SORI: A Secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks

Qi He†   Dapeng Wu‡   Pradeep Khosla†

†Carnegie Mellon University, Dept. of Electrical & Computer Engineering, Pittsburgh, PA 15213

‡University of Florida, Dept. of Electrical & Computer Engineering, Gainesville, FL 32611

Tel. (352) 392-4954, Email: wu@ece.ufl.edu. URL: http://www.wu.ece.ufl.edu.

Abstract— In an ad-hoc network, intermediate nodes on a communication path are expected to forward packets of other nodes so that the mobile nodes can communicate beyond their wireless transmission range. However, because wireless mobile nodes are usually constrained by limited power and computation resources, a selfish node may be unwilling to spend its resources in forwarding packets which are not of its direct interest, even though it expects other nodes to forward its packets to the destination. It has been shown that the presence of such selfish nodes degrades the overall performance of a non-cooperative ad hoc network.

To address this problem, we propose a Secure and Objective Reputation-based Incentive (SORI) scheme to encourage packet forwarding and discipline selfish behavior. Different from existing schemes, under our approach, the reputation of a node is quantified by objective measures, and the propagation of reputation is efficiently secured by a one-way-hash-chain-based authentication scheme. Armed with the reputation-based mechanism, we design a punishment scheme to penalize selfish nodes. The experimental results show that the proposed scheme can successfully identify selfish nodes and punish them accordingly.

Keywords: security techniques and systems, non-cooperative ad hoc networks, packet forwarding, selfish behavior, incentive, reputation

I. INTRODUCTION

The convergence of wireless communication (e.g., IEEE 802.11 and Bluetooth) and mobile computing devices (e.g., laptop, personal digital assistant (PDA), and wearable computer) offers great promise of providing us with unprecedented connectivity and mobility which enable us to enjoy untethered computing, at any place and at any time. One of the attractive paradigms under such a convergence is ad-hoc networks which can be easily and dynamically formed by a group of wireless mobile nodes without assistance from any fixed communication infrastructure such as base stations or access points. In an ad-hoc network, the transmission range of a mobile node is limited due to the power constraint, and there is no fixed communication infrastructure to facilitate packet forwarding; hence, the communication between two nodes beyond the transmission range relies on intermediate nodes to forward the packets. However, because mobile nodes are typically constrained by power and computing resources, a selfish node[1] may not be willing to use its computing and energy

---

[1]A node that does not fulfill the responsibility of forwarding packets is called "selfish node" or "misbehaving node". "Selfish node" and "misbehaving node" are two interchangeable terms in the discussion throughout this paper.

resources to forward packets that are not directly beneficial to it, even though it expects others to forward packets on its behalf. It has been shown [7] that the presence of such selfish nodes significantly degrades the overall performance of a non-cooperative ad-hoc network. Here, by "non-cooperative", we mean that a node is not willing to forward packets of other nodes unless it can benefit from the packet forwarding.

To address the above problem, we propose a Secure and Objective Reputation-based Incentive (SORI) scheme to encourage packet forwarding and discipline selfish nodes. The unique features of our SORI scheme are that 1) the reputation of a node is quantified by *objective* measures, 2) the propagation of reputation is *computationally-efficiently secured* by a one-way-hash-chain-based authentication scheme, and 3) the reputation of a node is only propagated to its neighbors but not entire network since the reputation of a node is only used by its neighbors in our scheme, which *reduces communication overhead*. Equipped with the reputation-based mechanism, we design a punishment scheme to penalize selfish nodes. The experimental results show that the proposed scheme can successfully identify selfish nodes and punish them accordingly.

The incentive schemes for packet forwarding in the literature basically fall into two categories, namely, *reputation-based* schemes and *pricing-based* schemes.

