

The Controversy of Selfish Nodes in Ad Hoc Networks

C.K. Toh*, Dongkyun Kim**, Sutaek Oh***, Hongseok Yoo**

* University of Hong Kong, Hong Kong

** Kyungpook National University, Korea

***IDIS, Korea

ck_away@hotmail.com, dongkyun@knu.ac.kr, stoh@idis.co.kr, hsyoo@monet.knu.ac.kr

Abstract—Currently, there are work related to selfish nodes in ad hoc mobile networks. In this paper, questions on the viability, validity, and presence of selfish nodes are raised and discussed. Various definitions and interpretations of the term “selfishness” are also presented, along with ways of determining and evaluating selfishness. In this paper, we introduce SCNP - Selfish Check Negotiation Protocol to allow nodes to negotiate for collaboration whenever possible. We also discuss the “impact” of being selfish and unselfish, especially on collaborative applications, and network communication performance. We discuss the relationship of selfishness with security and denial of service. Finally, we compare the pros and cons between punishment-based and incentive-based approaches. The incentive-based approach is advocated since it encourages nodes to be less selfish and more cooperative. Cooperativeness is a necessity for supporting user communications and service provisioning in an ad hoc wireless network.

Keywords— Selfish Nodes, Selfishness, Ad Hoc Networks

I. INTRODUCTION

In ad hoc mobile networks, nodes perform packet forwarding and routing. Packets initiated by a source node are forwarded to neighbouring nodes, which are then further relayed wirelessly towards the destination. This packet forwarding and routing process by other nodes is itself a collaborative behaviour. Without this function, packets could end up not delivered to their destinations.

In one extreme (which we termed as allcollaborative), all nodes in the ad hoc network are willing to forward packets for others so as to achieve the maximum possible packet delivery rate. For example, routers in our existing Internet are usually fully cooperative, building up-to-date routes to all possible destinations and relaying packets to nexthop routers. The all-collaborative scenario is not entirely fictitious since an ad hoc wireless network used by a government or law-enforcement agency can potentially consist of fully collaborative nodes in order to fulfil specific communication needs in an emergency or tactical situation. On the other extreme, if nodes are fully un-collaborative (which is unlikely), then the ad hoc mobile network is practically

useless since packet transmissions (control and data packets) beyond one radio hop is impossible. It is also economically infeasible to deploy a fully un-collaborative ad hoc mobile network where users' services do not exist.

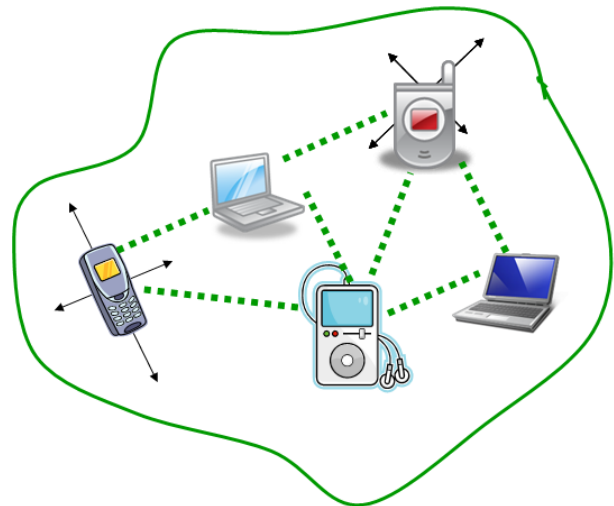


Figure 1. Ad Hoc Mobile Networks rely on neighboring nodes to forward packets towards their destinations.

In between these two extremes are ad hoc networks that contain both collaborative and un-collaborative nodes. Un-collaborative nodes are sometimes termed as selfish nodes [4]. In [5], selfish nodes are defined as nodes that maximize their own gain without regard to the welfare of other nodes. The dependability and relationship between routing and misbehavior of nodes were discussed and an analytical selfish model (which is tied specifically to a routing protocol) was proposed in [5].

