ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.9, September 2009, pp.2542−2557
doi: 10.3724/SP.J.1001.2009.00570

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

无线 Ad Hoc 网络最大生命周期路由算法的诚实机制*

谢志鹏[1+], 张 卿[2]

[1](复旦大学 计算机科学技术学院,上海 200433)

[2](Samsung Electronics Co., Ltd., Hwasung-City 445-701, South Korea)

Truthful Mechanisms for Maximum Lifetime Routing in Wireless Ad Hoc Networks

XIE Zhi-Peng[1+], ZHANG Qing[2]

[1](School of Computer Science, Fudan University, Shanghai 200433, China)

[2](Samsung Electronics Co., Ltd., Hwasung-City 445-701, South Korea)

+ Corresponding author: E-mail: xiezp@fudan.edu.cn

Xie ZP, Zhang Q. Truthful mechanisms for maximum lifetime routing in wireless ad hoc networks. *Journal of Software*, 2009,20(9):2542−2557. http://www.jos.org.cn/1000-9825/570.htm

Abstract: In this paper, existing lifetime-aware routing algorithms are divided into two categories: General Max-Min (GMM) algorithm and Conditional Max-Min (CMM) algorithm. Motivated by algorithmic mechanism design, two truthful mechanisms for these two types of algorithms are proposed: SMM for GMM algorithm and SMM-VCG for CMM algorithm. By giving appropriate payments to the relay nodes, the proposed mechanisms gurantee that existing algorithms achieve their design objectives even in the presence of selfish nodes. It is shown that the payment ratio is relatively small and stable due to the nature of lifetime-aware routing algorithms, which is also confirmed by experiments.
Key words: wireless ad hoc network; maximum lifetime; truthful mechanism

摘 要: 将已有的生命周期路由算法分成两类:普通 Max-Min(GMM)算法和条件 Max-Min(CMM)算法,然后为这两类算法分别提出它们的诚实机制.通过给予中继节点适当的报酬,这些诚实机制可以确保已有的算法在面对自私节点的时候也可以实现它们的设计目标.说明生命周期路由算法的本质可以使这种报酬率相对较低且比较稳定,实验结果也进一步证明了这一点.
关键词: 无线 Ad Hoc 网络;最大化生命周期;诚实机制
中图法分类号: TP393 **文献标识码**: A

1 Introduction

Wireless ad hoc network is a network composed of a collection of mobile nodes. The communication of nodes can be carried out without the support of any fixed infrastructure. A node can communicate directly with other

* Supported by the National Natural Science Foundation of China under Grant No.60503025 (国家自然科学基金)
Received 2008-04-29; Accepted 2008-08-21

nodes within its transmission range. To communicate with nodes beyond the range, intermediate nodes are required for relaying packets hop by hop. An ad hoc routing protocol is used to find paths between nodes (see Ref.[13] for a survey).

Due to the limited battery power of node, power-conservation is an important design criterion for ad hoc routing protocol. An ad hoc routing protocol should be power-aware. Current research on power-aware routing mainly focuses on two aspects: minimizing the consumed energy of communication (i.e. energy-efficient) and maximizing the lifetime of the whole network (i.e. lifetime-aware). An energy-efficient routing protocol tries to find a path which has the minimal consumed energy. However, the nodes in the minimal energy path will be drained out of energy quickly if all the packets are routed along this path, which results in the partition of the network. Therefore, it would be better to distribute the traffic load among nodes, which is discussed by lifetime-aware routing protocols.

Most previous works on power-aware routing have implicitly assumed that nodes are cooperative and truthful. A node is cooperative means that the node is willing to relay packets for other nodes. A node is truthful means that it will reveal its private information, such as its residual energy etc. However, these assumptions cannot be taken for granted from the view of an individual node. For a node, some resources, such as battery power, are scarce, and the forwarding actions will drain its battery. A node may tend to be selfish. It can refuse to relay packets for others, or just claim a very low residual energy so that it cannot be selected as a relay node by lifetime-aware routing protocols. We cannot simply expect the nodes to follow the designed routing protocols faithfully.

Selfishness has been actively studied in wireless ad hoc networks in recent years. Several protocols have been proposed to stimulate cooperation. These protocols are either punishment-based or payment-based. Nodes are punished for non-cooperation in punishment-based models while awarded for cooperation in payment-based models. Further more, Ref.[1] proposed Ad hoc-VCG, a reactive routing protocol ensures that nodes are cooperative and truthful while also achieving the globally desirable objective of energy-efficiency. However, as we have pointed out, lifetime is also an important metric for network. In the face of selfish nodes, how to make existing lifetime-aware routing algorithms achieve their design objectives is an imperative problem to be solved.

In this paper, we investigate the maximum lifetime routing mechanism design problem in ad hoc networks. We intensively study existing lifetime-aware routing algorithms, and divide them into two categories: General Max-Min (GMM) algorithm and Conditional Max-Min (CMM) algorithm. GMM algorithm selects the path which has the maximal lifetime. CMM algorithm selects the path which can maximize another criterion if there exist several paths whose lifetimes are longer than some threshold. Based on these two types of algorithms, we propose two mechanisms, Second-Max-Min (SMM) and SMM-VCG, which cope with the selfishness in the framework of algorithmic mechanism design[12]. Our mechanisms are truthful and work properly even nodes in network being selfish. We also present a DSR-like routing protocol Ad hoc-SMM to implement the proposed mechanisms.

The rest of the paper is organized as follows. Section 2 reviews some related work. Section 3 analyzes the mechanism design problem for lifetime-aware routing in wireless ad hoc networks. Section 4 proposes SMM mechanism for GMM algorithm, and proves the truthfulness of SMM mechanism. Section 5 proposes SMM-VCG mechanism for CMM algorithm. Section 6 presents an Ad hoc-SMM protocol for the implementation of mechanisms. Section 7 implements experiments to evaluate the performance of payment ratio. We conclude our work in Section 8.

2 Related Work

In this section, we review some related works. Existing lifetime-aware algorithms are introduced first. They do not consider the effect of selfish nodes. We find that the theory of algorithmic mechanism design is quite suitable for the analysis of our problem, thus we provide a brief introduction to the concept of mechanism design.

2.1 Lifetime-Aware algorithms

Several lifetime-aware algorithms have been proposed to maximize the lifetime of network. Min-Max battery cost routing (MMBCR) algorithm[14] avoids the path with nodes having the least battery capacity among all nodes in all possible paths. Conditional Max-Min battery capacity routing (CMMBCR) algorithm[14] selects the path with minimum total transmission power when there exist some possible paths and all nodes in these path have sufficient residual battery power (i.e., above a threshold). Maximum Residual Packet Capacity (MRPC) algorithm[10] identifies the capacity of a node not only by its residual battery power, but also by the expected energy spent in reliably forwarding a packet over a specific link, and selects the path which has the largest packet capacity at the critical node. Conditional Maximum Residual Packet Capacity (CMRPC) algorithm[10] is a conditional variant of MRPC, which uses minimum-energy path when there exist more than one paths and the packet capacity at all the nodes in these paths above a specific threshold. Lifetime Prediction Routing (LPR) algorithm[11] minimizes the variance in the remaining power of all the nodes and thereby prolongs the network lifetime.

