SIP: A Secure Incentive Protocol against Selfishness in Mobile Ad Hoc Networks

Yanchao Zhang
Department of ECE
University of Florida
Gainesville, FL 32611, USA
Email: yczhang@ufl.edu

Wenjing Lou
Department of ECE
Worcester Polytechnic Institute
Worcester, MA 01609, USA
Email: wjlou@ece.wpi.edu

Yuguang Fang
Department of ECE
University of Florida
Gainesville, FL 32611, USA
Email: fang@ece.ufl.edu

***Abstract*—Security in mobile ad hoc networks (MANETs) has received intensive attention recently, whereas the issue of selfish nodes, which may refuse to forward packets for others to save their own resources, is not well addressed yet. This kind of non-cooperative action would cause a severe problem that is more likely in MANETs compared to their wired counterpart. To cope with this problem, we propose SIP: a Secure Incentive Protocol to stimulate cooperation among those possible selfish nodes. The most attractive feature of SIP is that it does not rely on any pre-deployed infrastructure and provides highly secure incentives for selfish nodes to be cooperative in packet forwarding with low overhead and implementation complexity.**

Keywords-Security techniques and systems; Ad-hoc networks; Selfishness.

I. INTRODUCTION

Security in mobile ad hoc networks (MANETs) has received intensive attention recently. Most of the proposals were targeted for *malicious* nodes whose objectives are to attack the proper operation of networks without caring about their own gains. However, there is another kind of non-cooperative nodes, *selfish* nodes who may participate in forwarding network-wide control packets such as routing packets to maintain their own knowledge about the networks, while refuse to forward data packets for other nodes for the sake of saving their own resources such as battery energy, CPU cycles or available bandwidth. This kind of *selected forwarding* action may result in a severe problem that is more likely to happen in MANETs compared to their wired counterpart due to the fact that nodes in MANETs always rely on others to reach non-neighboring nodes. As reported in [1], if 10%-40% of the nodes in the network are selfish, the average throughput could degrade by 16%-40%.

There are two main approaches aiming to address this issue, namely, *reactive* approaches and *preventive* approaches. The former is intended to enforce the cooperation by firstly detecting the selfish nodes, avoiding routing through them, and then punishing them by spreading their bad reputations and thus isolating them [1-4]. The major concern, however, is that it seems difficult, if not impossible, to prevent the propagation of incorrect reputations (either good or bad) in a secure and efficient way. As for the latter, most of the proposals are related to provide some kinds of incentives for the selfish nodes to participate in packet forwarding, either self-organizing [5-7] or depending on pre-deployed infrastructure [8-10]. Besides, Srinivasan *et al.* also proposed a non-incentive-based preventive approach in [11], which allows mobile nodes (MNs) to choose whether to relay traffic or not based on their own energy efficiency of cooperation using game theory.

In this paper, we propose SIP: a Secure Incentive Protocol to secure and stimulate the cooperation in totally self-organized MANETs. A secure credit-based remuneration protocol is carefully designed to ensure the correct charging and rewarding of the credits on each node for packet sending, receiving, and forwarding. Our mechanism is motivated mainly by Hubaux *et al*. [5-6,9]. In [5], Buttyan and Hubaux proposed that intermediate nodes (INs) should be remunerated for the service they provided and presented two payment models based on a virtual currency called *nuggets*. Later, they improved their work in [6] with a mechanism based on a *nugget* counter which is implemented in a tamper-proof security module inside each node and handles the payment. Each node maintains a so-called pending nugget counter for each neighbor, which records the nuggets that should be awarded to one neighbor for the packets forwarded from that neighbor. And MNs exchange their nugget counter information by regularly running a nugget synchronization protocol. Salem *et al*. presented another session-based charging and rewarding scheme in [9] for a symmetric multi-hop cellular network where multiple hops exist in both the up-stream and down-stream. They relied on a centralized base station and a billing center to handle and secure the payment service for MNs.