The authors [5] pointed out the possible existence of different types of misbehaving nodes. Lazy nodes are nodes that are predominantly inactive in forwarding packets for others. They are also sometimes known as inactive nodes. Constrained nodes are nodes with limited energy to sustain on-going and future communications. Finally, malicious nodes are those that will cause harmful effects on the network, crippling communications and proper network operations. These nodes

can misroute, corrupt or drop packets. In this paper, we will focus primarily on selfish nodes.

The term "selfishness", therefore, has to be carefully studied and characterized, especially in the context of ad hoc mobile devices and users. In our view, an ad hoc mobile node cannot be selfish by itself.



Figure 2. Nodes that are constantly unwilling to forward packets for others may be regarded as exhibiting "selfish" behaviour in an ad hoc mobile network.

In this paper, we advocate the need to examine selfishness **from both device's and user's perspective**. A device might be perceived as "selfish" because it has been manufactured and constructed to conserve power consumption readily. Such behavior can be viewed by other more power-capable nodes as being "selfish" even though it is unintentional. This scenario will happen quite frequently since there exists many different heterogeneous mobile devices (PALM, PDA, Blackberry, etc), each with different CPU MIPS, memory size, display resolution, keyboard, battery power life, and storage capabilities.

From a **mobile user's perspective**, selfishness can be interpreted with a different dimension. The mobile user, coupled with his own preference, budget restrictions, thinking, and personality, may be unwilling to allow his or her ad hoc mobile device to forward other users' traffic all the time. Issues like device "on" duration, charges for transmissions over-the-air, power consumption, security threats, reduction in processing speed, disruption to on-going communications, etc., can be the governing factors that cause a mobile user to react in a selfish manner.

II. PREVIOUS WORK ON SELFISH EVALUATION

Many researchers had evaluated the impact of selfishness on communication performance in ad hoc networks. In [4], the authors proposed SPRITE - a simple, cheat proof, credit-based system for simulating cooperation among selfish nodes. They introduced the "message success rate" as a metric to evaluate performance of the SPRITE system. This rate refers to the percentage of messages that are successfully relayed from the sender to the destination node. The authors had also studied the impact of battery resource on message success rate.

In [2], the authors evaluated the impact of increasing misbehaving nodes and false accusations or detections on "network throughput" performance. They revealed that network performance drops with an increasing number of misbehaving nodes in the network.

In [11], the authors examine the impact of selfish nodes. They viewed selfish nodes as nodes that took advantage of other nodes for their own communication needs. The authors had identified four patterns of selfishness:

- Refusal to relay RREQ messages
- Refusal to send HELLO messages
- Intentionally delaying RREQ messages
- Relay data but not routing messages

They proposed detection methods based on: (a) listening for passive acknowledgment, and (b) timer to track on unreasonable delays. Their countermeasure method is to isolate the selfish node. Selfish evaluation was done using the throughput, false detection rate, and probability of connection degradation metrics. The authors also highlighted the serious consequences of false detection, i.e., mistaking unselfish nodes for selfish nodes.

In addition to the above mentioned metrics, we propose that selfishness should also be evaluated based on per route and the selfish metric should be a function of the number of uncooperative nodes over the total nodes in an ad hoc communication route. In our perspective, distinctions should therefore be made for selfishness of a node user and selfishness of a route.

Note that whenever a node is marked or blacklisted as "selfish", it is often viewed as behaving badly [1]. When will this node be removed from this blacklist? A misbehaving node can eventually change to be well behaving. How can one cope with such a transition? Hence, in a way similar to human's personality, we believe that selfishness can also change with time, location, conditions, etc. Summarizing, we view that selfishness should be regarded as a variable, not a constant. Selfishness is a time-varying condition-dependent state. Accurate modelling of selfishness is necessary to fully understand its impact on ad hoc network operation and communication performance.

III. DETERMINATION OF SELFISHNESS

How can one successfully conclude that a neighboring ad hoc mobile node is selfish? There are several literatures with various suggestions for the assessment and judgement of selfishness. Generally, they are based on an implicit method where passive traffic monitoring and reputation help in judging a node. Here, we also introduce the concept of explicit selfish determination. Both of these methods will be elaborated below.

A. Implicit & Explicit Methods

In [2], the number of packets a neighboring node has forwarded over a specified time window is used as a form of measure and subsequent grounds for classifying whether that a neighboring node is considered selfish or not. This is an implicit method.