These algorithms use well-defined metrics to describe the lifetime of a node. A metric usually considers parameters like residual battery power of nodes, transmission power between two connected nodes and others. By considering the residual battery power of nodes, a routing algorithm distributes the power consumption evenly among nodes. Besides the residual battery power of nodes, if a routing algorithm also considers the overall transmission power of a path, it will minimize the power consumption at the same time.

2.2 Algorithmic mechanism design

Algorithmic mechanism design considers the problems in a distributed environment where the participants cannot be assumed to follow the algorithm but rather their own self-interest. Reference [12] proposed the concept of algorithmic mechanism design, and gave a formal model to solve the mechanism design problems. The model can be described as following:

In a distributed environment, there are n agents. Each agent i has some private information t^i, called its *type*. For a mechanism design problem, there is an *output specification* that maps each type vector to a set of allowed outputs $o\in O$. Agent i's preferences are given by a *valuation function* $v(o,t^i)$. A *mechanism* defines a family of *strategies* A^i for each agent i. For each strategy vector $(a^1,\ldots,a^n)$, $a^i\in A^i$, the mechanism computes an *output* $o=o(a^1,\ldots,a^n)$, and a *payment* vector $p=p(p^1,\ldots,p^n)$. Agent i's *utility* is $u^i=p^i-v(o,t^i)$. It is the i's goal to maximize its utility. A mechanism is called *truthful* if for every agent i of type t^i and for every strategy a^{-i} of the other agents, i's utility is maximized when it declares its type t^i.

Several typical problems have been solved mechanism design problem. Reference [12] used the Vickrey-Clarke-Groves(VCG) mechanism[3,6,15] solving the shortest path problem, and designed an n-approximation mechanism for task scheduling problems. Other applications include BGP-based mechanism for low-cost routing[4], optimization problems in congestion control[8], and load balancing in distributed systems[5] etc.

2.3 Algorithmic mechanism design in ad hoc networks

In the context of wireless ad hoc networks, Refs.[1,16,18] etc. applied mechanism design theory to solve problems in ad hoc networks. Reference [1] investigated cost-efficient routing problem. It proposed a reactive routing protocol Ad hoc-VCG, which achieved the design objectives of truthfulness and cost-efficiency by paying to the intermediate nodes a premium over their actual costs for forwarding packets. If additionally assumed that all nodes have the same cost-of-energy, Ad hoc-VCG would choose the most energy-efficient path. Reference [18] investigated both routing and forwarding problems, and proposed Corsac protocol which integrates VCG mechanism with a novel cryptographic technique to address the challenge in ad hoc networks that a link cost is determined by two nodes together. Corsac applies efficient cryptographic techniques to designing a forwarding protocol to enforce routing decision such that fulfilling a routing decision can bring maximum utility to node. Reference [19] presented a systematic study of collusion resistance problem in incentive-compatible routing, where a group of nodes may cooperate to cheat. It showed that achieving Group Strategy Proof is impossible while achieving Strong Nash Equilibrium is possible. More specifically, it designed a scheme to achieve Strong Nash Equilibrium for collusion resistance, and gave a cryptographic method that prevents profit transfer between colluding nodes. Reference [16] investigated multicast problem, and designed several truthful multicast mechanisms without using VCG.

3 Problem Analysis

In this section, we present the mechanism design problem for lifetime-aware routing in wireless ad hoc networks by considering the behavior of each individual node. We analyze existing algorithms, divide them into two groups, and show that these algorithms cannot work properly in the case of selfish nodes.

A wireless ad hoc network *N* consists of a set of mobile nodes. The nodes have the following characteristics:

- Each node has some private information, such as its lifetime, and transmission cost etc;
- Every node is selfish but economically rational. A rational node means that the node is willing to forward packets for other nodes only when it can get payment equal to or greater than what it valuates;
- The goal of each node is to maximize its own utility. To get more utility, it may check about its private information;
- Each node belongs to different users. It means that the nodes would not collaborate to cheat.

Now, a source node *S* wants to send a message to a destination node *D*. There are *n* possible paths that can be found by using a DSR-like routing protocol. We should design a mechanism which can select a truthful path from all possible paths to maximize the lifetime of network.

Section 2 has reviewed the solutions that do not consider the influence of selfish nodes. We can divide these algorithms into two categories by algorithm structures:

- General Max-Min (GMM) algorithm

  In this type of algorithms, all nodes use a common metric of lifetime, which can be specified by a function $g(\cdot)$. The factors, such as the residual battery power of nodes and the transmission power between nodes, can be used as the parameters of function $g(\cdot)$. We treat the minimal lifetime of nodes in a path as *lifetime of path*. This algorithm favors the path which has the maximal lifetime, i.e. it selects a path as follows:

  $$\operatorname*{Max}_{j\in A}(\operatorname*{Min}_{i\in j}(g(R_j^i,...))) \quad (1)$$

  where R_j^i is the residual battery power of node i in path j, and A is the set of all possible paths. Algorithms such as MMBCR, MRPC, and LPR belong to the type of GMM.

- Conditional Max-Min (CMM) algorithm

  In this type of algorithms, if there exists several paths whose lifetime are longer than a specified threshold γ, it selects the path which can minimize another criterion $F(\cdot)$, i.e. it selects a path as follows:

Let $Q=\{\underset{i\in j}{\mathrm{Min}}(g(R_j^i,...))\geq\gamma, j\in A\}$, $o=\begin{cases}\underset{j\in A}{\mathrm{Max}}(\underset{i\in j}{\mathrm{Min}}(g(R_j^i,...))), & Q=\varnothing\\ \underset{j\in Q}{\mathrm{Min}}\left(\sum_{i\in j} f(T_j^i,...)\right), & Q\neq\varnothing\end{cases}$ (2)

where T_j^i is the transmission energy level of node i in path j, and $f(\cdot)$ is a representation of transmission cost of nodes. $F(\cdot)=\sum_{i\in j} f(T_j^i,...)$ is the total transmission cost of path j.

It can be seen that CMM algorithm integrates the advantage of an energy-efficient routing algorithm with the advantage of a lifetime-aware routing algorithm. It tries to satisfy two constraints simultaneously: distributing the power consumption of each node evenly and minimizing the overall transmission energy for each connection. Algorithms such as CMMBCR and CMRPC belong to the type of CMM.

As we have mentioned, a node may tend to be selfish. To prevent its battery power being consumed for other nodes, a node i may refuse to relay packets, or declare a very low lifetime so that the path j (i belongs to j) will not be selected as the output path by existing algorithms. In this situation, GMM and CMM will fail to work properly.

In this paper, we deal with the maximal lifetime routing problems while considering selfish nodes at the same time. Our objective is to design mechanism which ensures to route along the path selected by GMM or CMM algorithm, such that the path is truthful.