Reputation-based schemes utilize reputation in routing [7] and/or enforcing punishment [1]. However, the existing reputation-based schemes suffer from lack of effective mechanisms to measure and propagate reputation. Without quantitative and objective ways to measure reputation, and secure ways to propagate reputation, a reputation-based incentive scheme would not serve the purpose of stimulating packet forwarding, since reputation can be easily manipulated by selfish nodes in this case. Hence, this paper proposes a quantity to objectively measure reputation of a node, and a secure mechanism to propagate reputation, with the aim of resolving the drawbacks of the existing reputation-based incentive schemes.

Pricing-based schemes [2], [4], [12] treat packet forwarding as a service that can be priced, and introduce some form of virtual currency to regulate the packet-forwarding relationships among different nodes. However, these schemes require either tamper-resistant hardware [2] or virtual banks (trust authorities) that all parties can trust [4], [12]. In the case where tamper-resistant hardware is used, if a node (say, node A) sends much less traffic than another node (say, node B),

node A may drop most of the packets of node B without losing anything, since node A need not earn more money than necessary to support the small volume of its own traffic. In the case where a trust authority or virtual bank is required, it needs assistance from a fixed communication infrastructure to implement the incentive schemes, which is not applicable for a pure ad hoc network. In contrast to these, our SORI scheme is based on reputation and hence does not require tamper-resistant hardware or trust authorities; in addition, under our scheme, each node are motivated to maintain a good reputation in order to keep a desirable quality of its network connectivity.

The remainder of the paper is organized as follows. In Section II, we specify assumptions made in this paper. Under these assumptions, we first develop our basic scheme in Section III and then improve the basic scheme with a security enhancement described in Section IV. In Section V, we show simulation results to demonstrate the effectiveness of our scheme. Section VI concludes the paper.

II. ASSUMPTIONS

In this paper, we make the following assumptions.

1) The nodes in an ad hoc network under our consideration are *non-cooperative* in packet forwarding, that is, a node is not willing to forward packets of other nodes unless it can benefit from the packet forwarding. If nodes are cooperative, e.g., in military ad hoc networks, there is no need to use incentive mechanisms.
2) There is *no conspiracy* among nodes.
3) *Broadcast transmission*: A packet can be received by all the neighbors of the transmitting node (within its transmission range) because of the broadcast nature of the wireless medium.
4) *Desire to communicate*: All the participating nodes have the desire to communicate with some others.
5) *Invariant identity*: No node changes its identity during its life time.
6) *Selfish but not malicious*: A node may be selfish in terms of conservation of power and computing resources, but not malicious, which means that it will not try something that could be more expensive in consuming energy and computing resources than cooperating in packet forwarding.
7) *Promiscuous Mode*: Each node operates in a promiscuous mode, i.e., each node listens to every packet transmitted by its neighbors even if the packet is not intended for the node; and each node is able to determine who transmits the packet.

III. BASIC SCHEME

This section presents the basic scheme of our incentive mechanism, under the assumptions listed in Section II. The basic scheme consists of three components, namely, neighbor monitoring, reputation propagation, and punishment, which are described as follows.

A. Neighbor Monitoring

In our scheme, neighbor monitoring is used to collect information about the packet-forwarding behavior of the neighbors. Due to the promiscuous mode that we assume, a node is capable of overhearing the transmissions of its neighbors. With this capability, a mobile node N can maintain a *neighbor node list* (denoted by NNL_N) which contains all of its neighbor nodes that node N learns of by overhearing. In addition, node N keeps track of two numbers, for each of its neighbor (denoted by X), as below.

- $RF_N(X)$ (Request-for-Forwarding): the total number of packets that node N has transmitted to X for forwarding.
- $HF_N(X)$ (Has-Forwarded): the total number of packets that have been forwarded by X and noticed by N.

The two numbers are updated by the following rules. When node N sends a packet to node X for forwarding, the counter $RF_N(X)$ is increased by one. Then N listens to the wireless channel and check whether node X forwards the packet as expected. If N detects that X has forwarded the packet before a preset time-out expires, the counter $HF_N(X)$ is increased by one.

