

# 基于机制设计理论的自组网节点合作协议

郭建立 吴智博 董 剑 杨孝宗 刘宏伟

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

**摘 要** 在移动自组网中,如何激励自私节点,使之参与网络合作,是当前研究的热点问题.文中提出了 LMOCP 协议,采用 VCG 机制计算转发节点的支付.通过引入邻居发现机制,并改进路由发现过程,使网络中控制消息的数量从  $O(n^3)$  减少到  $O(n^2)$ . 对协议的正确性进行了分析,并证明了 LMOCP 协议是事后纳什可实施的,在所有节点都是理性的这一共同知识的假定下,每个节点的最优策略是诚实地报告自己的转发价格.通过 ns-2 对协议进行了验证,仿真结果显示,与现有的几种重要协议相比,LMOCP 协议具有更低的消息负载和网络延迟.

**关键词** 移动自组网;自私节点;合作;VCG 机制;邻居发现

**中图法分类号** TP393 **DOI 号**: 10.3724/SP.J.1016.2009.00483

## A Cooperation Protocol for Ad Hoc Networks with Selfish Nodes Based on Mechanism Design

GUO Jian-Li WU Zhi-Bo DONG Jian YANG Xiao-Zong LIU Hong-Wei

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

**Abstract** In mobile ad hoc networks, how to stimulate selfish nodes to participate in the network cooperation is currently a hot research field. This paper proposes the LMOCP protocol, which applied VCG mechanism to calculate payments for relay nodes. The introduction of a neighbor discovery process and the improvement on routing discovery process can together decrease the overhead of control message from  $O(n^3)$  to  $O(n^2)$ . By encrypting the RREQ, relay nodes can not discard RREQ any more. The validity of the protocol is analyzed, and the LMOCP protocol is proven to be Ex Post Nash implementable under the assumption that node rational is the common knowledge, and the optimal strategy of each node is to honestly report their prices. Ns-2 was used for simulation and the result shows that LMOCP had lower message overhead and shorter network delay compared with several important protocol.

**Keywords** mobile ad hoc networks; selfish node; node cooperation; VCG mechanism; neighbor discovery

## 1 引 言

在早期的移动自组网<sup>[1]</sup>中,所有节点都属于同

一组织,具有同一目的,这些节点间的相互合作是自然而然的.目前移动自组网中成熟的路由协议,如 DSR 和 AODV,也都假设网络中的节点是合作的,它们都乐意为其它节点提供数据转发服务.

收稿日期:2008-07-06;最终修改稿收到日期:2009-01-06. 本课题得到国家“八六三”高技术研究发展计划项目基金(2006AA01A103)、国家自然科学基金(60503015)资助. 郭建立,男,1980 年生,博士研究生,主要研究方向为移动自组网、移动计算. E-mail: gjl@ftcl.hit.edu.cn. 吴智博,男,1954 年生,教授,博士生导师,主要研究领域为计算机体系结构、移动存储、移动计算. 董 剑,男,1978 年生,博士,讲师,主要研究方向为容错计算、移动计算. 杨孝宗,男,1939 年生,教授,博士生导师,主要研究领域为移动计算、容错计算. 刘宏伟,男,1971 年生,博士,副教授,主要研究方向为容错计算、移动计算.

近年来,随着硬件技术的不断进步,民用自组网(如临时在大学教室或飞机场中搭建的自组网)开始大规模出现.在这些网络中,各节点分别隶属于不同的个人或组织,它们缺乏共同的目的,节点间的合作将无法得到保证,某些节点可能会为了节省资源(如能量等)而表现出自私行为,丢弃那些需要转发的数据.

在移动自组网中,自私节点是个严重的问题,即使只有少数节点表现出自私行为,就可能对网络的性能造成很大的影响.文献[2]通过博弈论方法从理论上证明了自组网中不存在自发的节点合作,需要某种外部机制来迫使节点间的合作.文献[3]指出,当网络中存在 10%~40% 的自私节点时,将会导致整个网络的性能下降 16%~32%.

一种解决自私节点问题的方法是为节点转发数据提供补偿,并且使得补偿大于转发数据的代价,以激励节点参与数据转发过程.但是每个节点的转发价格(主要与传输功率和电池剩余电量有关)是不同的,自私节点为了获得超额利润,可能会夸大自己的转发价格,这就要求我们必须使用某种方法迫使节点诚实地报告自己的转发价格.于是,一些研究者开始探索利用博弈论方法分析移动自组网<sup>[2,4]</sup>,并采用机制设计理论<sup>[5-7]</sup>设计补偿方案.

Anderegg<sup>[8]</sup>等最先把机制设计方法应用到移动自组网中,提出了 Ad hoc-VCG 协议,迫使每个节点报告自己真实的转发价格. LOTTO<sup>[9]</sup>是对 Ad hoc-VCG 的改进,其主要贡献是把消息负载从  $O(n^3)$  降低到  $O(n^2)$ . 但是, LOTTO 也引入了新的问题,就是需要由源节点收集网络的拓扑信息,并负责计算最小价格路径(Lowest Cost Path, LCP)和转发节点的支付,这需要假定源节点是可信的,显然不符合实际情况.

本文提出的 LMOCP(a Low Message-Overhead Cooperation Protocol)协议是对 Ad hoc-VCG 和 LOTTO 的进一步改进,对其支付方案进行了修改,使源节点失去了作弊动机.通过引入邻居发现机制,并改进路由发现过程,进一步减少了网络中控制消息的数量.还对 LMOCP 协议的正确性进行了分析,并证明了 LMOCP 协议是事后纳什可实施的,在所有节点都是理性的这一共同知识的假定下,每个节点的最优策略是诚实地报告自己的转发价格.最后,通过 ns-2 仿真实验进一步对协议性能进行了验证,并对 Ad hoc-VCG 和 LOTTO 进行对比分析.

本文第 2 节介绍相关工作;第 3 节简要介绍机

制设计方面的基本知识;第 4 节指出 Ad hoc-VCG 协议中存在的几个问题;第 5 节详细介绍 LMOCP 协议;第 6 节进行理论分析;第 7 节是仿真实验;最后第 8 节给出全文总结.

## 2 相关工作

针对移动自组网中自私节点问题,出现了许多解决方法<sup>[10-11]</sup>,大体上可以分为两类:基于信誉(reputation-based)的方法<sup>[3,12-13]</sup>和基于虚拟货币(credit-based)的方法<sup>[8-9,14-17]</sup>.

基于信誉的方法,主要是通过对节点建立信誉值<sup>[18]</sup>,并对不合作节点进行惩罚,来迫使节点进行合作. Marti 等<sup>[3]</sup>最先采用看门狗(watchdog)技术检测网络中的自私节点. Buchegger<sup>[12]</sup>进一步提出采用二手信息计算节点的信誉值.文献[13]对信誉值计算过程中所存在的问题进行了讨论.