4 SMM Mechanism and Analysis

In this section, we propose SMM mechanism for GMM algorithm. Instead of considering the behavior of every node, we treat the nodes in a path as a whole in SMM and select the path with the maximum lifetime while pay it according to the second maximum lifetime among the paths' lifetime.

4.1 SMM mechanism for GMM

To design a mechanism, we should provide its output function, define practical valuation function of nodes, and present appropriate payment function.

The output function $o(\cdot)$ is given firstly. According to the lifetime declaration of each node (a node can declare its lifetime at will), the output function $o(\cdot)$ selects a path from all possible paths by using GMM algorithm. We assume that there are n possible paths between the source node S and the destination node D, and there are m nodes in all these n paths. The output function $o(\cdot)$ can be represented as follows:

$$o(a^1...a^m)=\underset{j\in A}{\mathrm{Max}}(\underset{i\in j}{\mathrm{Min}}(a_j^i)) \quad (3)$$

where a_j^i is the lifetime declaration of node i in path j.

The valuation function of nodes is defined by

$$v(o,t^i)=\begin{cases}0, & i\notin o\\ \dfrac{c}{t^i}, & i\in o\end{cases} \quad (4)$$

where t^i is the lifetime of node i.

Equation (4) means that if i does not belong to the output path o, i's evaluation would be 0; otherwise, i's evaluation is inversely proportion to its lifetime. Intuitively, the shorter lifetime a node i has, the more likely that i is not willing to forward packets for others. Therefore, node i would expect more payment if the source node S wants i to forward packets for him.

The goal of a selfish node is to maximize its utility, so it tends to choose favorable strategy and becomes truthless. If a node declares false lifetime, the output function $o(\cdot)$ may select an improper path because the output is calculated according to the lifetime declaration of each node. We have to design an appropriate payment function which can meet the requirements of nodes while compatible with the output of the algorithm.

We treat the minimal lifetime declaration of nodes in path j as j's lifetime declaration. We assume that the nodes whose lifetime declaration is minimal in all n possible paths are $q_1,\ldots,q_n$, thus the lifetime declarations of these n paths are $a_1^{q_1},\ldots,a_{n-1}^{q_{n-1}},a_n^{q_n}$. We can simply denote them by $a_1,\ldots,a_{n-1},a_n$. Let path o be the path which has maximum lifetime declaration among all these n paths (o's lifetime declaration is a_o, it is the output path), and let path s be the path which has the second maximum lifetime declaration among all these n paths (its lifetime declaration is a_s). We define the payment function $p(\cdot)$ by

$$p(a_j^i)=\begin{cases}0, & \forall j\neq o, i\in j\\ \dfrac{c}{a_s}, & j=o, i\in o\end{cases} \tag{5}$$

Eq.(5) means that the payment to nodes which do not belong to the output path would be 0, and the payment to nodes in the output path is related to path s, i.e. the payment to nodes in the output path is related to the path which has the second maximum lifetime declaration in all possible paths.

We call our mechanism the Second-Max-Min (SMM) mechanism and will analyze it in the next section. We will prove that SMM mechanism is truthful, i.e. every node's strategy is to report its type, so the lifetime declaration of each node is equal to its type ($a_j^i=t_j^i$).

Let us consider the network in Figure 1 as an example. There are three paths between S and D: Si_1D, Si_2i_3D, and Si_2i_4D. The lifetime of Si_1D is 3, the lifetime of Si_2i_3D is 4, and the lifetime of Si_2i_4D is 2. By applying SMM mechanism, we choose Si_2i_3D, which holds the maximum lifetime in all these three paths, as the output path. The payment to nodes (i_2,i_3) in Si_2i_3D is related to Si_1D, which has the second maximum lifetime (3 in this case) in all these three paths. The payment to i_2 is $\frac{c}{3}$, which is larger than its valuation $\frac{c}{5}$, and i_2's utility is $\frac{c}{3}-\frac{c}{5}$. The payment to i_3 is $\frac{c}{3}$, which is also larger than its valuation $\frac{c}{4}$, and i_3's utility is $\frac{c}{3}-\frac{c}{4}$. The payment to i_1 and i_4 is 0.

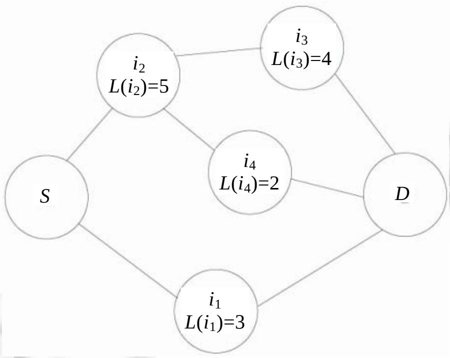

Fig.1 An example for SMM mechanism

4.2 SMM mechanism analysis

Here we show that the design of SMM mechanism can ensure GMM algorithm to get its desired output in the presence of selfish nodes.

First, SMM mechanism guarantees the voluntary participation of all nodes. If a node can get payment equal to or greater than its valuation (i.e. its utility is non-negative), it is willing to participate in the protocols. In our mechanism, if a node does not belong to the path selected as the route from S to D, its valuation is 0 and its payment is 0 too, thus its utility is 0. If a node i belongs to the output path j, its valuation is equal to or less than c/a_j, while i's payment is c/a_s, where s is the path which has the second maximum lifetime declaration in all possible paths. The utility of i is $c/a_s-c/a_j$, which is non-negative because $a_s \le a_j$.

Second, SMM mechanism is truthful, i.e. every node is truthful and declares its type. It is clear that if all nodes declare their type, our mechanism guarantees the source node chooses algorithm desired path. Here, we prove that SMM mechanism is truthful.

Theorem 1. SMM mechanism is truthful.

Proof: To prove our mechanism is truthful, we need to show that every node cannot get more utility than what it gets when declaring its true lifetime, that is, cheating cannot increase the utility of a node. It is proved in two steps: (1) The lifetime declaration of each path is truthful; (2) The lifetime declaration of each node in a path is truthful.

(1) The lifetime declaration of each path is truthful.

Here, we treat all the nodes in a path as a whole, and consider the behavior of a path. A path may be untruthful, and it can over-declare or under-declare its lifetime. The function $o(\cdot)$ selects a path as the output according to the lifetime declaration of each path. We assume that there are n possible paths between S and D. When every node declares its true lifetime, let path o be the path having maximum lifetime (o's lifetime is t_o, and o is the output path).

1) Prove that the utility of every node in o is maximized when o declares its true lifetime no matter what lifetimes the other paths declare.

The utility of a node i in o is determined by the path s having second maximum lifetime. We consider the behavior of s.

(I) s declares its true lifetime t_s ($a_s=t_s$).

– If o declares its true lifetime t_o ($a_o=t_o$), i's utility is $u_1=\frac{c}{a_s}-\frac{c}{t_o^i}$, which is larger than 0 since $a_s \le t_o \le t_o^i$.