Another approach is based on the use of "reputation" [6][7][8][9]. In [9], the authors detect non-cooperative nodes by using and spreading rumors. In their distributed reputation-based system, they gather observed (first-hand) and collected (second-hand) behavior of nodes and evaluate them on the nodes' future cooperation. Successful detection of misbehaving nodes will result in the isolation of that node.

In [8], the authors focus on reputation propagation, agreement, and convergence. It was urged that the speed of reputation propagation is an important factor for the convergence speed of reputation agreement. In a deviated approach, the authors in [7] establish reputation based on nodes' past history of relaying packets. In their approach, if a node fails to route a packet after promising to do so, it will get a low reputation and may subsequently be weeded out of the network. They confirmed that the routing of packets, with the consideration of nodes' reputation, could improve communication throughput and during routing, the next-hop node with the highest reputation is chosen instead of shortest path.

In [10], the authors proposed a hybrid scheme that combines the features of reputation- and price-based schemes. A debit counter is used to maintain balances of node services to other nodes. A passive monitor is present to preliminary identify selfish nodes and report the observations to the reputation manager. The reputation manager maintains and updates reputation ratings about other nodes based on its' own observations along with those gathered from other nodes. The authors had also performed comparisons of their scheme with that proposed in [2].

Other than the above, we introduce here explicit methods of determining selfishness. Explicit methods involve users/nodes specifying their degrees of collaboration in their beacons or other control packets. By periodically exchange such control packets, mobile nodes can learn about the selfish characteristics of their neighbors. In addition to determining ways to assess and judge on selfishness, we identify the need to distinguish the difference between **device** selfishness versus **user** selfishness. More importantly, we felt that an ineffective method of selfish determination can result in:

- Classifying nodes to be selfish too easily, rendering the network less useful since routable paths are now greatly reduced.
- Classifying nodes to be unselfish too easily, resulting in packets being dropped by neighboring nodes that are actually selfish.

These issues would be of great interests to ad hoc network service providers. Figure 3 shows a flowchart illustrating how a node can assess neighboring nodes as being selfish or unselfish. The check on "forwarding packets" is an implicit based method while "further verification" can rely on explicit based schemes.

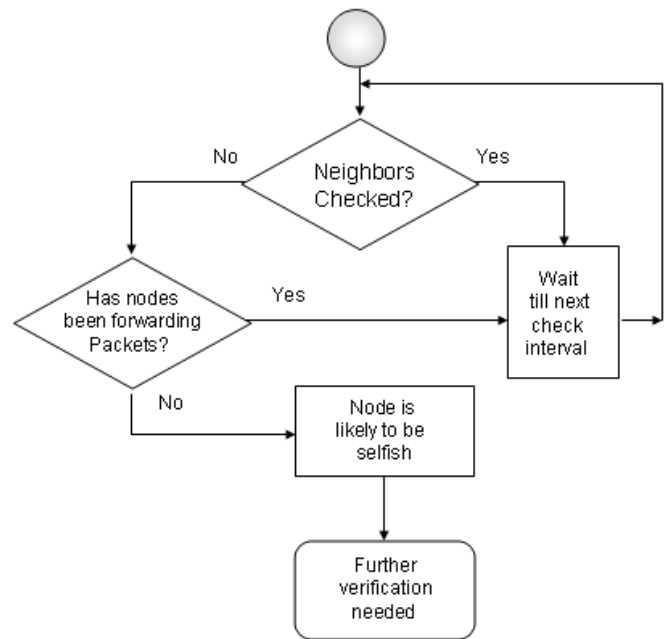


Figure 3: A Flowchart Illustrating Selfish Check.

B. The Need for Negotiation

Let us consider this scenario. An ad hoc mobile user migrating to a new environment might find himself enclosed in a community that is highly selfish in behavior. The user discovers that he is not able to send data to different destinations in the proximity. What other options will be available to him under such a circumstance? Clearly, if it is possible, he would like to migrate to another community in another location that houses more collaborative users but this is not generally possible or ideal. Hence, it is likely that negotiations on collaboration among nodes in an ad hoc mobile network are needed in order to negate or alleviate selfishness or hostility. Based on this rationale, we introduce a new distributed selfish check and negotiation protocol (SCNP), which is discussed below.

