater. If the number of issued SIDs reaches a threshold value, the corresponding MN is deprived of network access forever because it cannot receive a token again. Although this method has a lot of interesting features, it requires much memory space and processing overhead in each MN, which are very limited resources for wireless MNs. Besides, sparse density of MNs and high mobility degrade the performance of this method since they make it difficult for an MN to find a sufficient number of neighbors.

The AD-MIX protocol [32] discourages selfish MNs from dropping packets by hiding final destination addresses of the packets. Because some packets forwarded by them can be returned to themselves in the end, even selfish MNs do not drop packets. AD-MIX, however, deliberately forces a path to have loopback, resulting in longer path length and more resource consumption, thereby shortening the lifespan of the overall MANET.

The Secure Message Transmission (SMT) protocol [33] utilizes multiple paths. Instead of handling or punishing selfish MNs directly, it circumvents them by sending redundant data via multiple paths simultaneously. If only some parts of the redundant data arrive successfully, the destination can reconstruct the message. According to the ratio of successful packet arrival, each of multiple paths is endowed a rate of reliability. A path with a too low rate is excluded from the candidate path set. SMT is a good fit for supporting quality of service requirements of real-time traffic.

After Pretty Good Privacy (PGP) [34, 35] was suggested, several researchers proposed methods to evaluate trust of each user in a network. Although they are not selfishness prevention schemes, it is worth introducing them as they help MNs use more reliable paths and improve performance using trust evaluation information. Watchdog [6] and TWOACK [14] also bypass unreliable MNs, rather than force them to cooperate with others. In PGP all entities play as certificate authorities; thus, they can verify and sign others' public keys independently. Instead of avoiding defects from a centralized authority architecture, each entity should have a directory to store certificates for all other entities. Each PGP user evaluates others in four levels of trustworthiness of a certificate and its issuer.

Capkun *et al.* [36] show that the small world phenomenon [37] emerges in the PGP certificate graph. The small world phenomenon means that any pair of people in the United States are connected with a chain through five or six acquaintances. Based on this theory, they observe that every MN can be reached through a chain of a few MNs; the last MN has a certificate for the destination MN, while each MN contains just a limited number of certificates for others. In other words, an MN depends on acquaintances' opinion to trust other MNs.

Similar to PGP, Theodorakopoulos *et al.* [38] suggest two types of opinions: the *trust* value (how much you can trust the quality of this MN's information) and the *confidence* value (how sure I am about this opinion). The objective is to make each MN accurately evaluate the trustworthiness of MNs with which it has not previously interacted through relations with current acquaintances, whereas PGP utilizes only directly assigned trust values. In this research a *trust graph* is introduced that uses all MNs and trustworthiness each MN has for other MNs as nodes and directed edges, respectively. In simulation the authors set up such that misbehaving MNs always have the best opinion for neighbor misbehaving MNs, which implies consideration of collusion.

Pirzada *et al.* [39] subdivide trust into several specific categories. For instance, any MN can be trusted for packet forwarding but not for secure routing. Also, they argue that discrete representation of trust, as in PGP systems, is not adequate to reflect all kinds of features in MANETs. Instead, they allow trust levels to move in a continuous range.

Conclusion and Future Direction

Table 2 summarizes important features of selfishness prevention schemes introduced in this article. Generally speaking, reputation-based

After PGP was suggested, several researchers have proposed methods to evaluate trust of each user in a network. Although they are not selfishness prevention schemes, it is worth introducing them as they help MNs use more reliable paths and improve performance using trust evaluation information.

Many of these problems need to be addressed carefully so that selfish MNs should not be rewarded and attempts should be made to encourage MNs to move away from any potential selfishness characteristics.

schemes are more scalable than credit-based schemes, but may not be used in future MANETs unless an alternative to the promiscuous listening mode can be developed. For credit-based schemes, scalability is one of the most urgent issues, as expected. Game theory methods may surpass both reputation-based and credit-payment schemes in many ways, but MNs have to collect large amounts of information for the entire network. Additionally, all MNs playing a game should attempt to increase their profits continuously without exception.