  Otherwise

– If o declares false lifetime a_o and $a_o>t_o$, i's utility is still u_1 because o still can be selected as the output path.

– If o declares false lifetime a_o and $a_o<t_o$, (A) if $a_s \le a_o<t_o$, i's utility is still u_1; (B) if $a_o<a_s$, o cannot be selected as the output path, and i's utility is 0.

From (I) we can get that: if s declares its true lifetime, the utilities of nodes in o are maximized when o declares its true lifetime.

(II) s declares false lifetime a_s and $a_s<t_s$. The analysis is similar to that for (I).

(III) s declares false lifetime a_s and $a_s>t_s$:

(i) If $t_o \ge a_s>t_s$, the analysis is similar to that for (I).

(ii) If $a_s>t_o$,

– if o declares its true lifetime t_o ($a_o=t_o$), i's utility is 0 since o cannot be selected as the output path.

– if o declares false lifetime a_o and $a_o>t_o$, (A) if $t_o<a_s \le a_o$, i's utility is $\frac{c}{a_s}-\frac{c}{t_o^i}$. There must exist some nodes in o, such as the node with the minimal lifetime, whose utilities are smaller than 0 since $a_s>t_o$; (B) if $t_o<a_o<a_s$, o cannot be selected as the output path, and its utility is 0.

– if o declares false lifetime a_o and $a_o<t_o$, o cannot be selected as the output path, and i's utility is 0.

From (II) and (III), we can also get that the utility of every node in *o* is maximized when *o* declares its true lifetime if *s* declares false lifetime.

Table 1 gives a list of *i*’s utility in all the possible cases discussed above.

2) Prove that the utility of every node in path *j* ($t_j \le t_o$) is maximized when *j* declares its true lifetime no matter what lifetimes the other paths declare.

The utility of a node *i* in path *j* is related to the path *o* which has maximal lifetime declaration a_o among all the other paths (do not include *j*). If $a_j \le a_o$, *j* cannot be selected as the output path, and *i*’s utility is 0. If $a_j > a_o$, *j* can be selected as the output path, and *o* would be the second maximum lifetime declaration path. *j*’s utility is related to path *o*. Table 2 gives a list of *i*’s utility in all the possible cases. Here we do not discuss them in detail. As an example, we analyze the situation when $a_o=t_o$, $a_j>t_j$ and $a_j>a_o$. In this case, *j* can be selected as the output path and *o* is the second maximum lifetime declaration path. *i*’s utility is $\frac{c}{a_o}-\frac{c}{t_j^i}$. There must exist some nodes in *j*, such as the node with the minimal lifetime, whose utilities are smaller than 0 since $a_o=t_o>t_j$.

Table 1 Utility of a node in path *o*

o’s utility		$a_s=t_s$	$a_s<t_s$	$a_s>t_s$	
				$<t_o$	$>t_o$
$a_o=t_o$		$u_1\ge 0$	$u_2>0$	$u_3>0$	0
$a_o>t_o$	$\ge a_s$	u_1	u_2	u_3	≤ 0
	$<a_s$				0
$a_o<t_o$	$\ge a_s$	u_1	u_2	0	0
	$<a_s$	0	0		

Table 2 Utility of a node in path *j*

j’s utility		$a_o=t_o$	$a_o<t_o$	$a_o>t_o$	
				$>t_j$	$<t_j$
$a_j=t_j$		0	0	0	$u_1>0$
$a_j>t_j$	$\le a_o$	0	0	0	u_1
	$>a_o$	<0	<0	<0	
$a_j<t_j$	$\le a_o$	0	0	0	0
	$>a_o$				u_1

(2) The lifetime declaration of each node in a path *j* ($j\ne o$) is truthful.

For a node *i* in path *j*,

1) *i* is the node which has the minimal lifetime in path *j*, then *i*’s behavior represents the behavior of path *j*. If *i* over-declares its lifetime, the lifetime declaration of path *j* will be over-declared. If *i* under-declares its lifetime, the lifetime declaration of path *j* will be under-declared. We can analyze this kind of situation similarly to the analysis in step (1) above, so we omit it here.

2) *i* is not the node which has the minimal lifetime in path *j*. We assume that *k* is the node which has the minimal lifetime in path *j*, and *i*’s lifetime is t_j^i, *k*’s lifetime is t_j^k. If *i* declares false lifetime $a_j^i > t_j^i$, *i* cannot get more utility because *i*’s utility is determined by *k*. If *i* declares false lifetime $a_j^i \le t_j^i$: (A) $t_j^k \le a_j^i < t_j^i$, *i* cannot get more utility; (B) $a_j^i < t_j^k$, *i* will be the node which has the minimal lifetime declaration. We can also analyze it similarly to the analysis in step (1) above, so we omit it here.

All analyses above are based on the assumption that every node’s lifetime declaration information is unknown and cannot be altered by others, which is ensured by the PKI-based security model to be discussed in Section 6. If a node *i* can alter the lifetime declaration of all its predecessors in a path *j*, it will declare its true lifetime and make its lifetime declaration the minimal one by increasing the lifetime declaration of other nodes. This behavior will increase the possibility that path *j* is selected as the output path. It will also increase the possibility that node *i* get more utility at the cost of sacrificing the others’ utility.

To measure the payment, we define *payment ratio*: Let path *j* be the output path, payment ratio is the ratio of payment for path *j* to valuation of path *j*. We have the following theorem for payment ratio.

Theorem 2. For SMM mechanism, let path *o* be the output path, which has the maximum lifetime in all possible paths from *S* to *D*; and path *s* has the second maximum lifetime. Let Max(a_o^i) denote the maximal lifetime declaration of nodes in path o, Min(a_o^i) denotes the minimal lifetime declaration of nodes in path o, and Min(a_s^i) denotes the minimal lifetime declaration of nodes in path s, then

$$\frac{\mathrm{Min}(a_o^i)}{\mathrm{Min}(a_s^i)} \le \beta \le \frac{\mathrm{Max}(a_o^i)}{\mathrm{Min}(a_s^i)} \tag{6}$$

Proof: We assume that the output path o is $\{S, i_1, ..., i_{l_o}, D\}$, then

$$\beta = \frac{\dfrac{l_o \cdot c}{a_s}}{\sum_{k=1}^{l_o} \dfrac{c}{a_o^{i_k}}} = \frac{\dfrac{l_o}{\mathrm{Min}(a_s^i)}}{\sum_{k=1}^{l_o} \dfrac{1}{a_o^{i_k}}} \begin{cases} \le \dfrac{\dfrac{l_o}{\mathrm{Min}(a_s^i)}}{\dfrac{l_o}{\mathrm{Max}(a_o^i)}} = \dfrac{\mathrm{Max}(a_o^i)}{\mathrm{Min}(a_s^i)} \\ \ge \dfrac{\dfrac{l_o}{\mathrm{Min}(a_s^i)}}{\dfrac{l_o}{\mathrm{Min}(a_o^i)}} = \dfrac{\mathrm{Min}(a_o^i)}{\mathrm{Min}(a_s^i)} \end{cases}.$$

The payment ratio β can be used as an important measure to the performance of mechanism. If β is close to 1, the premium that the source node pays to intermediate nodes is low. It means that the mechanism achieves the design objective of algorithm at little additional cost. While β is far larger than 1, the premium that the source node pays to intermediate nodes is high. It means that the mechanism achieves the design objective of algorithm at high additional cost. The essence of a lifetime-aware routing algorithm is to distribute the power consumption evenly among nodes, which leads to the result that the lifetime of nodes has the tendency of closing to each other. From Theorem 2, we can conclude that β is close to 1 when the maximal lifetime of nodes in the output path is close to the minimal lifetime of nodes in path which has the second maximum lifetime. Therefore, we can infer that SMM mechanism has excellent payment ratio, which is relatively small and stable. Our conclusion is confirmed by experiments.