Given $RF_N(X)$ and $HF_N(X)$, node N can create a record called *local evaluation record* (denoted by $LER_N(X)$), for the neighbor node X. The record $LER_N(X)$ consists of two entries, that is, $G_N(X)$ and $C_N(X)$, where $G_N(X) = \frac{RF_N(X)}{HF_N(X)}$ and $C_N(X)$ is a metric called *confidence*, used to describe how confident node N is for its judgement on the reputation of node X. In our scheme, we set $C_N(X) = RF_N(X)$; that is, the more packets transmitted to X for forwarding, the better estimation about how well the neighbor X does forwarding.

B. Reputation Propagation

With the fore-mentioned neighbor monitoring, a node could build a record of the reputation of its neighboring nodes. However, our initial experimental result shows that actions (dropping selfish nodes' packets) solely based on one's own observation of its neighbors cannot effectively punish selfish nodes. To address this problem, reputation propagation is employed to have neighbors share the reputation information of other nodes, so that a selfish node will be punished by all of its neighbors (who share the reputation information about its misbehavior) instead of just the ones who get hurt by the selfish node. The reputation propagation works as follows:

1) Each node N periodically updates its $LER_N(X)$ for each neighbor node X based on the changes of $RF_N(X)$ and $HF_N(X)$, and it broadcasts the updated record to its neighborhood if $G_N(X)$ has been significantly changed.
2) Node N uses its $LER_N(X)$ and $LER_i(X)$ (where i is in the NNL_N) to calculate its *overall evaluation record* of X (denoted by $OER_N(X)$) as below:

$$OER_N(X) = \frac{\sum_{i \in NNL_N \cup \{N\}, i \neq X} \lambda_N(i) \cdot C_i(X) \cdot G_i(X)}{\sum_{k \in NNL_N \cup \{N\}, k \neq X} \lambda_N(k) \cdot C_k(X)} \quad (1)$$

where $\lambda_N(i)$ is the *credibility* that node i has earned from the perspective of node N. In current scheme, we

choose $\lambda_N(i) = G_N(i)$, especially, $\lambda_N(N) = 1$, and $\lambda_N(i) = 0$ if $RF_N(i) = 0$, which means a node would not have any credit from N if it has not been requested by N to forward any packet.

C. Punishment

With the reputation measure $OER_N(X)$ obtained, node N can punish its neighbor X by probabilistic dropping as follows. If the $OER_N(X)$ is lower than a preset threshold, node N takes punishment action by probabilistically dropping the packets originated from X. The probability of dropping is

$$p = \begin{cases} q - \delta & \text{if } q > \delta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $q = 1 - OER_N(X)$ and $0 < \delta < 1$ is the margin introduced for the following consideration: a dropping action could be occasionally triggered by some phenomena such as collision, rather than selfishness. Without the margin, two nodes may keep increasing the dropping probability and consequently fall into a retaliation situation. The margin is designed to help well-behaving nodes avoid this situation by treating each other a little bit more generously.

To summarize, this section presents our basic incentive scheme, which consists of neighbor monitoring, reputation propagation, and punishment. Neighbor monitoring is to collect information about misbehavior of neighbors and objectively quantify reputation of neighbors. Reputation propagation is aimed at sharing information among neighboring nodes to make the reputation measure more accurate (from statistical perspective, i.e., the weighted averaging in (1) helps removing dependence of the reputation measure on location and node identity). Punishment is to encourage packet forwarding and discipline selfish nodes.

The basic scheme is vulnerable to some tricks played by selfish nodes and we improve it by a security enhancement presented in section IV.