Similar to [6], we assume that each MN has a tamper-proof security module to ensure the correct functioning of the designed protocol. We also borrowed from [9] the idea that each IN puts non-forged stamps on the forwarded packets as the proof of forwarding. However, our work differs significantly from theirs in many aspects. Firstly, the application scenario addressed in [9] is a multi-hop cellular network with a fixed infrastructure, while in this paper we consider totally infrastructureless ad hoc networks where MNs take care of charging and rewarding themselves. Compared with [6], SIP adopts a session-based approach instead of a packet-based one. And session endpoints are responsible for the whole payment procedure and thus INs are relieved from the burden of maintaining nugget counters for neighbors and

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and under grant N000140210554.

periodically synchronizing them. Therefore, the related processing and communication overhead is much smaller. In addition, SIP is designed to be tolerant of a broad range of attacks. Moreover, we adopt a novel key establishment procedure to reduce the overhead of establishing inside-session keys, which could also be utilized to realize other security-related services besides SIP.

II. PRELIMINARIES

Before presenting our scheme, we first define the system model and the node behavior model adopted in our design in this section.

A. System model

We assume that each MN is equipped with a tamper-proof smartcard such as SIM cards in GSM networks, which deals with security related functions. And our proposed SIP is implemented in the smartcard, so its correct functioning could be guaranteed. This assumption accord with the claim in [12] that the physical security or the tamper-proof property is indispensable to meet the requirements of high-level security modules. Furthermore, each smartcard has its entity ID which can also be used to identify the entity of the MN where a smartcard resides. To facilitate our presentation, we refer to the smartcard ID as the node ID and do not distinguish among the smartcard of one MN, the user of one MN, and the MN itself.

To guarantee the security of SIP, we further require that each smartcard contains a private number and a public number defined as follows. Firstly, an appropriate public prime p and a generator α of $Z_p^*(2 \le \alpha \le p-2)$ are negotiated and stored in each smartcard during manufacturing process, where Z_p^* is the *multiplicative group* of integers modulo p [13]. Secondly, a secret number $x(1 \le x \le p-2)$ is randomly chosen as the private number, and α^x mod p serves as the public number which is signed by its manufacturer's RSA private key (called public number certificate). For simplicity, in what follows we abbreviate α^x mod p to be α^x. Therefore, each smartcard contains a key tuple, namely, (x, α^x, certificate).

In addition to the key tuple, each smartcard holds the RSA public keys of its own manufacturer and several other common manufacturers. In this way, when two MNs become neighbors, they could exchange and verify the public number certificates of each other with the pre-stored public keys of the manufacturers, and then establish a shared mater key of α^{xy} mod p based on the Diffie-Hellman protocol [13].

SIP uses "credits" as the incentives to stimulate packet forwarding. For this purpose, each smartcard has a credit counter (CC) which is pre-charged with a certain amount of credits before shipped out. The charging and rewarding on a node is done by decreasing or increasing the CC in that node. And the CC will retain its value even when the MN is power-off. When the MN is power-on again, it could still reuse the credits in the CC even in another SIP-enabled ad hoc network.

B. Node model

We assume that MNs are *selfish* so that they are reluctant to serve others for free. Since SIP is credit-based, selfish nodes are also *greedy* in the sense that they will try to cheat for credits, either by paying less or gaining more. For example, if possible, they will try to bypass SIP to request free service without paying for it; or they will try not to forward a packet if they could gain from doing so; or if possible they will try to reward themselves for the work they did not do. However, they are also *rational*, which means that they only attempt to cheat if the expected benefit of doing so is greater than that of acting honestly. How to deal with the *malicious* nodes, whose only objectives are to interrupt the operation of networks without consideration of their benefits, is beyond the scope of this paper.

As an addition, we assume that a selfish node is smart enough so that he/she has full control over the communication interface, processing units, and so on. For instance, he/she could manipulate the input and output of the smartcard, while has no control over the smartcard itself and the protocols implemented in it. He has no knowledge about the keys stored in the smartcard and could not change CC in an unauthorized way either.