5 SMM-VCG Mechanism and Analysis

In this section, we will consider another type of output function $o(\cdot)$, which uses CMM algorithm. To get CMM algorithm desired path in the presence of selfish nodes, we propose a hybrid mechanism SMM-VCG, which integrates the idea of SMM mechanism with Ad hoc-VCG mechanism[1].

5.1 SMM-VCG mechanism for CMM

Before going into the description of SMM-VCG mechanism, we introduce a definition: the *class* of lifetime.

Definition 1. We distribute the values of lifetime among classes. Using the threshold γ as borderline, and d as step (d>0 and γ>d), the lifetime classes have the format like

$$(\gamma-k_1\cdot d),\ldots,(\gamma-d),\gamma,(\gamma+d),\ldots,(\gamma+k_2\cdot d).$$

If the value of lifetime is L and $(\gamma+k\cdot d)\le L<(\gamma+(k+1)\cdot d)$, L belongs to the class $(\gamma+k\cdot d)$. We denote the class of L as $C(L)$.

Here we provide the output function $o(\cdot)$, valuation function $v(\cdot)$ and payment function $p(\cdot)$ of SMM-VCG mechanism.

The output function $o(\cdot)$ selects a path by using CMM algorithm. $o(\cdot)$ can be represented as follows:

$$\text{Let}\quad Q=\{\mathop{\mathrm{Min}}_{i\in j}(a_j^i)\ge\gamma,\ j\in A\},\ o(a_j^i,b_j^i)=\begin{cases}\mathop{\mathrm{Max}}_{j\in A}(C(\mathop{\mathrm{Min}}_{i\in j}(a_j^i))), & Q=\varnothing \\ \mathop{\mathrm{Min}}_{j\in Q}\left(\sum_{i\in j} b_j^i\right), & Q\ne\varnothing\end{cases} \tag{7}$$

where a_j^i is the lifetime declaration of node *i* in path *j*, and b_j^i is the transmission cost declaration of node *i* in path *j*.

The valuation function of nodes is defined by

$$v(o,t^i,h^i)=\begin{cases}0, & i\notin o\\ \dfrac{c}{C(t^i)}, & Q=\varnothing, i\in o\\ \dfrac{c}{C(t^i)}+h^i, & Q\neq\varnothing, i\in o\end{cases} \tag{8}$$

where h^i is the transmission cost of node *i*, and *c* is a constant (*c*'s value will be determined by the following description).

Eq.(8) means that if *i* does not belong to the output path *o*, *i*'s evaluation would be 0; otherwise

- If $Q=\varnothing$, *i*'s evaluation is only related to its lifetime, since the output function $o(\cdot)$ does not take the transmission cost of *i* into account;
- If $Q\neq\varnothing$, *i*'s evaluation is related to its lifetime and transmission cost, since the output function $o(\cdot)$ considers the lifetime and transmission cost of *i* together.

As we have known, a selfish node may declare its lifetime and transmission cost untruthfully. To get the prospective output in the presence of selfish nodes, the payment function $p(\cdot)$ is designed as follows:

$$p(a_j^i,b_j^i)=\begin{cases}0, & j\neq 0\\ \dfrac{c}{C(a_s)}, & Q=\varnothing, j=o, i\in j\\ \dfrac{c}{\gamma}+VCG(b_j^i), & Q\neq\varnothing, j=o, i\in j\end{cases} \tag{9}$$

where *s* is the path which has the second maximum lifetime declaration in all possible paths, a_s is the lifetime declaration of path *s*, b_j^i is the transmission cost declaration of node *i* in path *j*, γ is the specified threshold, $VCG(b_j^i)$ is the VCG-payment to node *i* in path *j* according to *i*'s transmission cost declaration b_j^i. Assumed that the maximum VCG-payment to nodes is *M* (i.e. $\text{Max}(VCG(b_j^i))\le M$), we choose $c=M\cdot\gamma\cdot\left(\dfrac{\gamma}{d}-1\right)$.

Eq.(9) means that if *i* does not belong to the output path *o*, the payment to *i* is 0; otherwise

- If $Q=\varnothing$, the form of payment is similar to the one in SMM mechanism, since the output *o* is selected by using GMM algorithm;
- If $Q\neq\varnothing$, $p(\cdot)$ is the sum of two parts: SMM payment and VCG payment. SMM payment is different from the payment in SMM mechanism in that the payment is related to the threshold of CMM algorithm instead of the path whose lifetime is the second maximum one in all possible paths. VCG payment is computed by VCG mechanism, which has been illustrated in Ref.[1]. It can be represented as:

$$VCG(b_j^i)=|SP^{-i}|-|SP|+b_j^i \tag{10}$$

where *SP* is the shortest path from *S* to *D*, |*SP*| is the total cost of *SP*, SP^{-i} is the shortest path from *S* to *D* which does not contain node *i* as an intermediate node.

5.2 SMM-VCG mechanism analysis

It is apparent that the payment to a node *i* is larger than the valuation of *i* in SMM-VCG mechanism. Here we focus on proving the truthfulness of SMM-VCG mechanism.

Theorem 3. SMM-VCG mechanism is truthful.

Proof: If $Q=\varnothing$, the mechanism only considers the lifetime declaration of nodes. The truthfulness of SMM-VCG can be implied from Theorem 1.

If $Q\neq\varnothing$, we can also prove the truthfulness of SMM-VCG in two steps: (1) The lifetime declaration of each path is truthful; (2) The lifetime declaration of each node in a path is truthful. The proof is similar to the proof of Theorem 1. Here, we will not discuss them in detail but give some key points in step (1).

1) Consider a path j in Q. The utility of a node i in j is non-negative because j has the chance to be selected as the output path. We assume j's lifetime is a_j, $a_j\geq\gamma$. Now we consider the situation that j declares a false lifetime a_j: (A) If j under-declares its lifetime below γ ($a_j<\gamma$), i's utility must be 0; (B) If j declares its lifetime no less than γ ($a_j\geq\gamma$), SMM payment does not change. Therefore, if i wants to increase its utility, the only thing it can do is to declare a false transmission cost and try to increase its VCG payment. But from Ad hoc-VCG, we know that cheating cannot increase a node's VCG-payment and the transmission cost declaration of a path is truthful too.