Remark 1: Objective quantification of reputation

- The reputation of a node is objectively measured based on the packet forwarding ratio of the node, and it directly affects the service the node can obtain from other nodes.
- The reputation of a node is weighted by the confidence, which is objectively determined by the number of packets that the node is requested to forward, and the total number of packets forwarded and observed. Such a weighting is rational from a statistical perspective, since the more samples that are used, the more reliable the estimation of reputation will be.
- The reputation of a node is also weighted by the credibility of the nodes which contribute to the calculation of the reputation. This makes it difficult for a selfish node to play with multiple identities, trying to use one identity to propagate fake information in order to improve its reputation under another identity. Since as a selfish node who intrinsically has very little credibility under any of its identities (otherwise it wouldn't be a selfish node), weighting the reputation by credibility effectively

limits the contribution that one identity can make to the calculation of the reputation under another identity. That is, if an identity (denoted by I_A) is used by a selfish node, this selfish node cannot effectively get a good reputation for the identity I_A by using another identity to propagate fake information. An extreme example is that a selfish node which never helps others on packet forwarding will earn a zero credibility from any other node and therefore cannot make any contribution to the calculation of the reputation under any other identity.

- A limitation of the reputation calculation is that the objectivity of the calculated reputation depends on the probability of transmission collision. This is because the measure $HF_N(X)$ may not reflect the total number of packets that are transmitted from node N to node X and have been correctly received by node X, due to packet collision in the wireless medium. However, our simulation results in Section V show that our incentive scheme can identify selfish nodes and punish them under a light or medium network load. For the case when the network is heavily loaded, congestion control is required. Obviously, no incentive scheme will work well under a heavy network load, without the aid of congestion control.

IV. SECURITY ENHANCEMENT

This section presents a security enhancement to fix the vulnerabilities of the basic scheme, which could be potentially exploited by selfish nodes.

A selfish node may play the following tricks in order to benefit itself without being detected by the basic scheme:

- Impersonate a node that is near the selfish node and has a good reputation, in order to send its own packets (using other's identity).
- Impersonate a node with a good reputation, in order to broadcast fake observation information in order to boost its reputation calculated by other nodes.

To address the above problems, without assuming the existence of any authentication infrastructure such as a public key infrastructure (PKI), we construct an authentication mechanism based on a one-way-hash chain [6] as below. Node N gets its identity, denoted by ID_N, by choosing a random number r_N and recursively applying a pseudo-random function h on r_N by k times,[2] that is, $ID_N = H_k(r_N)$ which is recursively obtained by

$$H_i(r_N) = \begin{cases} h(H_{i-1}(r_N)) & \text{if } i \in \{1, 2, \cdots, k\} \\ h(r_N) & \text{if } i = 0 \end{cases} \quad (3)$$

When N is joining an ad-hoc network, it broadcasts its identity ID_N and all its neighbors receive this identity and put this identity into their $NNLs$. The neighbors will use this identity

---

[2]The k is the length of the one-way-hash chain and this length limits the maximum numbers of the messages to generate before a new one-way-hash chain must be created. How to handle unlimited message authentication by switching one-way-hash chains was discussed by Perrig *et al.* [10].

to authenticate messages originated or forwarded by this node (identified by ID_N) thereafter.

Next, we describe our procedure of message authentication. Our procedure of authentication is the same as that suggested by Perrig *et al.* [10] for broadcast authentication. Node N partitions the time into equal intervals and assigns the i-th interval with a key (denoted by K_i) which is $K_i = H_{k-i}(r_N)$ in the one-way-hash chain. The messages sent in an interval will be accompanied by a message authentication code (MAC) which is computed with the corresponding key K and the message M as the input, denoted by $MAC(K, M)$. For instance, the content of packet P_i sent in the i-th interval is
$$\{M_i || MAC(K_i^{'}, M_i) || K_{i-d}\}$$
where M_i is the message to be sent in the i-th interval; the key $K_i^{'}$ is obtained from $K_i^{'} = f(K_i)$ where f is the second pseudo-random function[3] and $K_i = H_{k-i}(r_N)$; d is the key disclosure delay (for example, in the $(i-d)$-th interval, the message is authenticated by key K_{i-d}, and key K_{i-d} will be disclosed in the i-th interval). Once receivers receive a packet, they check if the key used for the MAC is already disclosed. If the key has not been disclosed, they cache the message and will check its authenticity at the time when the K_i is disclosed; otherwise, they discard the packet because the key was disclosed before they received the packet and the MAC could be potentially forged. In addition, a packet with an invalid MAC will be discarded.