C. Proposed Selfish Check Negotiation Protocol (SCNP)

We proposed SCNP to provide a way to confirm a node's selfishness via an exchange handshake, where a node challenges the other node's intention and willingness to collaborate.

The requesting node sends a QUERY message to the next-hop neighbor. The neighbor then decides whether to reply positively to the request, decline, or ignore. This exchange allows the users to respond accordingly and may negotiate further (in terms of mutual agreements on subsequent exchanges, incentives, etc.) prior to making a decision on the level of collaboration. However, care should be exercised as a node and the user should not be constantly interrupted in order to respond to such requests. In our view, it is necessary for SCNP when passive monitoring (based on packet forwarding frequency) is inadequate to make good judgement on assessing

a node's selfishness. Table 1 below outlines the SCNP handshake protocol:

Begin	
Retry1:	Requester sent QUERY to next-hop node Wait for REPLY If receive REPLY and case (REPLY) of: Reject → Goto End1 Maybe → Goto Nego1 Accept → Goto Done1 Else TIMEOUT and Goto End2
Done1:	No Need for Negotiation; Goto End2
Nego1:	Requester sent "offer" to next-hop node Next-hop node evaluates "terms of offer" Next-hop node decides on ANSWER Case (ANSWER) of: Reject → Goto End2 if (Retry > 2) Else Goto Retry1: Accept → Send REPLY; Goto End2
End 1:	Negotiation not possible; End.
End 2:	Negotiation Successful; End of Negotiation;

Table 1: The SCNP Handshake Protocol

To avoid excessive negotiations that do not converge, SCNP should impose on the number of retry attempts on offers and counter-offers. Nonetheless, SCNP has the feature of allowing nodes to deliberate on "terms of offer" prior to deciding to be collaborative or not.

On another aspect, negotiation and agreement could be based on: (a) per flow, (b) per application, or (c) per user. To reduce the amount of control overhead, the selfish check and negotiation procedure should be invoked based on demand (i.e., when there is a need to relay traffic) and based on per application basis (i.e., an application can include a bundle of flows). A "broader" or so-called "blanket" agreement among users can, therefore, cover many aspects of communication collaboration. This, therefore, requires future investigations.

To evaluate the usefulness and importance of SCNP, we performed simulation using ns-2. Our SCNP is implemented over AODV routing protocol and the underlying MAC is 802.11 based. The physical space is 1500x1500m and the radio transmission range is 250m. We use UDP CBR traffic with a packet length of 1000 bytes. One hundred nodes are present in the ad hoc mobile network and the SCNP check interval is once every 60sec. Results obtained are averaged over 20 simulation runs. Parameters of interests to us are: (a) number of route failures, (b) control overhead incurred, and (c) number of packet losses with and without the presence of SCNP. We evaluate SCNP performance under varying number of selfish nodes present in the network.

As shown in Fig. 4, as we increase the percentage of selfish nodes, the number of route failures increases considerably when SCNP is absent. This is because selfish

nodes will not forward RREQ and REPLY packets, resulting in route request failures. Also, selfish nodes will not forward DATA packets, resulting in existing routes to fail. With the presence of negotiation, such failures are greatly reduced.

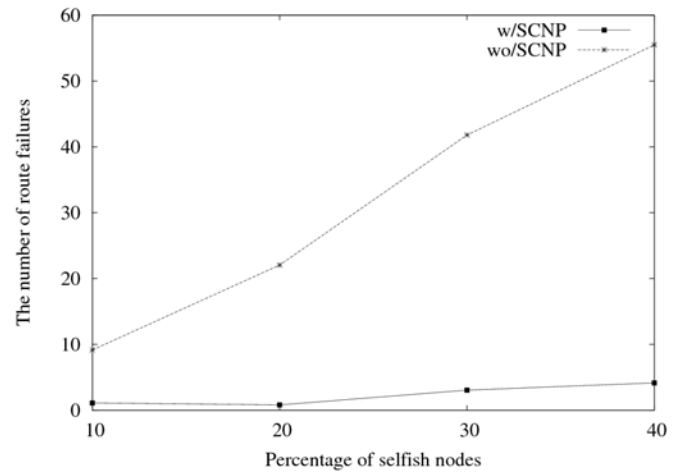


Figure 4. No. of Route Failures vs. Selfish Nodes.