2) Consider a path j in $A-Q$. The utility of a node i in j is 0 if j declares its true lifetime a_j ($a_j<\gamma$). Now we consider the case that j declares a false lifetime a_j. If $a_j<\gamma$, i's utility does not change. If $a_j\geq\gamma$: (A) If j is not selected as the output path, i's utility does not change; (B) If j is selected as the output path, i's utility is:

$$\frac{c}{\gamma}+VCG(b_j^i)-\frac{c}{C(t_j^i)}\leq\left(\frac{c}{\gamma}-\frac{c}{\gamma-d}\right)+M=M\cdot\gamma\cdot\left(\frac{\gamma}{d}-1\right)\cdot\left(\frac{1}{\gamma}-\frac{1}{\gamma-d}\right)+M=0\ .$$

It means that i's utility may decrease.

Theorem 4. For SMM-VCG mechanism,

- If $Q=\varnothing$, let path o be the output path, let path s be the path which has the second maximum lifetime. Let $\mathrm{Max}(a_o^i)$ denote the maximal lifetime declaration of nodes in path o, $\mathrm{Min}(a_o^i)$ denote the minimal lifetime declaration of nodes in path o, and $\mathrm{Min}(a_s^i)$ denote the minimal lifetime declaration of nodes in path s, then:

$$\frac{C(\mathrm{Min}(a_o^i))}{C(\mathrm{Min}(a_s^i))}\leq\beta\leq\frac{C(\mathrm{Max}(a_o^i))}{C(\mathrm{Min}(a_s^i))}\qquad(11)$$

- If $Q\neq\varnothing$, let path o be the output path, let $\mathrm{Max}(a_o^i)$ denote the maximal lifetime declaration of nodes in path o, then:

$$\beta<\frac{C(\mathrm{Max}(a_o^i))}{\gamma-d}\qquad(12)$$

Proof: We assume that the output path o is $\{S,i_1,\ldots,i_{l_o},D\}$, then

- If $Q=\varnothing$,

$$\beta=\frac{\dfrac{l_o\cdot c}{C(a_s)}}{\sum_{k=1}^{l_o}\dfrac{c}{C(a_o^{i_k})}}=\frac{\dfrac{l_o}{C(\mathrm{Min}(a_s^i))}}{\sum_{k=1}^{l_o}\dfrac{1}{C(a_o^{i_k})}}\begin{cases}\leq\dfrac{\dfrac{l_o}{C(\mathrm{Min}(a_s^i))}}{\dfrac{l_o}{C(\mathrm{Max}(a_o^i))}}=\dfrac{C(\mathrm{Max}(a_o^i))}{C(\mathrm{Min}(a_s^i))}\\[2ex]\geq\dfrac{\dfrac{l_o}{C(\mathrm{Min}(a_s^i))}}{\dfrac{l_o}{C(\mathrm{Min}(a_o^i))}}=\dfrac{C(\mathrm{Min}(a_o^i))}{C(\mathrm{Min}(a_s^i))}\end{cases}.$$

- If $Q\neq\varnothing$,

$$\beta=\frac{l_o\cdot\frac{c}{\gamma}+\sum_{k=1}^{l_o}VCG(b_o^{i_k})}{\sum_{k=0}^{l_o}\left(\frac{c}{C(a_o^{i_k})}+b_o^{i_k}\right)}<\frac{l_o\cdot\frac{c}{\gamma}+l_o\cdot M}{l_o\cdot\frac{c}{C(\mathrm{Max}(a_o^i))}}=\frac{C(\mathrm{Max}(a_o^i))}{\gamma-d}.$$

6 Ad Hoc-SMM Protocol

To implement our proposed SMM and SMM-VCG mechanisms, we make some modifications to DSR[7] and propose a new Ad hoc-SMM protocol.

DSR is a reactive routing protocol which is used to find the shortest hop path between source and destination. In DSR, when a node wants to establish a route, it broadcasts a route request packet to its neighbors. An intermediate node that received the route request packet processes the request as follows:

- If the node has seen the route request before, discard the packet;
- If the node's address already exists in the route record in the request, discard the packet;
- If the target of the request matches the node's address or exists in the node's cache, return a route reply packet to the source;
- Otherwise, the node re-broadcasts the request to its neighbors.

Here, we propose Ad hoc-SMM protocol based on DSR. Some important points of Ad hoc-SMM protocol are listed below (which makes Ad hoc-SMM different from DSR):

Firstly, we collect the private information of immediate nodes during the transmission of request packet. The information includes lifetime declaration and link cost etc, which is private for each immediate node. We adopt the cryptographic technique proposed in Ref.[18] to collect link costs and prevent cheating. To prevent a node's lifetime declaration information from being known or altered by other nodes, we adopt a PKI (Public Key Infrastructure) based security model. In this model, the keyed *encryption algorithm* is known to all the nodes in the network, the *encryption* and *decryption keys* are generated by the source node. When the source node starts a new route discovery, it puts the encryption key in the route request packet. Every intermediate node uses the encryption key in the received route request packet and the public encryption algorithm to encrypt its private declaration. After receiving the route reply packet, the source node uses the decryption key to decrypt the lifetime declaration of each intermediate node in the packet.

Secondly, we avoid the route cache optimization techniques used in DSR. The cached routes cannot represent the current state of nodes because each node's type keeps changing. In our implementation, the source node periodically refreshes its cache and triggers a new route discovery process, the intermediate node does not respond to the route requests with cached routes.

Thirdly, unlike DSR, a node processes the route request packet even it has seen the request before. We cannot simply discard the packet because the later arrived request packet may have longer minimal lifetime declaration. In our implementation, we ignore the judgment whether the node has seen the request before. To prevent heavy traffic, node will discard a request packet if it has seen the request more than several times.

Fourthly, instead of selecting the shortest-hop path from several possible candidates, we try to choose a path which can maximize the lifetime of network. When the source node wants to send a packet to the destination, it starts a timer and launches the route discovery phase. During a period of time T, the source node may receive several possible paths from the destination. Each path has the information of lifetime and transmission cost declaration of nodes in the path. The source node can choose a path from these paths by using the output function $o(\cdot)$ and calculate the payments to each node in the selected path by using the payment function $p(\cdot)$.