III. OPERATION OF SIP

In this section, we detail the operation of SIP and its security considerations.

A. Overview

We adopt a credit-based payment system which charges or rewards nodes for the service they receive or provide. Different from previous proposals in which the payment for packet forwarding by intermediate nodes is covered by either the source or destination, we argue that both the source and destination should pay simply because both of them benefit. The payment proportion between them is adjustable and can be negotiated during the session initialization phase. For brevity, in this paper we assume that both of them pay half of the total credits.

SIP is implemented in the smartcard of each MN. We require that, whenever one MN has a packet intended for the other non-neighboring node, it must first pass the packet to its smartcard where a special SIP header is added before sending it. Whenever an IN receives a packet destined for other nodes, it must also pass the packet to its smartcard for SIP processing before forwarding it.

SIP is session-based and mainly consists of three phases. During the first *Session initialization* phase, a session initiator (SI) negotiates session keys and other information with a session responder (SR) and INs between them. And each IN puts a non-forged stamp on each data packet forwarded and SI/SR collect those stamps for later rewarding use in the next *Data forwarding* phase. The final phase is R*ewarding* phase, in which each IN is awarded a certain number of credits based on the number of forwarded packets.

The format of an IP packet with SIP header is shown in Fig. 1. Since SIP deals with the IP packet forwarding at each node,

SIP header should be inserted between IP header and the transport protocol header if any. The *Protocol* field in the IP header should be set accordingly so that one MN knows that the packet belongs to a SIP packet and passes it to the smartcard for processing. The first four fields of SIP header are common to all SIP packets. The *SI ID* (ID_{SI}) and *Session IN* (SN_{SE}) fields are the same for all the packets in one session which uniquely identify a session. And each packet also contains a unique inside-session packet sequence number SN_{PK} whose value is initialized to zero and used in the increasing order at SI, and is initialized to the maximum possible value and used in the decreasing order at SR. Since the first three fields uniquely identify a packet, we will skip the first three fields and focus on the remaining fields for different types of packets in the remainder of this paper.

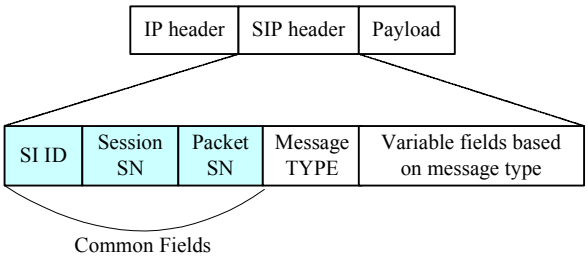

Figure 1. SIP header format

B. Protocol details

The proper operation of SIP depends on the correct functioning of underlying on-demand unicast routing protocols. However, SIP is routing-independent in the sense that it could coexist with any on-demand unicast routing protocol such as DSR and AODV. As mentioned before, we assume that *selfish* nodes participate in forwarding routing packets to maintain their own knowledge about the networks and one end-to-end path from SI to SR has been established.

We use ID_{SI}, ID_{SR} and ID_i to denote the smartcard IDs, and n_{SI}, n_{SR} and n_i to denote the private numbers of SI, SR and IN_i (the *i*th IN), respectively.

1) Session initialization: If one MN (SI) intends to initiate a session with the other non-neighboring MN (SR), it first sends SR a KAGREE-type control packet which contains its public number certificate and subsequent session traffic information such as the expected amount of total traffic, the session duration, the packet arrival rate, the rewarding frequency, and optionally the payment splitting method. Whenever receiving a routing error packet or no response for a KAGREE packet sent after some time, SI has to resend a KAGREE packet to resume the corrupted session.

Each IN, once receiving a KAGREE packet, has a chance to choose whether or not to forward the packets for this session based on its current resource availability and potential traffic information described in the received packet. It can refuse to participate without bearing any punishment if it does not want to. In this case, SI will have to rely on the underlying routing protocol to discover a new route or switch to an alternate cached route. If one IN decides to participate, it needs to verify the certificate of SI using the pre-stored public key of SI's manufacturer, keep an authentic copy of it, and then forward the KAGREE packet to the next hop. Those actions could be automated by setting the parameters of the smartcard accordingly.