This enhancement makes it difficult for a selfish node which has a bad reputation to send out its packets or broadcast fake observation information to affect the others calculation of its reputation by impersonating a node with a good reputation. This is because the MAC is computationally difficult to forge without the key of that node. This design eliminates the need for a PKI or other forms of authentication infrastructures which are usually not practical in an ad-hoc network environment. Moreover, our design is efficient because the authentication is done with a one-way-hash function which is computationally much cheaper than the digital signature used in many other schemes.

V. Simulation Results

In this section, we implement our basic incentive scheme on a simulator and evaluate its performance under various settings. The purpose of this section is to demonstrate the effectiveness of our scheme in identifying selfish nodes and punish them accordingly. This section is organized as follows. Section V-A describes the simulation setting, while Section V-B illustrates the performance of our scheme.

A. Simulation Setting

Our incentive scheme is implemented on ns-2 [9]. We simulate a wireless ad-hoc network with 50 mobile nodes randomly deployed in an area of 670×670 square meters. We use the Distributed Coordination Function (DCF) of IEEE 802.11 as the medium access control layer protocol, and

---

[3]The second pseudo-random function is used to avoid using the same key multiple times in different cryptographic operations, hash chains and MACs.

dynamic source routing (DSR) [5] as the routing protocol. The radio transmission range for each node is 250 meters and the transmission data rate is 2 Mbits/s. The physical layer model is either the free space or the two-ray propagation model [11], depending on the separation distance between the transmit antenna and the receive antenna. The height of both the transmit antenna and the receive antenna is 1.5 meters.

The random waypoint mobility model [3] is used to generate the moving direction, the speed and the pause duration of each node. The node speed is uniformly distributed between 0 and 20 m/s, and the pause duration is exponentially distributed with an expectation of 600 seconds.

Among the 50 nodes, 5 nodes are randomly selected and assigned as selfish nodes. A selfish node in our simulation probabilistically drops packets from other nodes unless it is the destination of the packet.

For each simulation, we first set the total number of connection, denoted by N_{conn}. Then, we randomly generate N_{conn} source-destination pairs; the generated source-destination pairs may be duplicated, that is, the same source-destination pair may have multiple connections. Each connection lasts for 10 simulated seconds. Once a connection is terminated at the end of the 10th simulated second, a new source-destination pair is randomly generated and a connection is set up between the newly generated source-destination pair. Since the source-destination pairs are randomly generated (each node is equally likely to be selected to form a pair), the traffic is uniformly distributed among different nodes. In addition, the constant bit rate (CBR) traffic model in ns-2 is employed for all the connections. Each simulation is executed for 1000 simulated seconds. We set the threshold δ in Eq. (2) to 0.1, for all the simulations.

Next, we evaluate the performance of our scheme.

B. Performance Evaluation

We organize this section as below. In Sections V-B.1 through V-B.3, we investigate how our incentive scheme performs under various number of connections, different data-rate of CBR traffic, and various dropping probability of selfish nodes, respectively. Section V-B.4 shows the overhead incurred by our incentive scheme.

1) Performance under Various Number of Connections: This experiment is to show the performance of our incentive scheme under various number of connections N_{conn}. In this experiment, a selfish node drops all the packets from other nodes unless it is the source or the destination of the packet; the data rate of all the CBR connections is fixed to 1 packet/sec.