基于虚拟货币的方法,主要是通过为转发节点提供一定数量的虚拟货币作为转发数据的补偿,来激励节点进行合作. Anderegg 等<sup>[8]</sup>提出了 Ad hoc-VCG 协议,在其中使用机制设计原理,把数据转发过程看作一个二级密封价格拍卖活动. Zhong<sup>[14]</sup>等指出 Ad hoc-VCG 协议不是诚实的,并给出了一个解决方法. Wang 等<sup>[9]</sup>提出的 LOTTO 方法,把路由发现过程中的消息负载从  $O(n^3)$  降低到  $O(n^2)$ . Eidenbenz 等<sup>[15]</sup>使用拓扑控制方法确定节点的传输功率,并对 Ad hoc-VCG 的支付方案进行修改,使源节点满足个人理性约束. Cai 等<sup>[16]</sup>提出 TERM 方法,在自由空间传播模型的假设下,利用递归拍卖找出最小传输功率路径. Dorsey<sup>[17]</sup>提出 XPM 方法,把路由选择看作一个组合交换过程,利用博弈论方法寻找最优路径.

Nisan 在其著名文章<sup>[19]</sup>中首次提出了算法机制设计(algorithmic mechanism design),正式把机制设计理论引入到计算机科学中. Feigenbaum 等<sup>[20]</sup>提出了分布式算法机制设计,并给出一个简单的分布式 VCG 算法<sup>[21]</sup>.目前这方面的工作还仅限于理论方面的研究<sup>[22-25]</sup>,主要集中在 VCG 机制方面,针对一个高度抽象化的模型——工作调度模型.

## 3 机制设计预备知识

本节仅给出机制设计中的基本知识,详细内容请参阅文献[5-7,22].

**定义 1 (机制).** 一个机制  $M = (\Sigma_1, \dots, \Sigma_N, g(\cdot))$  规定了每个代理人  $i$  的策略集  $\Sigma_i$  以及一个结果规则(outcome rule)  $g: \Sigma_1 \times \dots \times \Sigma_N \rightarrow O$ , 使得对于每个策略组合  $s = (s_1, \dots, s_N) \in \Sigma_1 \times \dots \times \Sigma_N$ ,  $g(s)$  是由机制  $M$  所实现的结果(outcome).

对于代理人的每个策略组合, 机制  $M$  都要根据结果规则产生一个输出  $o$ , 每个代理人  $i$  对输出  $o$  有一个效益函数(utility function)  $u_i(o, \theta_i)$ , 其中  $\theta_i$  是代理人  $i$  的类型.

**定义 2 (机制实现).** 对于机制  $M = (\Sigma_1, \dots, \Sigma_N, g(\cdot))$ , 如果对  $\forall (\theta_1, \dots, \theta_N) \in \Theta_1 \times \dots \times \Theta_N$ , 有  $g(s_1^*(\theta_1), \dots, s_N^*(\theta_N)) = f(\theta)$ , 则我们说机制  $M$  实现了社会选择函数(social choice function)  $f(\theta)$ , 其中  $(s_1^*, \dots, s_N^*)$  是由机制  $M$  导出的博弈的均衡解.

**定义 3 (占优解).** 对任意类型组合  $\theta$ , 无论其它代理人选择何种策略, 代理人  $i$  总有一个最优反应策略  $s_i^*$ , 即对  $\forall \theta_i \in \Theta_i$  有

$$u_i(g(s_i^*(\theta_i), s_{-i}(\theta_{-i})), \theta_i) > u_i(g(s_i(\theta_i), s_{-i}(\theta_{-i})), \theta_i), \quad \forall s_{-i}, \forall \theta_{-i}.$$

**定义 4 (事后纳什均衡解).** 对任意类型组合  $\theta$ , 只要其它代理人都选择均衡策略, 则代理人  $i$  总有一个最优反应策略  $s_i^*$ , 即对  $\forall \theta_i \in \Theta_i$  有

$$u_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i) > u_i(g(s_i(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i), \quad \forall \theta_{-i}.$$

**定义 5 (诚实机制).** 如果机制  $M$  满足以下两个条件, 则称其为诚实机制:

- (1) 每个代理人的策略是报告自己的类型;
- (2) 真实地报告自己的类型是占优策略.

## 4 Ad Hoc-VCG 协议存在的问题分析

Ad hoc-VCG 通过功率控制方法节能, 利用式(1)计算两个相邻节点间的最小传输功率:

$$P_{i,j}^{\min} = \frac{P_i^{\text{emit}}}{P_{i,j}^{\text{rec}}} \times P_{i,j}^{\text{rec}_{\min}} \quad (1)$$

其中,  $P_i^{\text{emit}}$  为节点  $i$  的发射功率,  $P_{i,j}^{\text{rec}}$  为节点  $j$  的接收功率,  $P_{i,j}^{\text{rec}_{\min}}$  为最小接收功率. 利用 VCG 机制计算转发节点的支付, 对于转发节点  $v_i$ , 其支付为

$$\text{Pay}(v_i) = \text{Cost}(LCP^{-i}) - \text{Cost}(LCP) + c_i \times P_{i,i+1}^{\min} \quad (2)$$

其中  $\text{Cost}(LCP)$  是 LCP 的总价格,  $LCP^{-i}$  是节点  $v_i$  的替代路径,  $\text{Cost}(LCP^{-i})$  是  $LCP^{-i}$  的总价格,  $c_i$  是节点  $v_i$  的能量价格系数(cost-of-energy).

**问题 1.** Ad hoc-VCG 不是诚实机制.

文献[8-9]中声称 Ad hoc-VCG 是诚实机制, 而在其证明过程中仅得出纳什均衡解, 根据定义 5, 诚实机制要求节点真实报告自己的类型是占优策略, 显然其证明是错误的.

这里通过一个反例进行说明. 如图 1 所示, 假定所有节点的能量价格系数都为 1,  $P_{S,2}^{\min} = P_{3,D}^{\min} = 4$ ,  $P_{S,1}^{\min} = P_{1,D}^{\min} = 6$ , 当  $v_2$  以  $P_2^{\text{emit}} = 6$  发送数据时, 节点  $v_3$  的接收功率  $P_{2,3}^{\text{rec}} = 6$ , 最小接收功率  $P_{\min}^{\text{rec}} = 1$ , 根据式(1), 节点  $v_2$  的最小传输功率  $P_{2,3}^{\min} = 1$ . 如果在计算  $P_{2,3}^{\min}$  时, 节点  $v_3$  作弊, 使  $P_{2,3}^{\text{rec}}$  减小到 1, 则  $P_{2,3}^{\min}$  将增大到 6, 这时路径  $\langle v_S, v_2, v_3, v_D \rangle$  不再是 LCP, 节点  $v_2$  的支付为 0; 如果此时  $v_2$  也作弊, 以  $P_2^{\text{emit}} = 6$  发送消息, 但却声明  $P_2^{\text{emit}} = 2$ , 则  $P_{2,3}^{\min}$  将变为 2, 这时路径  $\langle v_S, v_2, v_3, v_D \rangle$  是 LCP, 并且当节点  $v_3$  以  $P_{2,3}^{\min} = 2$  传输数据时,  $v_3$  能够收到来自  $v_2$  的数据, 根据式(2)可以计算出节点  $v_2$  的支付为 4, 扣除  $v_2$  转发数据的代价 2,  $v_2$  依然能够获得 2 个单位的收益, 可见节点  $v_2$  真实地报告自己的价格并不是占优策略, 因此 Ad hoc-VCG 不是诚实机制.