The same process continues until the KAGREE packet reaches SR. If SR agrees to start the session, it forms a KACK-type packet by attaching its node ID and public number certificate, and sending it to SI. Upon receiving a KACK packet, each IN will append its ID and public number certificate to the KACK packet, and forwards it further. After authenticating all the certificates in the KACK packet, SI acquires the authentic public numbers of SR and all the INs. Meanwhile, each IN also acquires the authenticated public number of SI.

Through the above procedure, SI learns all the INs participating in the session. Therefore, the shared session keys between SI and IN_i can be computed as $K_i = h(ID_{SI} \mid ID_i \mid SN_{SE} \mid \alpha^{n_{SI} n_i})$ [1]. Similarly, the shared session key between SI and SR is computed as $K_{IR} = h(ID_{SI} \mid ID_{SR} \mid SN_{SE} \mid 0 \mid \alpha^{n_{SI} n_{SR}})$. In this way, SI could establish a shared session key with each of the participating nodes including SR. We can have two session keys between SI and SR, of which K_{IR} is used for the same purpose of data integrity check as K_i, while the other one $K_E = h(ID_{SI} \mid ID_{SR} \mid SN_{SE} \mid 1 \mid \alpha^{n_{SI} n_{SR}})$ is used for the end-to-end data encryption.

Immediately after the session initialization, SI will send a KSYNC-type packet to distribute a common key among all the nodes involved in the session. SI can select a key K_{COMM} randomly and encrypt it using the corresponding derived session keys with each IN and SR. We will talk about the use of K_{COMM} shortly.

2) Packet forwarding: The format of a Data-type SIP packet is as <TYPE=DATA, MAC, CMAC, Payload>. The payload part could be encrypted with the shared session key KE between SI and SR with some efficient symmetric cryptography algorithms such as RC4 [13].

The variable fields in a DATA-type SIP header include a packet authentication code (MAC) and a Common MAC (CMAC). If the packet is initiated at SI, the initial MAC of the packet is computed as $MAC_0 = h(SN_{SE} \mid SN_{PK})$. The subsequent IN_i replaces the MAC field with a new one $MAC_i = h(MAC_{i-1} \mid K_i)$ and forwards the packet to the next hop. When receiving the packet, SR will form MAC_{final} by hashing the received MAC value with K_{IR} and then saves it along with SN_{PK} and the length of the packet into a RECEIPT table maintained by SR in its smartcard. To minimize the communication overhead, SR accumulates a certain number of receipts and sends them back together to SI in a SIP control packet called RECEIPT, which contains one tuple of < SN_{PK}, MAC_{final}, *Packet Length*> for each covered packet.

[1] "|" denotes the concatenation of packets. *h(M|K)* denotes the keyed hash or packet authentication code (MAC) of packet M with key K. h() could be any popular hash function such as MD5 or SHA [13].

After sending one RECEIPT packet, SR deletes the related information from its RECEIPT table. Upon receiving one RECEIPT packet, SI could authenticate the receipts by recomputing the related MAC_{final} values and comparing them with those contained in the RECEIPT packet. Then it stores the authenticated packet sequence numbers along with packet length information into a REWARD table, which is maintained by SI in its smartcard and used to record the information for which INs have not been remunerated yet.

For a data packet initiated at SR, the same hop-by-hop keyed hash processing applies. However, when receiving the packet, SI directly stores SN_{PK} and the packet length information in the REWARD table after authenticating the MAC_{final} in the data packet. No receipt is needed for that case.

Here, the reason why we apply hop-by-hop keyed hash is to ensure that each IN did participate in the packet forwarding and the packet was successfully received by the intended destination. This could defend against attacks such as forging receipts or overcharging SI and SR. Without such precautions, some greedy nodes could collude and list non-participating node(s) in the rewarding list.