Whereas many researchers are trying to develop robust and efficient methods, some people raise pessimistic prospects about this research. Huang *et al.* [40] address drawbacks and impracticality of both reputation and credit-payment incentive methods. Illustrating the adoption cycle of MANETs, they even assert that these methods are not needed at all, especially in the early stages of MANET deployment due to the following reasons:

- They cause much overhead in systems.
- MNs on the outskirts of a network may not have a chance to earn credits because other MNs do not use them as relays.
- The credit-based methods are not appropriate for real-time applications since MNs have to buffer packets until they has enough credits.
- It is hard to fairly manage credits.
- It is not clear if the resources saved by selfish behavior can be enough to tamper with MNs, considering the cost.

Just because this characteristic is currently overburdening in MANETs does not mean we should put aside this research. As the operation of MANETs completely relies on voluntary participation of each MN, it is very dangerous to trust all MNs naively. Aside from the final economic reason, we anticipate that all the technical problems could be resolved as time goes by. First, much overhead is due mainly to the centralized architecture of existing approaches. A lot of control messages should be transmitted to a central server periodically. Hence, extensive research has focused on designing self-organized algorithms without a central authority, and actually many efficient schemes have been proposed, although each of them still has a few weaknesses. Second, the problem of badly positioned MNs is not always true, considering node mobility in MANETs. Furthermore, many algorithms try to solve the possible problems by either distributing some credits to all MNs periodically or allowing free best effort forwarding as well as priced priority forwarding. Reputation-based mechanisms do not suffer from the problem in origin. Third, the credit-based method certainly may not be suitable for real-time traffic, but users who want critical real-time service need to establish their own network or purchase credits with real money. For fair management of credits, more and more robust algorithms are emerging every day.

There remain several stumbling blocks that need to be cleared. First, we need a new mechanism which can replace the promiscuous listening in many reputation-based systems. The promiscuous mode cannot be used in MANETs using topology control or smart (directional) antennas. Second, most current algorithms are useless if MNs continuously change their IDs (spoofing attack). It is no use either building up reputation or saving credits associated with a specific ID. Basically, if we want to extend selfishness prevention methods to wireless sensor networks, all sensor nodes ought to have unique IDs, which is against the nature of sensor networks constituted by thousands of sensors. Many of these problems need to be addressed carefully so that selfish MNs should not be rewarded, and attempts should be made to encourage MNs to move away from any potential selfishness characteristics.

Acknowledgment

This work has been supported by the Ohio Board of Regents Doctoral Enhancement Funds.