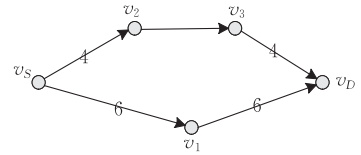


图 1 Ad hoc-VCG 不是诚实机制的一个例子

**问题 2.** Ad hoc-VCG 中源节点有作弊动机.

同样通过一个反例进行说明. 如图 2 所示, 源节点  $v_S$  与目的节点  $v_D$  之间的 LCP 为  $\langle v_S, v_1, v_2, v_3, v_4, v_5, v_D \rangle$ ,  $\text{Cost}(LCP) = 6$ , 对于转发节点  $v_1$ , 其替代路径  $LCP^{-v_1}$  为  $\langle v_S, v_6, v_7, v_D \rangle$ ,  $\text{Cost}(LCP^{-v_1}) = 3 + 3 + 3 = 9$ , 由式(2)知, 节点  $v_1$  的支付为

$$\text{Pay}(v_1) = \text{Cost}(LCP^{-v_1}) - \text{Cost}(LCP) + c_1 \times P_{1,2}^{\min} = 9 - 6 + 1 = 4.$$

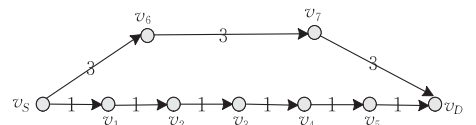


图 2 源节点通过作弊增加自己的支付的一个例子

同样的, 可以计算出

$$\text{Pay}(v_2) = \text{Pay}(v_3) = \text{Pay}(v_4) = \text{Pay}(v_5) = 4.$$

因此, 源节点的支付为

$$\text{Pay}(v_S) = -(\text{Pay}(v_1) + \text{Pay}(v_2) + \text{Pay}(v_3) + \text{Pay}(v_4) + \text{Pay}(v_5)) = -20.$$

如果源节点  $v_s$  减小边  $v_s \rightarrow v_6$  的价格,使其从 3 减小到 1,则 LCP 不变,但每个转发节点的替代路径的价格变小了(从 9 减小到 7).根据式(2),每个转发节点的支付将从 4 减小到 2,从而源节点的支付将从 -20 增大到 -10,显然,源节点  $v_s$  可以通过作弊增加自己的支付.

**问题 3.** 路由发现过程中节点可能丢弃 RREQ.

**定理 1.** 在路由发现过程中,当节点  $v_i$  收到一个 RREQ,并且其中含有从源节点  $v_s$  到节点  $v_i$  的路径  $Path_{s,i}$ ,其总价格记为  $Cost(Path_{s,i})$ ;之后,如果节点  $v_i$  再次收到 RREQ,其中含有路径  $Path'_{s,i}$ ,并且  $Cost(Path'_{s,i}) > Cost(Path_{s,i})$ ,则丢弃该 RREQ 并不会影响节点  $v_i$  的支付.

证明. 见附录.

如图 3 所示,对于节点  $v_5$ ,如果其首先从节点  $v_4$  收到含有  $Path_{s,5} = \langle v_s, v_1, v_2, v_3, v_4, v_5 \rangle$  的 RREQ,节点  $v_5$  会按照协议规定继续广播该 RREQ.之后,当节点  $v_5$  收到来自  $v_7$  的含有  $Path'_{s,5} = \langle v_s, v_1, v_2, v_3, v_7, v_5 \rangle$  的 RREQ 时,丢弃该 RREQ 并不会影响自己的支付.但是,  $Path'_{s,5}$  中含有的信息对节点  $v_4$  却很重要,目的节点  $v_D$  需要利用  $Path'_{s,5}$  计算节点  $v_4$  的支付.

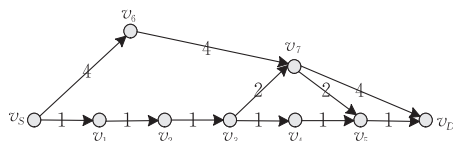


图 3 路由发现过程中节点丢弃 RREQ 的一个例子

**问题 4.** Ad hoc-VCG 中消息负载(message overhead)过重.

在 Ad hoc-VCG 中,一次路由发现过程总共需要转发  $O(n^3)$  个 RREQ.当网络中节点数量较多时,消息负载会很重,将严重影响网络性能.在对 80 个节点所进行的仿真实验中,Ad hoc-VCG 在一次路由发现过程中总共需要转发三万多个 RREQ,过重的消息负载还增大了路由发现过程的延迟,使延迟长达四五十秒钟.

## 5 LMOCP 协议

### 5.1 系统模型

本文把移动自组网看作一个有向边加权图  $G = (V, E, w)$ ,其中  $V = \{v_1, \dots, v_n\}$  是节点集合,  $E \subseteq V \times V$  是连接节点的有向边集合,如果节点  $v_i$  能够向  $v_j$  直接发送数据,则  $e_{i,j} \in E$ ,加权函数  $w: E \rightarrow \mathbb{R}$  把

有向边集合映射到实数集,对每条有向边  $e_{i,j}$ ,  $w(e_{i,j})$  表示节点  $v_i$  向节点  $v_j$  传送一个数据包的价格.另外,还假定图  $G$  是顶点二连通的,即去掉图  $G$  中任意一个顶点以及与其相关的边,图  $G$  依然是连通的.

采用式(1)计算节点的最小传输功率,并假定:在路由发现过程中,节点  $v_i$  使用全向天线以最大功率  $P_{\max}^{\text{emit}}$  发送控制消息;在数据传输过程中,为了节省能量,节点  $v_i$  使用最小传输功率  $P_{i,j}^{\min}$  发送数据.每个节点  $v_i$  有一个参数  $c_i$ ,表示节点  $v_i$  的能量价格系数,  $c_i$  的值依赖于节点  $v_i$  当前电池的电量以及对电池充电的难易程度.有向边  $e_{i,j}$  的加权值  $w(e_{i,j})$  等于  $P_{i,j}^{\min}$  和  $c_i$  的乘积,即  $w(e_{i,j}) = P_{i,j}^{\min} \times c_i$ .

本文还假定网络中节点是自私和理性(rational)的,每个节点都努力使自己的效用最大化,并且自私节点不会恶意破坏网络或者攻击其它节点,节点间也不存在共谋.