The other field CMAC in a DATA-type packet is used to defend against the "free riding" attack described in [9] where two dishonest nodes on the route may attempt to send data between them without paying for it. For instance, after the session initialization phase, an IN might start to send data without going through the smartcard by forging a SIP data header pretending to be a data packet of the session he/she is serving. Any downstream node colluding with him/her can simply extract the payload and drop the packet. Both of them would not pay for the service. In [9], the authors used a per-hop encryption/decryption scheme to prevent this kind of attack. Here, we use a different per-hop CMAC checking method to defend against this attack.

CMAC is initialized as $CMAC = h(payload|K_{COMM})$. Each IN, before forwarding the packet, will recompute the CMAC value and verify its correctness. Only when it is authentic, will the data packet be forwarded. Otherwise, the IN will simply drop the packet. Since all the keys are stored in the smartcard, the packet without passing through the smartcard will not be able to have the correct CMAC and any IN between two colluding nodes will drop the free riding packets accordingly. Therefore, the free riding attack is effectively defeated.

3) Charging and rewarding: As we mentioned earlier, our intention is to reward each IN for the traffic they relayed while charge the source and destination for the service they received. Our charging and rewarding scheme is carefully designed to ensure that the charging and rewarding is correctly done.

Suppose the charging rate of each IN for successfully traffic relay is a constant, μ/byte. In our scheme, whenever SI or SR sends a packet, say l bytes, the CC of the packet source is decreased by $N_{IN}l\mu$, where N_{IN} is the number of intermediate nodes between SI and SR and is obtained during the session initialization phase. Similarly, whenever receiving a packet, the CC of the intended destination (either SR or SI) is also decreased by the same credits as before. These actions are taken when a SIP packet is passed to the smartcard of the source or the destination for SIP processing. The amount of credits charged at this time is actually more than they should be since both of them only need to pay half of the credits. We overcharge them at this time in order to urge them to send REWARD packets later when the overcharged credits will be returned to them.

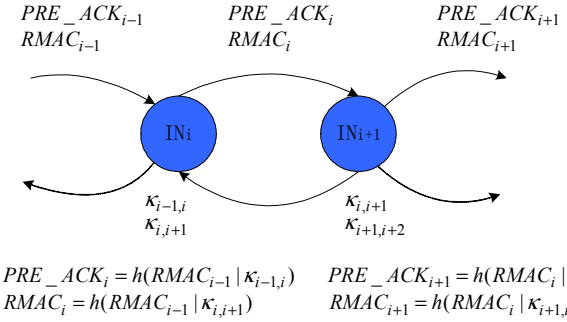

Figure 2. Passive acknowledgement

If the CC of SI or SR does not have enough credits, its smartcard will deny processing packets and the session is suspended. To resume the session, they either have to wait until earning enough credits by relaying traffic for other nodes or try to negotiate the payment proportion with each other through KAGREE packets.

It is up to SI to decide when and how frequently to reward INs as described in the KAGREE packet. SI can accumulate a certain amount of packets to be rewarded and send them together in a REWARD-type packet to SR. Generally, if the session is stable, the REWARD packets can be sent less frequently. While when the session is unstable, for example, when route errors occur frequently due to the node mobility, the REWARD packets need to be sent more frequently to remunerate INs in a timely manner before they move. The format of a REWARD packet is as *<TYPE=REWARD, Number of Bytes, Rewarding node list, PRE-ACK, RMAC>*. Here, the reason why we use Number of Bytes, which is the total length of packets to be rewarded, rather than the number of packets as the rewarding unit is to allow SI and SR to exchange packets with different sizes.

The number of bytes for which each IN is to be awarded could be in plaintext, but should be integrity-protected with the RMAC (reward packet authentication code) field to prevent potential modifications by dishonest INs. To urge the INs to forward the REWARD packets, we utilize a passive acknowledgement approach (see Fig. 2) to confirm the forwarding from the previous node. For this purpose, each REWARD packet also includes a PRE-ACK field which is used to provide acknowledgement to the previous hop.