References

[1] D. P. Agrawal and Q.-A. Zeng, "Chapter 13. Ad Hoc and Sensor Networks," *Introduction to Wireless and Mobile Systems*, Brooks/Cole-Thomson Learning, 2003, pp. 297–348.
[2] N. Jain and D. P. Agrawal, "Current Trends in Wireless Sensor Network Design," *Int'l. J. Distributed Sensor Networks*, vol. 1, no. 1, Jan./Apr. 2005, pp. 101–22.
[3] R. Molva and P. Michiardi, "Security in Ad hoc Networks," *Proc. Pers. Wireless Commun.*, 2003.
[4] H. Deng, W. Li, and D. P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 40, no. 10, Oct. 2002, pp. 70–75.
[5] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Commun.*, vol. 11, no. 6, Dec. 2004, pp. 6–28.
[6] S. Marti *et al.*, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. ACM MobiCom 2000*, pp. 255–65.
[7] Y. Yoo, S. Ahn, and D. P. Agrawal, "A Credit-Payment Scheme for Packet Forwarding Fairness in Mobile Ad Hoc Networks," *Proc. IEEE ICC 2005*.
[8] S. Buchegger and J.-Y. L. Boudec, "Cooperative Routing in Mobile Ad-hoc Networks: Current Efforts Against Malice and Selfishness," *Proc. Mobile Internet Wksp. 2002*.
[9] K. Paul and D. Westhoff, "Context Aware Detection of Selfish Nodes in DSR based Ad-hoc Networks," *Proc. IEEE GLOBECOM 2002*, pp. 178–82.
[10] S. Buchegger and J.-Y. L. Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes: Fairness In Dynamic Ad-hoc NeTworks)," *Proc. ACM MobiHoc 2002*, pp. 226–36.
[11] P. Michiardi and R. Molva, "Core: a Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks," *Proc. 6th IFIP Conf. Sec. Commun. and Multimedia 2002*.
[12] P.-W. Yau and C. J. Mitchell, "Reputation Methods for Routing Security for Mobile Ad Hoc Networks," *Proc. Mobile Future and Symp. Trends Commun. 2003*, pp. 130–37.
[13] H. Miranda and L. Rodrigues, "Friends and Foes: Preventing Selfishness in Open Mobile Ad Hoc Networks," *Proc. Int'l Conf. Distrib. Comp. Sys. Wksps. 2003*, pp. 440–45.
[14] K. Balakrishnan, J. Deng, and P. K. Varshney, "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Commun. and Net. Conf. 2005*.
[15] W. J. Adams, G. C. Hadjichristofi, and N. J. Davis IV, "Calculating a Node's Reputation in a Mobile Ad Hoc Network," *Proc. IEEE Int'l. Perf. Comp. and Commun. Conf. 2005*, pp. 303–07.
[16] Y. Liu and Y. R. Yang, "Reputation Propagation and Agreement in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Commun. and Net. Conf. 2003*, pp. 1510–15.
[17] L. Buttyán and J.-P. Hubaux, "Enforcing Service Availability in Mobile Ad-Hoc Networks," *Proc. ACM MobiHoc 2000*, pp. 87–96.
[18] L. Buttyán and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vol. 8, no. 5, Oct. 2003, pp. 579–92.
[19] L. Anderegg and S. Eidenbenz, "Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad hoc Networks with Selfish Agents," *Proc. ACM MobiCom 2003*, pp. 245–59.

[20] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," *Proc. IEEE INFOCOM 2003*, pp. 1987–97.
[21] N. B. Salem *et al.*, "A Charging and Rewarding Scheme for Packet Forwarding in Multihop Cellular Networks," *Proc. ACM MobiHoc 2003*, pp. 13–24.
[22] B. Raghavan and A. C. Snoeren, "Priority Forwarding in Ad Hoc Networks with Self-Interested Parties," *Proc. Wksp. Economics of Peer-to-Peer Sys. 2003*.
[23] J. Crowcroft *et al.*, "Modelling Incentives for Collaboration in Mobile Ad Hoc Networks," *Proc. WiOpt 2003*.
[24] W. Wang, X.-Y. Li, and Y. Wang, "Truthful Multicast Routing in Selfish Wireless Networks," *Proc. ACM MobiCom 2004*, pp. 245–59.
[25] V. Srinivasan *et al.*, "Cooperation in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM 2003*, pp. 808–17.
[26] A. Urpi, M. Bonuccelli, and S. Giordano, "Modelling Cooperation in Mobile Ad Hoc Networks: A Formal Description of Selfishness," *Proc. WiOpt 2003*.
[27] P. Michiardi and R. Molva, "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad Hoc Networks," *Proc. WiOpt 2003*.
[28] R. Mahajan *et al.*, "Experiences Applying Game Theory to System Design," *Proc. ACM SIGCOMM Wksp. Practice and Theory of Incentives and Game Theory in Networked Sys. 2004*, pp. 183–90.
[29] D. Hales, "From Selfish Nodes to Cooperative Networks — Emergent Link-Based Incentives in Peer-to-Peer Networks," *Proc. IEEE Int'l. Conf. Peer-to-Peer Comp. 2004*, pp. 151–58.
[30] H.-Y. Wei and R. D. Gitlin, "Incentive Scheduling for Cooperative Relay in WWAN/WLAN Two-Hop-Relay Network," *Proc. IEEE Wireless Commun. and Net. Conf. 2005*.
[31] H. Yang, X. Meng, and S. Lu, "Self-Organized Network-Layer Security in Mobile Ad Hoc Networks," *Proc. ACM Wksp. Wireless Security 2002*, pp. 11–20.
[32] S. Sundaramurthy and E. M. Belding-Royer, "The ADMIX Protocol for Encouraging Participation in Mobile Ad Hoc Networks," *Proc. IEEE Int'l. Conf. Network Protocols 2003*, pp. 156–67.
[33] P. Papadimitratos and Z. J. Haas, "Secure Data Transmission in Mobile Ad Hoc Networks," *Proc. ACM Wksp. Wireless Sec. 2003*, pp. 41–50.
[34] P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, 1995.
[35] The International PGP Homepage, http://www.pgpi.org
[36] S. Capkun, L. Buttyán, and J.-P. Hubaux, "Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph," *Proc. ACM New Sec. Paradigms Wksp. 2002*, pp. 28–35.
[37] S. Milgram, "The Small World Problem," *Psychology Today*, vol. 61, 1967.
[38] G. Theodorakopoulos and J. S. Baras, "Trust Evaluation in Ad Hoc Networks," *Proc. ACM Wksp. Wireless Sec. 2004*, pp. 1–10.
[39] A. A. Pirzada and C. McDonald, "Establishing Trust in Pure Ad-hoc Networks," *Proc. 27th Conf. Australasian Comp. Sci. 2004*, pp. 47–54.
[40] E. Huang, J. Crowcroft, and I. Wassell, "Rethinking Incentives for Mobile Ad Hoc Networks," *Proc. ACM SIGCOMM 2004*, pp. 191–96.