### 5.2 协议描述

在应用 VCG 机制时,不仅要发现从源节点到目的节点的 LCP,还要计算出 LCP 上各转发节点的替代路径.为此,目的节点需要知道整个网络的拓扑结构.这主要通过路由发现过程来完成,每个 RREQ 中包含一部分网络拓扑信息,目的节点从 RREQ 中获取网络拓扑信息,以重建整个网络的拓扑结构.为了使目的节点能够收集到完整的网络拓扑信息,就需要传送大量的 RREQ,而我们却期望传送尽量少的 RREQ.

一种可行的减少 RREQ 数量的方法是,让每个 RREQ 包含更多的信息,并且不同的 RREQ 中包含的信息尽量不同(内容互异).这样,只需要少量的 RREQ,目的节点就能够收集到完整的网络拓扑信息.

#### 5.2.1 邻居发现过程

在 LMOCP 中,实现了一个简单的邻居发现过程.每个节点除了发现它的所有邻居外,还要计算每个邻居到自己的最小传输功率.具体过程如下:对任意节点  $v_i$ ,首先把自己的最大传输功率  $P_i^{\max}$  放入消息中,并周期地以功率  $P_i^{\max}$  广播该消息;对于  $v_i$  的邻居  $v_j$ ,当其收到来自节点  $v_i$  的广播消息后,首先把节点  $v_i$  放入自己的邻居集合  $NB_j$  中,然后根据式(1)计算从节点  $v_i$  到自己的最小传输功率  $P_{i,j}^{\min}$ ,并把  $P_{i,j}^{\min}$  放入缓存中.

在一次邻居发现过程后,每个节点  $v_j$  不仅能够

确定自己的邻居集合  $NB_j$ , 而且还知道每个邻居到自己的最小传输功率。

### 5.2.2 路由发现过程

当源节点有数据要发送时, 首先去缓存中查找可用路由, 如果没有命中, 则启动路由发现过程, 发送 RREQ, 其中包含

$\langle D, SeqNo, S, c_S, NeibNum_S, \{(v_i, P_{i,S}^{\min}): v_i \in NB_S\} \rangle$ ,  $S$  和  $D$  分别是源和目的节点的 ID,  $SeqNo$  是 RREQ 的序列号,  $c_S$  是源节点的能量价格系数,  $NeibNum_S$  是源节点的邻居数量,  $\{(v_i, P_{i,S}^{\min}): v_i \in NB_S\}$  是邻居节点到源节点  $S$  的最小传输功率的集合。

对任意转发节点  $v_j \notin \{S, D\}$ , 当其收到一个 RREQ 消息后, 执行以下动作:

1. 检查  $\langle S, D, SeqNo \rangle$ . 如果节点  $v_j$  保存的从  $S$  到  $D$  的序列号为  $SeqNo'$ , 并且  $SeqNo' > SeqNo$ , 则  $v_j$  就认为这是一个旧的请求, 丢弃它。

2. 检查是否已经处理过该 RREQ. 如果在 RREQ 中发现  $\langle v_j, c_j, NeibNum_j, \{(v_i, P_{i,j}^{\min}): v_i \in NB_j\} \rangle$ , 则说明该 RREQ 到达过节点  $v_j$ , 并且  $v_j$  已经处理过它了, 而其又沿着某条回路再次回到节点  $v_j$ , 丢弃该 RREQ, 以减少网络中 RREQ 的数量。

3. 检查 RREQ 中是否包含新的信息. 令  $\langle S, v_{k_1}, \dots, v_{k_m} \rangle$  是 RREQ 路由过的路径, 如果对所有  $1 \leq i \leq m$ ,  $v_{k_i}$  都位于  $v_j$  的缓存中, 则说明该 RREQ 中的信息已经包含在之前转发过的 RREQ 中了, 丢弃它。

4. 把 RREQ 路由过的节点放入  $v_j$  的本地缓存中。

5. 把  $v_j$  和  $c_j$  放入 RREQ 中。

6. 把  $v_j$  的邻居个数  $NeibNum_j$  放入 RREQ 中。

7. 把  $\{(v_i, P_{i,j}^{\min}): v_i \in NB_j\}$  放入 RREQ 中。

8. 继续广播该 RREQ。

目的节点收到的 RREQ 的结构如下:

$\langle D, seqNo, S, c_S, NeibNum_S, \{(v_i, P_{i,S}^{\min}): v_i \in NB_S\}, \dots, v_j, c_j, NeibNum_j, \{(v_i, P_{i,j}^{\min}): v_i \in NB_j\} \rangle$   
目的节点收到 RREQ 后, 从中取出每个转发节点  $v_k$  的信息:

$\langle v_k, c_k, NeibNum_k, \{(v_i, P_{i,k}^{\min}): v_i \in NB_k\} \rangle$

放入本地缓存. 当目的节点收集到网络中全部  $n$  个节点的信息后, 就开始构建网络的拓扑结构图, 然后调用 Dijkstra<sup>[26]</sup> 算法计算从源到目的节点的 LCP, 对于 LCP 上的每个节点  $v_i$ , 调用文献[27-28]中的算法计算替代路径  $LCP^{-i}$ , 并根据 LCP 和替代路径计算源节点和各转发节点的支付. 最后, 源节点和转发节点的支付以及 LCP 被放入 RREP 中, 沿着 LCP 的反方向, 以单播方式返回源节点。

### 5.2.3 路由发现过程优化

对于目的节点  $D$  的邻居  $v_i \in NB_D$ , 在其执行完上述步骤 1~7 后, 修改 RREQ 的最大跳数  $req\_ttl=1$ , 然后才对其进行广播. 当节点  $v_j$  收到  $v_i$  广播的具有  $req\_ttl=1$  的 RREQ 后,  $v_j$  只根据 RREQ 中的信息更新本地缓存, 并不广播该 RREQ, 以减少网络中 RREQ 的数量。

仿真过程中发现, 该优化过程可以减小网络中 RREQ 的数量, 尤其当目的节点的邻居数量较多时, 效果更明显。

### 5.2.4 数据转发过程

当源节点收到 RREP 后, 就开始发送数据, 数据的头部中包含以下信息:

$\langle S, v_1, P_{1,2}^{\min}, Pay(v_1), \dots, v_i, P_{i,i+1}^{\min}, Pay(v_i), \dots, v_k, P_{k,D}^{\min}, Pay(v_k), D \rangle$ ,

其中  $\langle S, v_1, \dots, v_i, \dots, v_k, D \rangle$  是从源节点  $S$  到目的节点  $D$  的 LCP,  $P_{i,i+1}^{\min}$  是从节点  $v_i$  到节点  $v_{i+1}$  的最小传输功率,  $Pay(v_i)$  是节点  $v_i$  的支付。