Before SI sends the reward packet, the $RMAC_0$ is initialized as the keyed hash of all the fields in the SIP header before PRE_ACK, and the shared master key $\kappa_{0,1} = \alpha^{n_S n_1} \bmod p$ between SI and IN_1, i.e. $RMAC_0 = h(SIP\ header\ before\ PRE\_ACK | \kappa_{0,1})$. At this time, the PRE_ACK field could be set to any value.

When an IN_i receives the REWARD packet, it will verify the RMAC first and then see if it is one of the nodes that should be rewarded by checking the rewarding node list. If yes, it will keep the credits as pending credits (μ times *Number of Bytes* in the REWARD packet) and then replaces the PRE_ACK field and RMAC field with new values as shown in Fig. 2, where $\kappa_{i,i+1}$ denotes the master key shared between IN_i and IN_{i+1}, i.e., $\alpha^{n_i n_{i+1}} \bmod p$. Subsequently, it forwards the REWARD packet to the next node on the rewarding list. Because of the broadcast nature of wireless channel, when the next hop node IN_{i+1} is forwarding the REWARD packet further to its next hop, IN_i is able to overhear that packet. The overheard PRE_ACK field is then sent to the smartcard of the IN_i as its proof of packet forwarding. At this time, the CC of node IN_i could be increased by the number of pending credits after verifying the overheard PRE_ACK value.

After SI overhears the REWARD packet forwarded by the first IN and authenticates the value of PRE-ACK for it, it will increase its own CC by half of the product of $N_{IN}\mu$ times *Number of Bytes* contained in the REWARD packet. Upon receiving the REWARD packet, SR is also able to increase its CC by the credits that have been overcharged, and then forms a short RRACK-type packet which contains only the PRE_ACK and send it back to the last IN. Once receiving this RRACK packet, the last IN will be able to increase its CC by the pending credits. In this way, SI and SR share the cost for the session traffic. We notice that there is no explicit motivation for SR to send the RRACK packet to the last IN. However, since the last IN is serving SR, if SR does not confirm its forwarded REWARD packet, it could choose to stop serving SR. So the SR is implicitly urged to acknowledge the REWARD packet forwarded by the last IN with the aim of avoiding service interruption. And similar to the RECEIPT packets, we could piggyback the REWARD packets in SIP data packets sent from SI to SR as well.

In our charging and rewarding scheme, *Rewarding node list* field is used to prevent one IN from transmitting REWARD packets to its colluding nodes who do not participate in the session. *RMAC* field is used to authenticate and protect the integrity of the packet and *PRE_ACK* field is used to confirm the forwarding of the packet. The forwarding at the current node is confirmed by overhearing the transmission of the same packet at the next hop node. The cryptographic operations designed could prevent the forgery of such information. Another possible attack is called *replay attack*, e.g., INs cheat for credits by replaying the old REWARD packets. To defend against this attack, each MN maintains a SEQUENCE table for the currently active sessions. Each entry in the SEQUENCE table consists of the node (SI) ID, the session number, and the most recent received inside-session packet sequence number of the REWARD packet for that session. The REWARD packet that has been processed before will not be processed again.

IV. SIMULATION STUDY

In this section, we evaluate the overhead of SIP and its impact on packet delivery ratio (PDR) using simulation. The simulation is implemented within the GloMoSim V2.03. We simulated an ad hoc network with 50 nodes randomly deployed in an area of 700x700 square meters. The MAC layer protocol used is the Distributed Coordination Function (DCF) of IEEE 802.11. Radio propagation range for each node is 250 meters, channel capacity is 2 Mbits/sec, and the underlying routing protocol used is AODV.

The random waypoint mobility model is used in the simulation and the node speed is within [0,20] m/s. And the different node mobility levels are achieved by changing the values of pause time. The traffic used in each simulation run is 20 CBR sessions. All the data packets are 512 bytes and are sent at a speed of 4 pkts/second. Each simulation is executed for 15 simulated minutes. Each result is averaged over 20 simulations using 20 different random seeds.