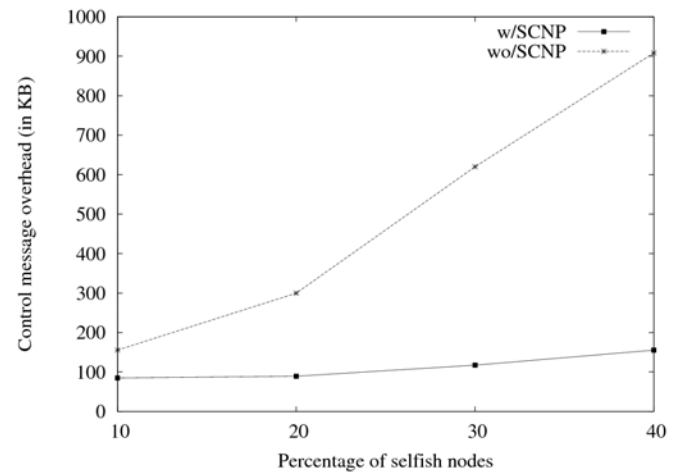


Figure 5. Control Overhead vs. Selfish Nodes.

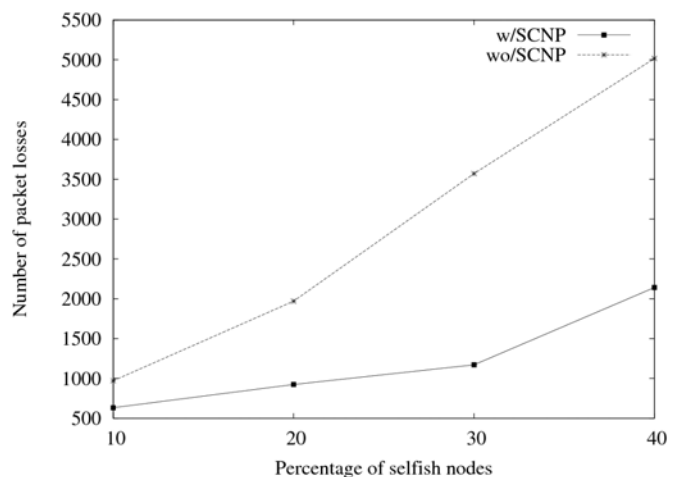


Figure 6: Packet Losses vs. Selfish Nodes.

The amount of control overhead incurred is shown in Fig. 5. It refers to the total number of control messages generated, including those generated from retransmitting route discovery requests and route failure messages. Clearly, as the percentage of selfish nodes increases, more route failures occur and hence significant amount of control messages are generated when SCNP is absent. Hence, it is clear that the initial control overhead incurred for SCNP negotiation is quickly justified by the reduction in overall control message if route failures were to occur when no negotiation feature is present.

Finally, we evaluate the amount of data packet losses as a result of increasing the percentage of selfish nodes. As shown in Fig 6, without SCNP, significant amount of packet losses occurred when selfish nodes exceed 20% of selfish nodes. Extensive route failures can substantially result in large packet losses. In the next section, we shall discuss other issues related to selfishness.

IV. FURTHER DISCUSSION

A. Selfish Degree & Programmability

An issue that requires further attention and study is the "**degree**" of selfishness. With the presence of various influencing factors and personalities, ad hoc mobile nodes will behave with different levels of selfishness. However, this is only true if users have access to and control of the collaborative functions within the nodes. This brings us to consider another issue: Should ad hoc mobile nodes be manufactured with a certain degree of collaborative programmability? This issue has not yet been addressed by researchers so far.

While ad hoc routing functions and software can be easily understood by network engineers and researchers, a typical user may not be able to change any existing ad hoc routing protocol functions within a mobile ad hoc device. Hence, should such dynamic collaborative features be made available, some kinds of application interfaces must exist so that normal users can adjust and set a node's collaborative behavior.