### 5.3 支付方案

重新定义最小价格路径 LCP 和替代路径。

**定义 6.** 令  $Path = \langle S (= v_0), v_1, \dots, v_{k-1}, D (= v_k) \rangle$  是从源节点  $S$  到目的节点  $D$  的一条路径,  $w(e_{i,i+1}) = P_{i,i+1}^{\min} \times c_i$  是转发节点  $v_i$  的价格, 则 LCP 为转发节点价格之和最小的路径, 即<sup>①</sup>

$$LCP = \arg \min_{Path} \sum_{m=1}^{k-1} w(e_{m,m+1}) \quad (3)$$

(注意: 在计算 LCP 时, 并不包含源节点的价格)

**定义 7.** 对于 LCP 上的转发节点  $v_i$ , 其替代路径  $LCP^{-i}$  为: 在不包含节点  $v_i$  的路径中, 转发节点价格之和最小的路径, 即

$$LCP^{-i} = \arg \min_{Path \wedge v_i \notin Path} \sum_{m=1}^{k-1} w(e_{m,m+1}) \quad (4)$$

令  $LCP = \langle S (= v_0), v_1, \dots, v_{m-1}, D (= v_m) \rangle$  是最小价格路径, 则 LCP 的价格为

$$Cost(LCP) = \sum_{i=1}^{m-1} w(e_{i,i+1}) \quad (5)$$

令  $LCP^{-i} = \langle S (= v_0), v_1, \dots, v_{L-1}, D (= v_L) \rangle$  是节点  $v_i$  的替代路径, 则  $LCP^{-i}$  的价格为

$$Cost(LCP^{-i}) = \sum_{i=1}^{L-1} w(e_{i,i+1}) \quad (6)$$

接下来定义节点的支付. 数据沿着 LCP 从源节点  $S$  路由到目的节点  $D$ , 对于不在 LCP 上的节点,

① 符号  $\arg \min_x f(x)$  表示在变量  $x$  的所有取值中, 使函数  $f(x)$  值最小那个  $x$  的值。



规定其支付为 0; 对于那些位于 LCP 上的转发节点  $v_i$ , 其支付为

$$Pay(v_i) = Cost(LCP^{-i}) - Cost(LCP) + w(e_{i,i+1}) \quad (7)$$

对于源节点  $S$ , 其支付为

$$Pay(S) = -\left(\sum_i Pay(v_i)\right) \quad (8)$$

其中  $v_i$  是 LCP 上的转发节点, 即由源节点  $S$  为 LCP 上的转发节点支付费用。

下面用一个例子说明节点支付的计算过程. 如图 4 所示, 从  $S$  到  $D$  的最小价格路径为  $LCP = \langle S, v_3, v_4, D \rangle$ ,  $Cost(LCP) = w(e_{3,4}) + w(e_{4,D}) = 3 + 2 = 5$ . 节点  $v_3$  的替代路径为  $LCP^{-v_3} = \langle S, v_1, v_2, D \rangle$ ,  $Cost(LCP^{-v_3}) = w(e_{1,2}) + w(e_{2,D}) = 4 + 2 = 6$ , 根据式(7), 节点  $v_3$  的支付为

$$Pay(v_3) = Cost(LCP^{-v_3}) - Cost(LCP) + w(e_{3,4}) = 6 - 5 + 3 = 4.$$

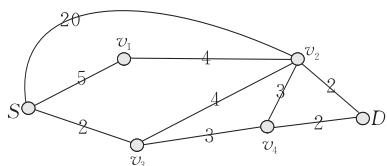


图 4 计算节点支付的一个例子

同样的, 节点  $v_4$  的替代路径为  $LCP^{-v_4} = \langle S, v_3, v_2, D \rangle$ ,  $Cost(LCP^{-v_4}) = 4 + 2 = 6$ , 节点  $v_4$  的支付为

$$Pay(v_4) = Cost(LCP^{-v_4}) - Cost(LCP) + w(e_{4,D}) = 6 - 5 + 2 = 3.$$

根据式(8), 源节点  $S$  的支付为

$$Pay(S) = -(Pay(v_3) + Pay(v_4)) = -(4 + 3) = -7.$$

显然, 无论源节点  $S$  如何报价, 都不能增加自己的支付。

#### 5.4 加密

第 4 节中曾提到, Ad hoc-VCG 中节点可能丢弃待转发的 RREQ. 在 LMOCP 协议中, 采用加密方法来解决这个问题。

假定目的节点  $D$  有一个公钥  $PK_D$  和一个密钥  $SK_D$ , 并且网络中所有节点都知道  $PK_D$  (这个假设是合理的, 如在一个提供接入服务的无线网络中, 运营商的接入点是目的节点, 而购买了服务的用户是源节点或转发节点). 修改算法的第 7 步为

7. 使用公钥  $PK_D$  对信息  $\{(v_i, P_{i,j}^{mm}): v_i \in NB_j\}$  加密, 并把密文放入 RREQ 中。

当转发节点  $v_j$  收到一个 RREQ 后,  $v_j$  只能得知 RREQ 经过了哪些节点, 并不能获得这些节点的信息(如最小传输功率), 因此就无法丢弃 RREQ. 对

于目的节点  $D$ , 当其收到 RREQ 后, 使用密钥  $SK_D$  对密文解密, 从而获得 RREQ 中的信息。

## 6 协议分析

本节中, 首先分析协议的正确性, 然后指出 LMOCP 协议是事后纳什可实施的, 并给出相应的证明, 最后对协议的消息负载进行分析。

### 6.1 协议正确性

协议正确性主要指当路由发现过程结束时, 目的节点  $D$  能够获得足够的网络拓扑信息. 在路由发现过程中, 只有步骤 1~3 用于删除无用的 RREQ. 显然步骤 1 和步骤 3 不会影响协议的正确性, 下面证明步骤 2 也不会影响协议的正确性。

**定理 2.** 令  $Path_{S,D} = \langle S, v_1, \dots, v_i, v_{L_1}, \dots, v_{L_k}, v_i, v_{i+1}, \dots, v_m, D \rangle$  是从源节点  $S$  到目的节点  $D$  的含有回路的一条路径, 当 RREQ 沿着路径  $Path_{S,D}$  第二次经过节点  $v_i$  时, 节点  $v_i$  将其丢掉并不会影响目的节点对 LCP 和替代路径的计算, 即不会影响协议的正确性。

证明。

(1) 因为  $\langle S, v_1, \dots, v_i, v_{i+1}, \dots, v_m, D \rangle$  组成一条不含回路的从  $S$  到  $D$  的路径, 沿着这条路径的 RREQ 不会被丢弃, 因此目的节点能够获得节点  $v_i (1 \leq j \leq m)$  的信息。

(2) 对于回路上的任意节点  $v_{L_j} (1 \leq j \leq k)$ , 如果存在一条从  $S$  到  $D$  的经过  $v_{L_j}$  的简单路径(不含回路), 则目的节点也能够获得节点  $v_{L_j}$  的信息。

(3) 对于回路上的任意节点  $v_{L_j} (1 \leq j \leq k)$ , 如果所有经过该节点的从  $S$  到  $D$  的路径中都含有回路, 则节点  $v_{L_j}$  的信息对计算 LCP 和替代路径没有帮助(因为 LCP 和替代路径都不含有回路, 因此 LCP 和替代路径中都不会含有节点  $v_{L_j}$ ), 丢弃它并不会影响协议的正确性。