Fig. 1 plots the average throughput of a well-behaving/selfish node vs. the total number of connections N_{conn}. Note that for each simulation run, the value of N_{conn} is fixed. In the figure, the average throughput of a well-behaving node is obtained by 1) summing up the number of packets correctly received by all well-behaving nodes, 2) dividing this sum by the total number of well-behaving nodes, and 3) dividing the result by the total simulation time,
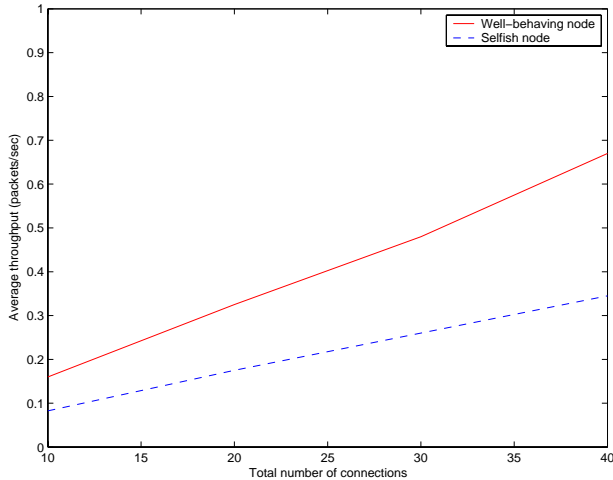
Fig. 1. Throughput under various number of connections N_{conn}

Fig. 2. Throughput under various data-rate

Fig. 3. Throughput

i.e., 1000 seconds. Similarly, the average throughput of a selfish node is obtained by 1) summing up the number of packets correctly received by all selfish nodes, 2) dividing this sum by the total number of selfish nodes, and 3) dividing the result by the total simulation time.

As depicted in Fig. 1, a well-behaving node achieves significantly higher average throughput than that of a selfish node; on average, a selfish node suffers about 50% throughput reduction, as compared to a well-behaving node. This shows that our scheme can identify a selfish node and punish it accordingly.

2) Performance under Various Data-rate: This experiment is to show the performance of our incentive scheme under various data-rate of CBR traffic. In this experiment, a selfish node drops all the packets from other nodes unless it is the source or the destination of the packet. For each simulation, the data rate of all the CBR connections is fixed; but for different simulation, the data rate of CBR connections changes from 1 to 10 packets/sec. For all the simulations, we set the total number of connections $N_{conn} = 10$.

Fig. 2 shows the average throughput of a well-behaving/selfish node vs. the data-rate of a CBR connection. As shown in Fig. 2, a well-behaving node achieves higher average throughput than that of a selfish node; however, the difference (in terms of percentage) between the throughput of a well-behaving node and that of a selfish node reduces with the increase of the data-rate. The reason for this reduction is the following: as the data-rate increases, the probability of transmission collision increases, which results in higher probability of mis-calculation of objective reputation (refer to Remark 1).