Finally, we consider the payment assignment to nodes. Payment usually exists in the form of virtual concurrency, for example, Nuglet proposed in Ref.[2] and Credit used in Ref.[17]. Two alternative models can be used for payment assignment: source model and central-bank model[1,17]. In source model, the source node is willing to pay to intermediate nodes for relaying packets. There is a strong assumption in source model: the source node acts truthfully (i.e. it would not tell lies about its own information or modify the private information of intermediate nodes). In fact, in the area of mechanism design, it is a tradition to treat the source node in such a special way[9]. With source model, a tamper-proof hardware is required at each node so that correct amount of virtual concurrency is added or deducted from node. In central-bank model, a central bank manages accounts for all network nodes. During a communication session, destination node keeps a record of truly incurred costs (without premiums) that source owes to intermediate nodes, and reports the information to central bank when there is a fine connection. The central bank then credits and debits nodes accordingly. It also pays premiums evenly to intermediate nodes and periodically debits accounts of all nodes evenly in order to compensate for paid premiums, which is treated as paying fee or tax for being part of network[1]. To source node, it is asked to pay for the actual costs to intermediate nodes, thus it is in the source's interest that true maximal lifetime path is computed and it declares its type truthfully. To destination node, it has no cost and it is in the destination's interest to report truthfully. All these make central-bank scheme truthful. With central-bank model, no tamper-proof hardware is required since virtual concurrency is managed by central-bank.

7 Experiments

As we have pointed out, our mechanisms have excellent payment ratio due to the nature of lifetime-aware algorithm. In this section, we conducted experiments to assess it.

The simulation consisted of a network of 50 nodes randomly distributed over a 700×700m^2 area. We used the CBR traffic at 4 packets per second, and the packet size was 512 bytes. Random connections were established. The source node refreshed its cache every other 10 sec. Each node was given enough battery power to finish the experiments. The initial values of battery power in all nodes are same. A node could dynamically adjust its transmission power based on the link distance d, and the transmission cost h is $K\cdot d^{\alpha}$, where α is the signal loss exponent. Four lifetime-aware routing algorithms, MMBCR, MRPC, CMMBCR, and CMRPC, were implemented in our experiments. For each algorithm, we try to find the influence of different parameters (such as the link distance d, the signal loss exponent α) on payment ratio.

In Fig.2, we present the payment ratio of MMBCR by using SMM mechanism when α=2. It can be observed that the ratio payment is very small and close to 1. We compare the situations when the maximum transmission range R of nodes is 150m, 200m and 250m respectively. The effect of transmission range increment lies in two aspects: (1) Each node covers more nodes, so there are more possible paths between the source node and the destination node. It will increase the balance of traffic on nodes and reduce the lifetime variance between nodes; (2) The range of transmission cost will increase due to $h=K\cdot d^{\alpha}$, which increases the lifetime variance between nodes. In Fig.2, we can find that the payment ratio for R=150m is higher than the payment ratio for R=200m and R=250m. This result can be viewed as the effect of the first aspect. The payment ratio for R=200m is close to the payment ratio for R=250m, which can be viewed as the balance between these two aspects.

In Fig.3, we present the payment ratio of MRPC by using SMM mechanism when R=150. In MRPC, the lifetime of a node is the ratio of its residual battery power to its transmission cost (we don't consider the link's packet error probability). Though the initial battery power of all nodes is same, the transmission cost of nodes is different because the transmission cost relates to the link distance. This experiment can be viewed as a simulation of the initial lifetime of all nodes is different. From Fig.3, we see that the payment ratio increases with the increment of α. It is because that the higher α, the higher the difference between initial lifetime of nodes. At the same time, we know that a lifetime-aware routing algorithm tries to minimize the variance of lifetime between nodes to increase the lifetime of network. It can be observed in Fig.3 that the payment ratio decreases with time.

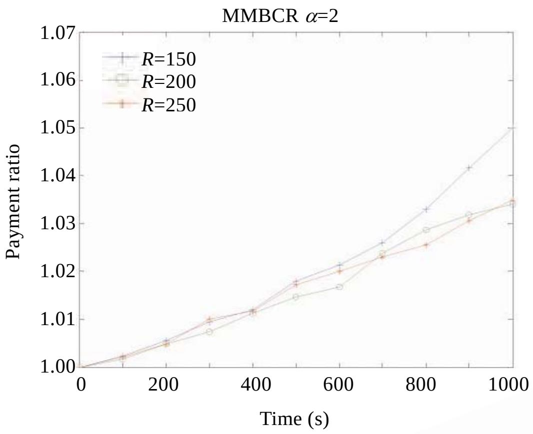

Fig.2 Payment ratio versus maximum transmission range of nodes in MMBCR when α=2

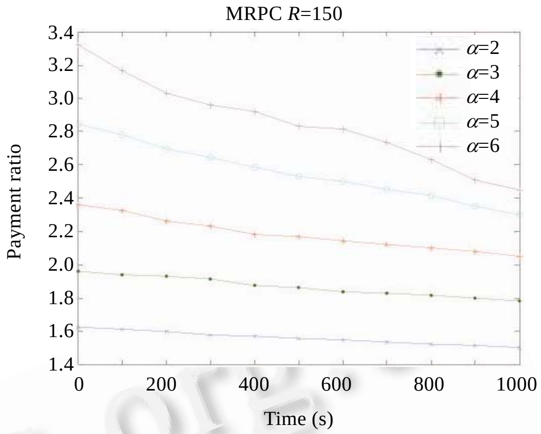

Fig.3 Payment ratio versus signal loss exponent in MRPC when R=150

In Fig.4, we present the payment ratio of CMMBCR by using SMM-VCG mechanism when α=2, γ=80%, d=2%. In SMM-VCG mechanism, SMM payment is much higher than VCG payment, so the dominant effect to payment ratio is the SMM payment. The payment ratio is high at the beginning of simulation because the SMM payment is related to the threshold γ. With the lifetime decrement of nodes, the lifetime of each node is close to γ and the SMM payment decreases too. Thus, the payment ratio can decrease and close to 1. After that, the output path is selected by using MMBCR algorithm, then the payment ratio increases. The lifetime of each node is converted to lifetime class, which causes that the payment ratio and increment speed are higher than that in Fig.2.

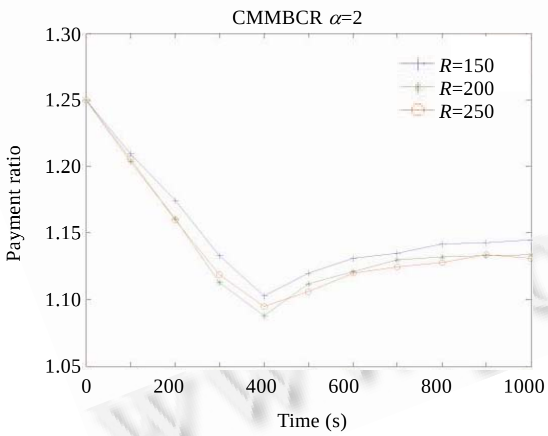

Fig.4 Payment ratio versus maximum transmission range of nodes in CMMBCR when α=2