证毕。

如图 5 所示, 所有从  $S$  到  $D$  并经过节点  $v_7$  的路径中都包含回路  $\langle v_2, v_7, v_8, v_2 \rangle$ , 而节点  $v_7$  既不在 LCP 上, 也不在替代路径上, 因此丢弃含有回路的 RREQ 并不会影响目的节点  $D$  的计算结果。

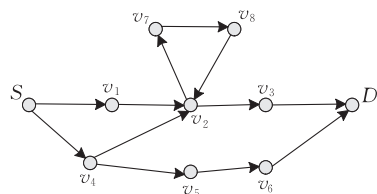


图 5 丢弃含有回路的 RREQ 不影响协议的正确性的例证

## 6.2 协议性质

**定理 3.** LMOCP 协议不是诚实机制。

证明. 可构建一个如图 1 所示的反例. 证毕.

**定理 4.** LMOCP 协议是事后纳什可实施的。

证明. 令  $s_i^* = \text{“诚实计算 } w(e_{j,i}), \text{ 并且诚实计算 } w(e_{i,j})\text{”}$ , 我们需要证明当节点  $v_j (j \in \{1, 2, \dots, N\} \text{ 且 } j \neq i)$  选择策略  $s_j^*$  时, 节点  $v_i$  的最优策略为  $s_i^*$ .

当所有节点都采用均衡策略时, 得到节点的真实价格组合, 记此时的最小价格路径为  $LCP_{\text{true}}$ ,  $v_i$  的替代路径为  $LCP_{\text{true}}^{-i}$ , 含有链路  $e_{i,j}$  的最小价格路径为  $LCP_{\text{true}}^{i,j}$ ,  $e_{i,j}$  的真实价格为  $TC_{i,j}$ .

首先证明, 对于任意节点  $v_i (v_i \notin \{S, D\})$ , 当其它节点都选择均衡策略时,  $v_i$  的最优策略为  $s_i^*$ . 分两种情况讨论.

情况 1. 节点  $v_i$  不在 LCP 上

此时,  $v_i$  采用均衡策略  $s_i^*$  时的支付为 0.

对于链路  $e_{j,i} (v_j \in NB_i)$ , 显然当  $v_i$  采用策略  $s_i = \text{“增大 } w(e_{j,i})\text{”}$  时, 并不能使  $v_i$  位于 LCP 上, 因此  $v_i$  的支付依然为 0; 当  $v_i$  采用策略  $s_i = \text{“减小 } w(e_{j,i})\text{”}$  时, 即便能够使  $v_i$  位于 LCP 上, 但在数据转发过程中,  $v_i$  无法收到  $v_j$  转发的数据, 因此也无法获益.

对于链路  $e_{i,j} (v_j \in NB_i)$ , 显然当  $v_i$  采用策略  $s_i = \text{“增大 } w(e_{i,j})\text{”}$  时,  $v_i$  不会位于 LCP 上, 其支付为 0; 当  $v_i$  采用策略  $s_i = \text{“减小 } w(e_{i,j})\text{”}$  时, 如果  $v_i$  在 LCP 上, 则通过式(7)可知:

$Pay(v_i) = Cost(LCP^{-i'}) - Cost(LCP') + w(e_{i,j})'$ . 显然  $LCP^{-i'} = LCP_{\text{true}}$ ,  $LCP' = LCP_{\text{true}}^{i,j}$ ,  $Cost(LCP') - w(e_{i,j})' + TC_{i,j} = Cost(LCP_{\text{true}}^{i,j})$ , 而  $Cost(LCP_{\text{true}}^{i,j}) > Cost(LCP_{\text{true}})$ , 因此有  $Pay(v_i) < TC_{i,j}$ .

所以,  $s_i^*$  是节点  $v_i$  的最优策略.

情况 2. 节点  $v_i$  位于 LCP 上.

同理可以证明, 此时节点  $v_i$  的最优策略还是  $s_i^*$ .

对于源节点  $S$ , 从式(5)和(6)中可以看出, 源节点并不影响 LCP 和替代路径的计算, 因此  $s_S^*$  也是源节点  $S$  的最优策略. 另外, 本文假定目的节点  $D$  是可信的, 始终采取策略  $s_D^*$ .

总之, 对任意类型组合  $\theta = (\theta_1, \dots, \theta_N)$ ,  $s^* = (s_1^*, s_2^*, \dots, s_N^*)$  都是纳什均衡策略, 即 LMOCP 是事后纳什可实施的. 证毕.

在事后纳什可实施机制中, 即使某一节点能够预先知道网络的拓扑结构, 在节点都是理性这一共同知识的假设下, 该节点依然会选用均衡策略, 作弊并不会为其带来任何好处, 即节点事后不会后悔. 事

后纳什可实施机制与诚实机制的主要区别是: 事后纳什可实施机制要求节点都是理性的这一共同知识的假设, 而诚实机制则不需要这个假设.

## 6.3 消息负载分析

在 LMOCP 协议中, 共有两个地方需要发送 (或转发) 控制消息: 邻居发现过程和路由发现过程. 在一次邻居发现过程中, 每个节点发送一个广播消息, 网络中  $n$  个节点共发送  $O(n)$  个消息. 在路由发现过程中, 节点每收到一个 RREQ 消息, 只要其中含有新的信息 (即新的节点), 就需要对其转发, 最坏情况下, 每个节点需要转发  $n-1$  个消息, 网络中  $n$  个节点最多需要转发  $O(n^2)$  个消息. 因此, LMOCP 协议总共需要发送  $O(n) + O(n^2) = O(n^2)$  个消息.

## 7 仿真实验及分析

本节采用 ns-2 对 LMOCP 协议进行验证, 并观察其对网络性能的影响, 同时对 Ad hoc-VCG<sup>[8]</sup> 和 LOTTO<sup>[9]</sup> 进行对比分析.

采用以下两个指标对协议性能进行评价:

(1) 消息负载. 在一次路由发现过程中, 网络中所有节点发送或转发的控制消息的总数.

(2) 延迟. 从源节点启动路由发现过程开始, 到发现可用路由为止所花费的时间.

首先观察网络规模对协议性能的影响. 固定仿真区域为  $600 \times 600$ , 改变网络中节点的个数, 从 20 增加到 80, 仿真结果如图 6 所示. 从图 6(a) 中可以看出, 随着节点数量的增多, 3 种协议所对应的消息负载都在增大, Ad hoc-VCG 协议的消息负载增加最快, LOTTO 和 LMOCP 协议的消息负载增加较慢. 从图 6(b) 中可以看出, 随着节点数量的增多, Ad hoc-VCG 协议的延迟也迅速增大, 从 2.9s 增加到 40.8s, 而 LOTTO 和 LMOCP 协议的延迟增加较缓慢, 尤其是 LMOCP 协议, 其延迟始终低于 3s.