In our view, a simple solution would be a single control bar with a range of 0% - 100% of collaborative openness. Another form of collaborative interface could be one that is multidimensional, where one specifies the degree of collaboration depending on: (a) existing battery power levels, (b) returns, rewards/incentives, (c) relationships (personal, trust, security, reputation, etc.) with the packet originator, (d) pricing, and others. Again, this requires future work.

In summary, we advocate that a node's degree of selfishness is dependent on the node's degree of programmability on its collaborative function by the mobile user. If a user has no access to the internal collaborative function, the collaborative behavior of a node will be decided by what that was previously programmed into the nodes by the manufacturers.

B. Selfishness & Security

While security is not the main focus of this paper, here we briefly discussed its relationship with selfishness and also present our views. In [1], the authors believed that selfish nodes are prone to making sophisticated attacks. They presented a context-aware inference method for a source node to "blame" and "rate" an accused node conclusively. Nonetheless, one could also view selfishness and security from another perspective.

In our view, with the increasing number of threats from virus, worms, and malicious code, ad hoc mobile nodes may be unwilling to forward packets on behalf of other nodes. Such behavior should not necessarily be viewed as selfish or antisocial but rather as being "careful", "protective" or "conservative".

Denial of Service (DoS) scenarios can happen when nodes are overly protective or selfish too. When routable paths are absent, data communications, service discovery and accesses are hindered. Hence, a service provider (ad hoc mobile server) cannot be overly selfish else clients would not be able to connect to it to access services available in the ad hoc network. This, therefore, can result in poor service availability [5] in the ad hoc mobile network. Consequently, we view that a highly secured node can be selfish (if it chooses to) or unselfish. Unselfishness does not necessarily imply vulnerability to security attacks. Selfishness, on the other hand, does not necessarily imply proneness to make security attacks. If a node is well equipped with security measures, it can still behave unselfishly.

Malicious nodes, however, are different since they are nodes that behave uncooperatively and notoriously in an intentional harmful manner (in a way similar to Internet hackers trying to disrupt the operation of the Internet and deny users accesses to the network and services). Therefore, this should be treated as a security issue where malicious nodes will be confined and threats resolved. Hence, future work is necessary to show clearly the relationship, reliance and dependency of selfishness and network security.

C. Punishment & Rewards

Recently, some researchers [2] suggested "punishing" or "penalizing" selfish nodes in an ad hoc mobile network. In [2], punishment takes the form of "isolation". A node monitors the relay transmission of neighboring mobile nodes. If these neighboring mobile nodes do not forward others' traffic, it is marked as uncooperative and this uncooperative reputation is further propagated throughout the whole network. In response, nodes in the network will isolate these uncooperative nodes, essentially disabling the multi-hop communications capabilities of these mobile nodes.

In our view, punishments may not necessarily be the best resolution. Punishing and penalizing nodes is one solution with a retributive characteristic that may result in other less desirable effects. For example, nodes that are punished may seek to create havoc or "revenge", which result in destructive behavior, crippling the network further.

Another issue is the punishment process. Since an ad hoc

network could potentially be fully distributed, there is no centralized authority and hence it leaves one to wonder how such "punishments" could be carried out? A thorough punishment to an accused selfish node would require all surrounding nodes to agree not to forward packets for/to this node. This would also require collaborative punishment agreement to be present and the need for distributed coordination performed wirelessly. If these accusation and punishment processes incur too much control overhead in the network, nodes will have less capacity to perform useful data transfer. Hence, we view that "punishment" is not necessarily the best choice since ad hoc networks do require some forms of collaboration in order for it to operate usefully. In fact, the authors from [11] had identified the undesirable consequences of false detections and unjustified punishments.

Instead of punishment or isolation, another approach proposed by other researchers is to introduce rewards or incentives in order to encourage collaborative behavior. This approach is more viable since users could reconsider positively and behave less selfishly if they are attracted by these incentives. Prior work [3] had suggested the allocation of electronic money and credits in amount that is proportion to the volume of data forwarded on behalf of others. Other forms of incentives that we would like to introduce include the exchange of packet forwarding credits, accumulation of points for cash rewards, Internet access time, or gifts from sponsors (similar to the redemption of reward points in credit cards).