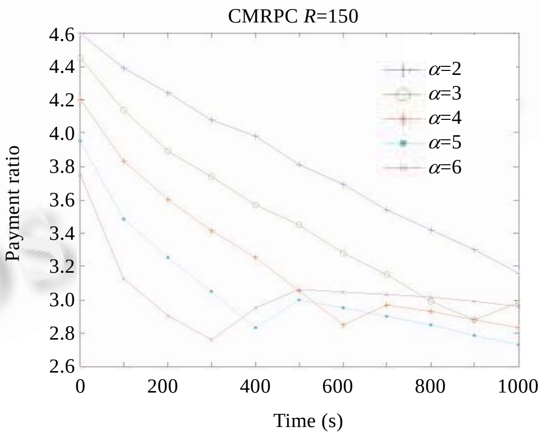

Fig.5 Payment ratio versus signal loss exponent in CMRPC when R=150

In Fig.5, we present the payment ratio of CMRPC by using SMM-VCG mechanism. As we have pointed out in Fig.3, the initial lifetime of nodes is different because the transmission cost of nodes is different. Similar to Fig.4, the payment ratio is high at the beginning of simulations and decreases with the lifetime decrement of nodes. We compare the payment ratio of different α by giving the same threshold γ. The higher α, the closer the initial lifetime of nodes to γ. Therefore, the payment ratio is smaller when α is higher at the beginning of simulations. When the lifetime of nodes reduces to γ, the payment ratio reaches the lowest point. After that, the output path is selected by using CMMBCR algorithm. The payment ratio has some fluctuation: increases first (due to the large lifetime variance between nodes), then decreases similar to Fig.3.

8 Conclusion

In this paper, we dealt with the problem of maximum lifetime routing in ad hoc network in the presence of selfish nodes. We designed two mechanisms, SMM for GMM algorithm and SMM-VCG for the CMM algorithm, by applying the framework of algorithm mechanism design. The basic idea of our mechanisms is to give appropriate payment to stimulate the cooperation of nodes, and to ensure that cheating can not increase (or may even decrease) the utility. In SMM mechanism, the payment to nodes in the output path is dependent on the path which has the second maximum lifetime in all possible paths. In SMM-VCG mechanism, we integrated the idea of SMM mechanism with Ad hoc-VCG. We proved that these two mechanisms are truthful, and proposed a routing algorithm Ad hoc-SMM to implement the mechanisms. Finally, we conducted experiments to evaluate the payment ratio performance of our mechanisms. It is shown that the payment ratios of mechanisms in different environments are relatively small and stable due to the nature of lifetime-aware routing algorithms.

We believe that Ad hoc-SMM is a first step in the design of a maximum lifetime routing protocol that achieves truthfulness, and it leaves open for better improvement. Ad hoc-SMM works based on the assumption that nodes do not collaborate to cheat. How to deal with cheating nodes coalitions will be investigated in our future work. Reference [19] provided us a clue to design a collusion-resistant routing protocol. Reference [19] assumes that colluding nodes do not fully trust each other unconditionally. It provided a method to make it impossible for each node to convince other nodes that it has taken the actions required by collusion. Consequently, other nodes are not willing to transfer profit to this node in fear that this node may be cheating them. There are two basic approaches that a node can convince other nodes about its own action: showing private type information it has sent to source node or showing payment messages it has received from source node. To prevent node showing private type information to other nodes, the message sent to and recognized by source node is digitally signed. To prevent the payment messages from be decrypted by colluding nodes, Ref.[19] developed a cryptographic technique which signs payment with restricted verifier signature and the signature can be verified only by player node and central bank. We will continue our work to deal with cheating nodes coalitions based on Ref.[19] and try to release the assumption that colluding nodes do not fully trust each other unconditionally.

References:

[1] Anderegg L, Eidenbenz S. Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In: Proc. of the 9th Annual Int’l Conf. on Mobile Computing and Networking. 2003. 245−259.

[2] Buttyan L, Hubaux JP. Stimulating cooperation in self-organizing mobile ad hoc networks. Mobile Networks and Applications, 2003,8:579−592.

[3] Clarke EH. Multipart pricing of public goods. Public Choice, 1971,11:17−33.

[4] Feigenbaum J, Papadimitriou C, Sami R, Shenker S. A BGP-based mechanism for lowest-cost routing. In: Proc. of the 21st Annual Symp. on Principles of Distributed Computing. 2002. 173−182.

[5] Grosu D, Chronopoulos AT. Algorithmic mechanism design for load balancing in distributed systems. In: Proc. of the IEEE Int’l Conf. on Cluster Computing. 2002. 445−450.

[6] Groves T. Incentives in teams. Econometrica, 1973,41:617−631.

[7] Johnson DB, Maltz DA. Dynamic source routing in ad hoc wireless networks. In: Imelinsky T, Korth H, eds. Proc. of the Mobile Computing. Kluwer Academic Publishers, 1996. 153−181.

[8] Karp R, Koutsoupias E, Papadimitriou C, Shenker S. Optimization problems in congestion control. In: Proc. of the 41st IEEE Annual Symp. on Foundations of Computer Science. 2000. 66−74.

[9] Mas-Colell A, Whinston M, Green JR. Microeconomic Theory. New York: Oxford University Press, 1995.

[10] Misra A, Banerjee S. MRPC: Maximizing network lifetime for reliable routing in wireless environments. In: Proc. of the IEEE Wireless Communications and Networking Conf. 2002. 800−806.

[11] Maleki M, Dantu K, Pedram M. Lifetime prediction routing in mobile ad hoc networks. In: Proc. of the IEEE Wireless Communications and Networking Conf. 2003. 1185−1190.

[12] Nisan N, Ronen A. Algorithmic mechanism design. In: Proc. of the 31st Annual ACM Symp. on Theory of Computing. 1999. 129−140.

[13] Royer EM, Toh CK. A review of current routing protocols for ad-hoc mobile wireless networks. IEEE Personal Communications, 1999,6(2):46−55.

[14] Toh CK. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. IEEE Communications Magazine, 2001,39(6):138−147.

[15] Vickrey W. Counterspeculation, auctions, and competitive sealed tenders. Journal of Finance, 1961,16:8−37.

[16] Wang W, Li XY, Wang Y. Truthful multicast routing in selfish wireless networks. In: Proc. of the 10th Annual Int’l Conf. on Mobile Computing and Networking. 2004. 245−259.

[17] Zhong S, Chen J, Yang YR. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: Proc. of the IEEE INFOCOM. 2003. 1987−1997.

[18] Zhong S, Li L, Liu YG, Yang YR. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: An integrated approach using game theoretical and cryptographic techniques. In: Proc. of the 11th Annual Int’l Conf. on Mobile Computing and Networking. 2005. 117−131.

[19] Zhong S, Wu F. On designing collusion-resistant routing schemes for non-cooperative wireless ad hoc networks. In: Proc. of the 13th Annual ACM Int’l Conf. on Mobile Computing and Networking. 2007. 278−289.

XIE Zhi-Peng was born in 1976. He is an associate processor at the School of Computer Science, Fudan University. His research areas are machine learning, formal concept analysis and wireless ad-hoc networks.

ZHANG Qing was born in 1979. After receiving a Ph.D. degree from Fudan University in 2004, he has been working as a senior software engineer at Samsung Electronics. His research areas are wireless ad hoc network and flash memory management.