为了更加清晰地对比 LOTTO 和 LMOCP 协议, 我们去掉图 6 中 Ad hoc-VCG 协议对应的曲线, 重画为图 7. 从图 7 中可以看出, LMOCP 协议的消息负载和延迟明显低于 LOTTO 协议.

接下来观察节点密度对协议性能的影响. 实验中, 30 个节点随机地分布在仿真区域内, 不断增大仿真区域的面积 (从  $600 \times 600$  增大到  $1000 \times 1000$ ), 仿真结果如图 8 所示 (为了更好地比较 LOTTO 和 LMOCP, 去掉了 Ad hoc-VCG 协议对应的曲线).

从图 8 中可以看出,节点密度对消息负载和延迟的影响很小,随着节点密度变小,消息负载和延迟都没有发生明显的变化.另外,从图 8(b)中还可以看出,LMOCP 的延迟要明显小于 LOTTO,主要有两个原因:(1) LMOCP 的消息负载要比 LOTTO

小;(2) 在 LOTTO 中,当节点收到 RREQ 消息后,并不会立即向源节点返回拓扑信息,而需要等待一小段时间(实验中等待 0.5s),用于计算所有邻居到自己的最小传输功率.

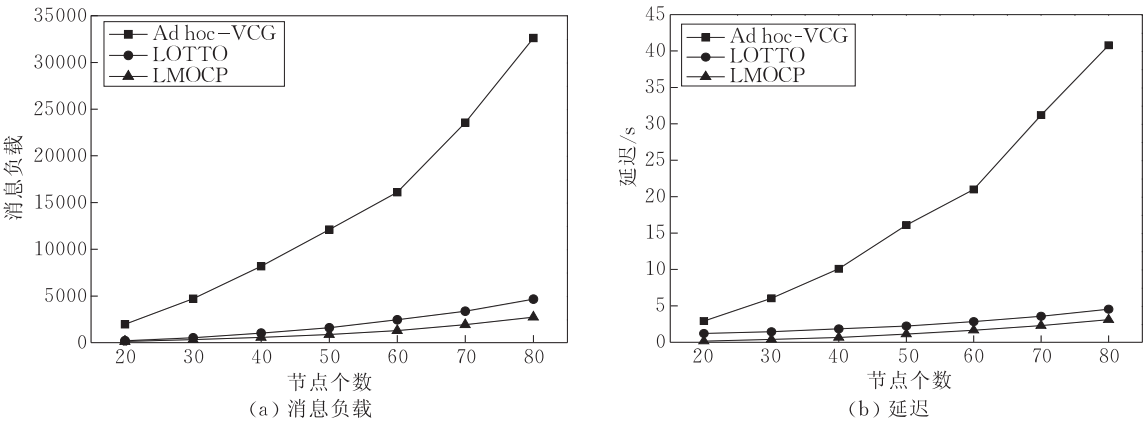


图 6 网络规模对性能的影响(1)

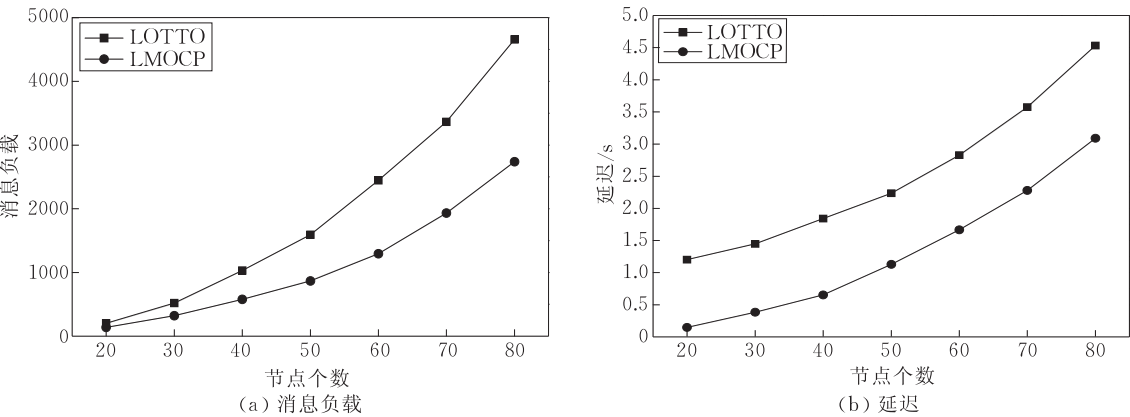


图 7 网络规模对性能的影响(2)

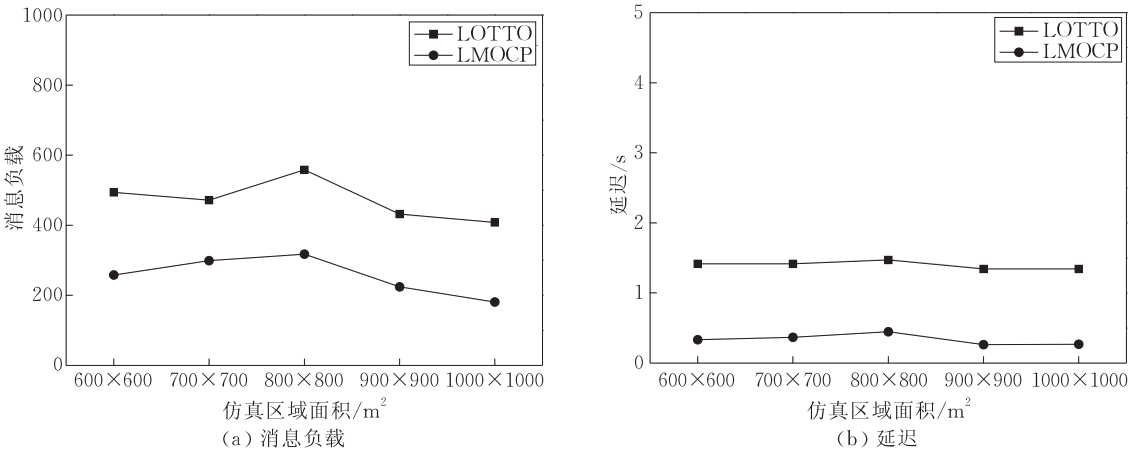


图 8 节点密度对性能的影响

8 结 论

本文详细介绍了 LMOCP 协议,这是一种应用

于移动自组网的节点合作协议,是对 Ad hoc-VCG 和 LOTTO 协议的一个改进,具有更低的消息负载. LMOCP 引入了邻居发现过程,节点周期地以最大功率发送广播消息,每个节点计算出所有邻居到



自己的最小传输功率;改进了路由发现过程,进一步减少了网络中控制消息的数量. 本文还对协议的正确性进行了分析,证明了 LMOCF 协议是事后纳什可实施的,在所有节点都是理性的这一共同知识的假定下,每个节点的最优策略是诚实地报告自己的转发价格. 最后,通过 ns-2 仿真实验对协议性能进行了验证,并对 Ad hoc-VCG 和 LOTTO 进行了比较分析.