3) Performance under Various Dropping Probability of Selfish Nodes: This experiment is to show the performance of our incentive scheme under various dropping probability of selfish nodes. In this experiment, each selfish node probabilistically drops the packets from other nodes unless it is the source or the destination of the packet. The dropping
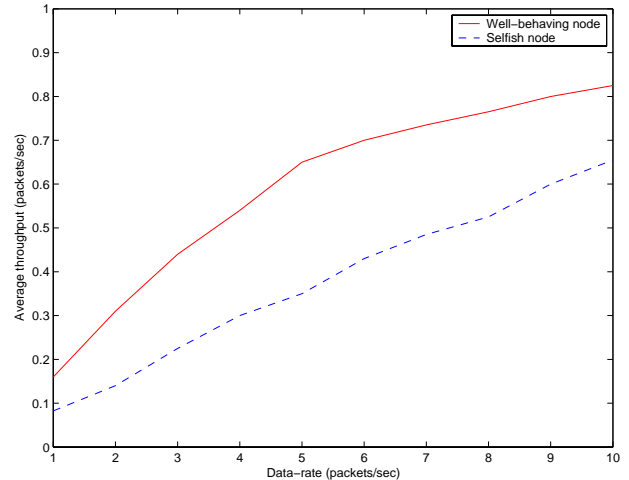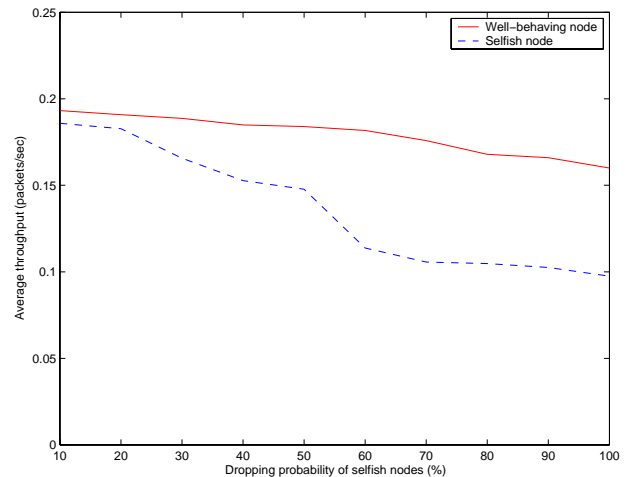
probability is the same for all selfish nodes and varies from 10% to 100% in different simulations. That is, for each simulation, the dropping probability of selfish nodes is fixed; but for different simulation, the dropping probability changes from 10% to 100%. For all the simulations, we set the total number of connections $N_{conn} = 10$ and fix the data rate of all the CBR connections at 1 packet/sec.

Fig. 3 plots the average throughput of a well-behaving/selfish node vs. the dropping probability of selfish nodes. It can be seen that as the dropping probability of selfish nodes increases, the gap between the throughput of a well-behaving node and that of a selfish node increases. Hence, our incentive scheme can not only distinguish the selfish nodes from the well-behaving nodes, but also impose a punishment proportional to the severity of the selfish behavior. This shows the effectiveness of our incentive scheme.

Fig. 3 also shows that the average throughput of a well-behaving node decreases with the increase of the dropping probability of selfish nodes. This is because the loss proba-

bility of a packet destined to a well-behaving node increases with the dropping probability of selfish nodes.

4) Overhead Incurred by Our Incentive Scheme: This experiment is to show the overhead incurred by our incentive scheme, as compared to the scheme that does not use our incentive mechanisms. In this experiment, a selfish node drops all the packets from other nodes unless it is the source or the destination of the packet; the data rate of all the CBR connections is fixed to 1 packet/sec. We do simulations for two schemes: one is our incentive scheme, and the other (which we call *benchmark scheme*) does not take any of the three actions, i.e., neighbor monitoring, propagating reputation, and punishing selfish nodes.

Fig. 4 plots the average throughput vs. the total number of connections N_{conn}. The dashed line in the figure shows the average throughput of a node (averaged over all nodes) under the benchmark scheme (without incentive mechanisms); here, since the benchmark scheme does not distinguish the selfish nodes from the well-behaving nodes, the average throughput of a selfish node should be the same as that of a well-behaving nodes, from the statistical perspective. The solid line in the figure shows the average throughput of a well-behaving node under our incentive scheme. It can be observed that the throughput of a well-behaving node under our incentive scheme is reduced, as compared to that under the benchmark scheme. This throughput reduction is what we call *overhead.* The reason for this throughput reduction is two-fold: first, reputation propagation consumes bandwidth and therefore reduces the net throughput; second, collision may cause mis-calculation of the reputation measure, leading to improper punishment on the well-behaving nodes.

Fig. 4 shows that the overhead incurred by our scheme is not more than 8%, which is small. Just because of this small overhead, we are able to propagate reputation, identify selfish nodes and punish them according to the severity of their misbehavior. It can also be seen that the overhead increases with the total number of connections N_{conn}. This is because the larger N_{conn}, the high probability of collision, which results in higher probability of reputation mis-calculation and hence larger overhead.