For simplicity, we assume each node has a constant charging rate of 1 credit/packet. To make our simulation more realistic, we introduce into each IN a power counter representing its residual power measured in number of packets one IN could transmit. The power counter is decreased by one each time a packet is forwarded.

If we assume that the initial values of CC and the power counter are A and B respectively, the average number of

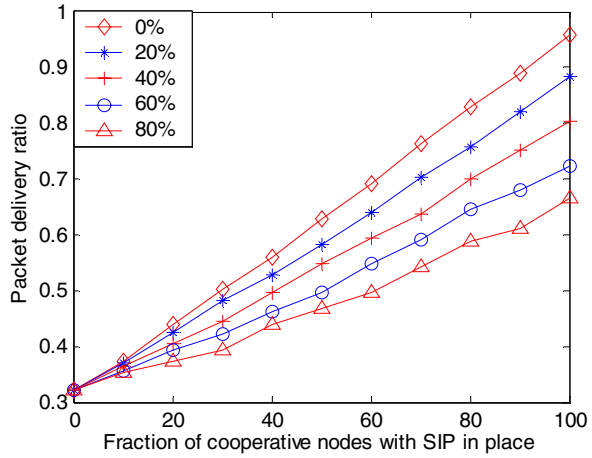

Figure 3. The impact of SIP on PDR

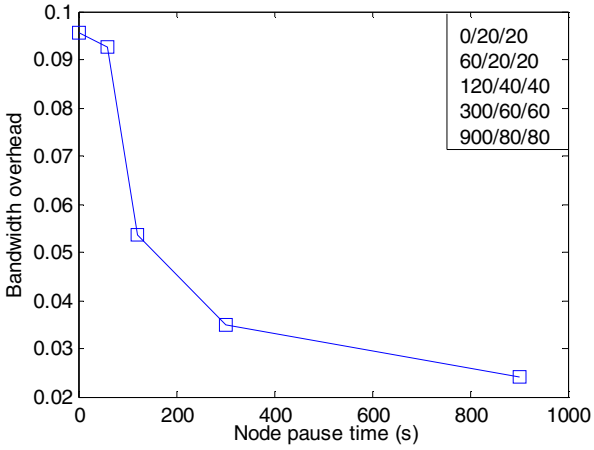

Figure 4. The bandwidth overhead of SIP protocol.

intermediate nodes is L, and the number of packets forwarded is C, then there exists a watershed as $(B-C)*L/2 \le A+C$. The left term denotes the number of credits one intermediate node needs to use up its residual power for transmitting its own data through L intermediate nodes after forwarding C packets for others, while the right term is its current CC value. The left term is set initially larger, but becomes smaller than the right term after some time. When attaining the watershed, nodes may choose not to forward packets for others any more since its current credits could cover their own potential traffic according to their residual power and other resources. We call those kinds of nodes *near-sighted nodes*. On the contrary, we use *far-sighted* nodes to denote those which always attempt to relay packets for others to gain as many as credits as possible for later use.

A. The impact of SIP on PDR.

Fig. 3 shows the impact of SIP on the packet delivery ratio (PDR) with the increase of cooperative nodes, where the X-axis refers to the fraction of cooperative nodes with SIP in place among all the 50 nodes and the rest are selfish nodes, i.e. the non-cooperative nodes that are always reluctant to serve others. Different curves in Fig. 3 reflect different ratios of the *near-sighted* nodes among all the cooperative nodes. The lowest point in the figure means that all the nodes are selfish. Therefore, a packet could only be successfully delivered when its source and destination are neighbors. We could see that with the increase of the number of cooperative nodes, the PDR increased significantly, which is intuitive. The figure also suggests that SIP indeed helps improve the PDR even when many cooperative nodes are the *near-sighted* nodes.