BIOGRAPHIES

YOUNGHWAN YOO [S'00, M'04] (ymomo@ececs.uc.edu) received B.S. and M.S. degrees in computer engineering from Seoul National University, Korea, in 1996 and 1998, respectively, and a Ph.D. degree in electrical engineering and computer science from the same university in 2004. Since 2004 he has been working with the OBR Center for Distributed and Mobile Computing, University of Cincinnati, as a post-doctoral researcher. He has served as a reviewer for a variety of journals and conferences. His research interests include routing and security issues in ad hoc and sensor networks, WDM, ATM, and the Internet.

DHARMA P. AGRAWAL [M'74, SM'79, F'87] (dpa@ececs.uc.edu) is the Ohio Board of Regents Distinguished Professor of Computer Science and Engineering and founding director for the Center for Distributed and Mobile Computing in the Department of ECECS, University of Cincinnati, Ohio. He has been a faculty member at North Carolina State University, Raleigh (1982–1998) and Wayne State University, Detroit, Michigan (1977–1982). His current research interests are energy-efficient routing and information retrieval in ad hoc and sensor networks, QoS in integrated wireless networks, use of smart multi-beam directional antennas for enhanced QoS, various aspects of sensor networks including environmental monitoring, and secured communication in ad hoc and sensor networks. His co-authored textbook *Introduction to Wireless and Mobile Systems* (Thomson) has been adopted throughout the world and revolutionized the way the course is taught. He is an editor for the *Journal of Parallel and Distributed Systems* and *International Journal of High Speed Computing*. He has served as an editor of *IEEE Computer* and *IEEE Transactions on Computers*. Recently, he has been invited to serve as a founding member of the editorial board of three new journals: *International Journal on Distributed Sensor Networks*, *International Journal of Ad Hoc and Ubiquitous Computing*, and *International Journal of Ad Hoc & Sensor Wireless Networks*. He has been Program Chair and General Chair for numerous international conferences and meetings. He has received numerous certificates and awards from the IEEE Computer Society. He was awarded a Third Millennium Medal by the IEEE for his outstanding contributions. He has also delivered keynote speeches for five international conferences. He has four patents and 16 patent disclosures in wireless networking. He has been selected as a Fulbright Senior Specialist for a term of five years. He is a Fellow of the ACM, AAAS, and WIF.