VI. CONCLUDING REMARKS

In this paper, we propose a Secure and Objective Reputation-based Incentive (SORI) scheme to encourage packet forwarding and discipline selfish behavior in a non-cooperative ad hoc network. The unique features of our SORI scheme are that 1) the reputation of a node is quantified by objective measures (through neighbor monitoring), 2) the propagation of reputation is secured by one-way-hash chain based authentication scheme, which is computationally efficient, and 3) the reputation of a node is only propagated to its neighbors, which greatly reduces communication overhead as compared to the schemes that maintain reputation globally. With the reputation measure obtained by the SORI scheme, we are able to design a punishment scheme to penalize selfish nodes. The experimental results show that the proposed
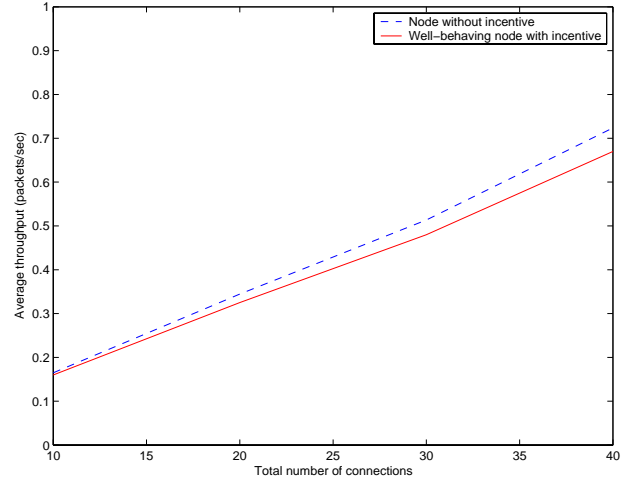

Fig. 4. Communication overhead of our scheme

scheme can successfully identify selfish nodes and punish them accordingly.

REFERENCES

[1] S. Buchegger and J. Le Boudec, "Performance analysis of the CONFI-DANT protocol: cooperation of nodes - fairness in distributed ad-hoc networks," *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, Switzerland, June 2002.

[2] L. Buttyan and J. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Boston, MA, USA, Aug. 2000.

[3] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communication & Mobile Computing,* Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483–502, 2002.

[4] M. Jakobsson, J. Hubaux, and L. Buttyan, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks," *Proceedings of Financial Crypto 2003*, Gosier, Guadeloupe, Jan. 2003.

[5] D. Johnson, D. Maltz, and J. Broch, "The dynamic source routing protocols for mobile ad hoc networks," *Internet Draft, IETF Mobile Ad Hoc Network Working Group*, Oct. 1999.

[6] L. Lamport, "Password authentication with secure communication," *Communications of the ACM,* vol. 24, no. 11, pp. 770–772, Nov. 1981.

[7] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Proceedings of Mobicom 2000*, Boston, MA, USA, Aug. 2000.

[8] S. Micali and R. Rivest, "Micropayments revisited," *Proceedings of the Cryptographer's Track at the RSA Conference 2002*, Bart Preneel (ed.), Springer Verlag CT-RSA 2002, LNCS 2271, pp. 149–163.

[9] NS developing group, "The network simulator – ns-2," available at http://www.isi.edu/nsnam/ns.

[10] A. Perrig, R. Canetti, D. Song, and J. Tygar, "Efficient and secure source authentication for multicast," *Proceedings of the Symposium on Network and Distributed System Security (NDSS 2001)*, Internet Society, 2001.

[11] T. S. Rappaport, *Wireless Communications: Principles & Practice,* Prentice Hall, 1996.

[12] S. Zhong, J. Chen, and Y. Yang, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," *IEEE INFOCOM 2003*, San Francisco, CA, USA, April 2003.