B. Network bandwidth overhead of SIP

Fig. 4 shows the network bandwidth overhead of SIP defined as the total number of SIP control packets versus the total number of data packets in the simulation time. And we adjust the sending frequency of RECEIPT/REWARD packets to remunerate INs in a timely manner according to the network mobility level. The notation *t/r/w* indicates the network mobility – pause time is *t* seconds, and the RECEIPT and REWARD frequency – a RECEIPT is sent for every *r* data packets while REWARD is sent for every *w* data packets. When the network mobility turns high, SIP control packets such as KAGREE/KACK/KSYNC packets need to be sent more frequently due to the frequent occurrence of routing errors. Combined with the increasing frequency of RECEIPT/REWARD packets, it will results in the increase of bandwidth overhead. We can observe this tendency from the figure. However, as we can see, the bandwidth overhead always remains at a low level.

V. CONCLUSIONS AND FUTURE WORK

In this paper we present SIP, a Secure Incentive Protocol for stimulating packet forwarding among selfish nodes in MANETs. SIP is based on a credit-based charging and rewarding scheme by which nodes in an ad hoc network are charged or rewarded for the service they receive or provide, and thus possible *selfish* but rational nodes would have an incentive to be cooperative. In order to prevent "credit fraudulence", SIP is carefully designed to guarantee the correct functioning of charging and rewarding, and also to defend a wide range of attacks. SIP is totally self-configured and does not require any pre-deployed infrastructure. It takes a source controlled and session-based approach to reduce the communication and computation overhead. The simulation results indicate that SIP is effective and lightweight with moderate implementation complexity.

The mechanism introduced in this paper is our first step on this issue and only applies to unicast traffic. Due to the space limitation, we will report the extensions of SIP to multicast traffic and other optimizations such as different charging rates for making SIP more flexible and adaptable to network dynamics in another separate paper.

REFERENCES

[1] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of ACM MobiCom*, Boston, Aug. 2000.

[2] S. Buchegger and J.Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes- Fairness In Distributed Ad Hoc NeTworks," in *Proc. of IEEE/ACM MobiHoc*, Lausanne, June 2002.

[3] K. Paul and D. Westhoff, "Context Aware Detection of Selfish Nodes in DSR based Ad-hoc Networks," in *Proc. of IEEE GLOBECOM*, Taiwan, China, Nov. 2002.

[4] Y. Liu and Y.R. Yang, "Reputation Propagation and Agreement in Mobile Ad-Hoc Networks", in *Proc. of IEEE WCNC*, New Orleans, Mar. 2003.

[5] L. Buttyan and J.P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs", in *Proc. of IEEE/ACM MobiHoc*, Boston, Aug. 2000.

[6] L. Buttyan and J.P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM Journal for Mobile Networks (MONET)*, Vol. 8, No. 5, Oct. 2003.

[7] L. Anderegg and S. Eidenbenz, "Ad hoc-VCG: A Trustful and Cost-Efficient Routing Protocol for Mobile Ad hoc Networks with Selfish Agents," in *Proc. of ACM MobiCom*, San Diego, Sep. 2003.

[8] B. Lamparter, K. Paul, and D. Westhoff, "Charging support for ad hoc stub networks," *Journal of Computer Communication, Special Issue on Internet Pricing and Charging: Algorithms, Technology and Applications*, Elsevier Science, Summer 2003.

[9] N.B. Salem, L. Buttyan, J.P.Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks," in *Proc. of IEEE/ACM MobiHoc*, Annapolis, June 2003.

[10] S. Zhong, J. Chen, and Y.R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," in *Proc. of IEEE Infocom*, San Francisco, Apr. 2003.

[11] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," in *Proc. of IEEE Infocom*, San Francisco, Apr. 2003.

[12] National Institute of Standards and Technology, "SECURITY RREQUIREMENTS FOR CRYPTOGRAPHIC MODULES," FIPS 140-1, Jan. 1994.

[13] A.J. Menezes, P.C. van Oorschot, S.A. Vanston: Handbook of Applied Cryptography, CRC Press, ISBN 0-8493-8523-7, 1996.