Further, we propose the concept of "shared interests" and the formation of "virtual communities" so as to encourage collaboration. Users interested in specific topics and themes will allow others with similar interests to share information and forward packets. This is analogous to subscribing to newsgroups or joining a multicast group. Communities with common interests and common missions or goals will likely collaborate in order to achieve common purpose and share useful information. Hence, users in an ad hoc network can also be made collaborative through the formation of such communities. This concept can also help in filtering out unwanted information and unwanted forwarding of other information packets.

Summarizing, if we consider economic, deployment, and business issues, it would make more sense to strive for an incentive- and reward-based system that would encourage ad hoc mobile network users to behave in an unselfish and collaborative manner so that effectively, nodes can communicate with one another and users can enjoy the benefits of mobile services anytime and anywhere.

V. CONCLUSIONS

In this paper, the controversy behind selfish nodes in mobile ad hoc networks is discussed. In particular, it questions the viability, validity, and interpretations of selfish nodes and their behavioral impact on applications and communication performance. The relationship between selfishness with migration, community, security and denial of service is further elaborated. We proposed a selfish check

negotiation protocol to allow nodes to negotiate for collaboration whenever possible. Simulation results reveal that this approach can greatly reduce route failures, packet losses, and control overhead. We have also advocated that incentive- and reward-based approaches will outweigh punishment-based approaches when it comes to encouraging nodes to be less selfish and more cooperative. Punishment approaches will also incur overhead and false detections of selfish nodes can result in unexpected loss of communication services for innocent nodes. Also, punishment can also result in revenge by punished nodes, which creates further destructive behavior. Finally, we propose the concept of shared interests and virtual communities to further encourage cooperation among nodes.

REFERENCES

- [1] K. Paul and D. Westhoff, Context Aware Detection of Selfish Nodes in DSR based Ad-Hoc Networks. In *Proceedings of IEEE Globecom*, (2002)
- [2] S. Marti, T. Giuli, K. Lai, and M. Baker, Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *Proceedings of ACM Mobicom*, (2000)
- [3] S. Buttyan, L. Hubaux and Jakobson, A Charging and Rewarding Scheme for Packet Forwarding in Multihop Cellular Networks. In *Proceedings of ACM Mobihoc*, (2003)
- [4] S. Zhong, J. Chen, and Y. Yang, Sprite: A Simple, Cheat-Proof, Credit-based System for Mobile Ad Hoc Networks. In *Proceedings of IEEE INFOCOM*, (2003)
- [5] M. Hollick, J. Schmitt, C. Seipl, and R. Steinmetz, On the Effect of Node Misbehavior in Ad Hoc Networks. In *Proceedings of IEEE ICC*, (2004)
- [6] Y. Liu and Y R Yang, Reputation Propagation and Agreement in Mobile Ad Hoc Networks, In *Proceedings of IEEE WCNC*, (2003)
- [7] P. Dewan, P. Dasgupta, and A. Bhattacharya, On Using Reputations in Ad Hoc Networks to Counter Malicious Nodes. In *Proceedings of QoS & Dynamic Systems Workshop*, (2004)
- [8] Y. Liu and Y. Yang, Reputation Propagation and Agreement in Mobile Ad Hoc Networks. In *Proceedings of IEEE WCNC*, (2003)
- [9] S. Buchegger and J. Boudec, The Effect of Rumor Spreading in Reputation Systems for Mobile Ad Hoc Networks. In *Proceedings of WiOpt Workshop*, (2003)
- [10] S. Soltanali, S. Pirahesh, S. Niksefat, and M. Sabaei, An Efficient Scheme to Motivate Cooperation in Mobile Ad Hoc Networks. In *Proceedings of International Conference on Networking and Services*, (2007)
- [11] S. Yokoyama, Y. Nakane, O. Takahashi, and E. Miyamoto, Evaluation of the Impact of Selfish Nodes in Ad Hoc Networks and Detection and Countermeasures Methods. In *Proceedings of International Conference on Mobile Data Management*, (2006)