## 参 考 文 献

- [1] Chlamtac I, Conti M, Liu J. Mobile Ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, 2003, 1(1): 13-64
- [2] Felegyhazi M, Hubaux J P, Buttyan L. Nash equilibria of packet forwarding strategies in wireless Ad hoc networks. *IEEE Transactions on Mobile Computing*, 2006, 5(5): 463-476
- [3] Marti S, Giulì T, Lai K, Baker M. Mitigating routing misbehavior in mobile Ad hoc networks//*Proceedings of the MOBICOM*. Boston, 2000: 255-265
- [4] Srivastava V, Neel J. Using game theory to analyze wireless Ad hoc networks. *IEEE Communications Surveys & Tutorials*, 2005, 4(7): 46-56
- [5] Andreu M C, Michael D W, Jerry R G. *Microeconomic Theory*. New York: Oxford University Press, 1995
- [6] Nisan N, Roughgarden T, Tardos E, Vazirani V V. *Algorithmic Game Theory*. New York: Cambridge University Press, 2007
- [7] Parkes D. Classic mechanism design. Chapter 2, Iterative combinatorial auctions; Achieving economic and computational efficiency[Ph. D. dissertation]. University of Pennsylvania, Pennsylvania, 2001
- [8] Anderegg L, Eidenbenz S. Ad-hoc-VCG: A truthful and cost efficient routing protocol for mobile ad-hoc networks with selfish agents//*Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*. San Diego, CA, USA, 2003: 245-259
- [9] Wang Y W, Singhal M. On improving the efficiency of truthful routing in MANETs with selfish nodes. *Pervasive and Mobile Computing*, 2007, 3(5): 537-559
- [10] Marias G F, Georgiadis P, Flitzanis D, Mandalas K. Cooperation enforcement schemes for MANETs: A survey. *Wireless Communications & Mobile Computing*, 2006, 6(3): 319-332
- [11] Yoo Y, Agrawal D P. Why does it pay to be selfish in a MANET?. *IEEE Wireless Communications*, 2006, 13(6): 87-97
- [12] Buchegger S, Boudec J-Y L. Self-policing mobile ad-hoc networks by reputation systems. *IEEE Communications Magazine*, 2005, 43(7): 101-107
- [13] Laniecep S, Demerjian J, Mokhtari A. Cooperation monitoring issues in Ad hoc networks//*Proceedings of the International Conference on Wireless Communications and Mobile Computing*. Vancouver, British Columbia, Canada, 2006: 695-700
- [14] Zhong S, Li E L, Liu Y G, Yang Y R. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: An integrated approach using game theoretical and cryptographic techniques//*Proceedings of the 11th ACM Annual International Conference on Mobile Computing and Networking*. Cologne, Germany, 2005: 117-131
- [15] Eidenbenz S, Resta G, Santi P. The COMMIT protocol for truthful and cost-efficient routing in Ad hoc networks with selfish nodes. *IEEE Transactions on Mobile Computing*, 2008, 7(1): 19-33
- [16] Cai J, Pooch U. Play alone or together-Truthful and efficient routing in wireless Ad hoc networks with selfish nodes//*Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. Washington, 2004: 457-465
- [17] Dorsey J G. Game-theoretic power management in mobile Ad hoc networks[Ph. D. dissertation]. Carnegie Mellon University Department of Electrical and Computer Engineering, Pittsburgh, Pennsylvania, 2004
- [18] Buchegger S, Mundinger J, Boudec J-Y L. Reputation systems for self-organized networks. *IEEE Technology and Society Magazine*, 2008, 27(1): 41-47
- [19] Nisan N, Ronen A. Algorithmic mechanism design. *Games and Economic Behavior*, 2001, 35(1): 166-196
- [20] Feigenbaum J, Shenker S. Distributed algorithmic mechanism design: Recent results and future directions//*Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*. Pittsburgh, PA, 2002: 1-13
- [21] Feigenbaum J, Papadimitriou C, Sami R. A BGP-based Mechanism for lowest-cost routing. *Distributed Computing*, 2005, 18(1): 61-72
- [22] Dash R K, Jennings N R, Parkes D C. Computational-mechanism design: A call to arms. *IEEE Intelligent Systems*, 2003, 18(6): 40-47
- [23] Feigenbaum J, Sami R, Shenker S. Mechanism design for policy routing. *Distributed Computing*, 2006, 18(4): 293-305
- [24] Archer A, Tardos P E. Frugal path mechanisms. *ACM Transactions on Algorithms*, 2007, 3(1): 1-22
- [25] Karlin A, Kempe D, Tamir T. Beyond VCG: Frugality of truthful mechanisms//*Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. Seattle, WA, USA, 2005: 615-626
- [26] Cormen T H, Leiserson C E. *Introduction to Algorithms*. 2nd Edition. Cambridge, MA: The MIT Press, 2002
- [27] Hershberger J, Suri S. Vickrey prices and shortest paths: What is an edge worth? //*Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*. Las Vegas, 2001: 252-259
- [28] Hershberger J, Suri S, Bhosle A. On the difficulty of some shortest path problems. *ACM Transactions on Algorithms*, 2007, 3(1)

附录. 定理 1 的证明.

**定理 1.** 在路由发现过程中, 当节点  $v_i$  收到一个 RREQ, 并且其中含有从源节点  $v_s$  到节点  $v_i$  的路径  $Path_{s,i}$ , 其总价格记为  $Cost(Path_{s,i})$ ; 之后, 如果节点  $v_i$  再次收到 RREQ, 其中含有路径  $Path'_{s,i}$ , 并且  $Cost(Path'_{s,i}) > Cost(Path_{s,i})$ , 则节点  $v_i$  丢弃该 RREQ 并不影响自己的支付.

证明.

对于转发节点  $v_i$ , 由式 (2) 知, 决定其支付的主要是  $Cost(LCP)$  和  $Cost(LCP^{-i})$ .

首先证明 LCP 决不会含有路径  $Path'_{s,i}$ . 利用反证法, 假定  $Path'_{s,i}$  是 LCP 的一部分, 即  $LCP = Path'_{s,i} + Path_{i,D}$ , 则  $Cost(LCP) = Cost(Path'_{s,i} + Path_{i,D})$

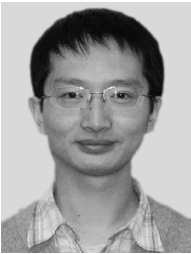
$$\begin{aligned} &= Cost(Path'_{s,i}) + Cost(Path_{i,D}) \\ &> Cost(Path_{s,i}) + Cost(Path_{i,D}) \\ &= Cost(Path_{s,i} + Path_{i,D}) \geq Cost(LCP), \end{aligned}$$

显然假设不成立.

同时,  $LCP^{-i}$  中也不会含有  $Path'_{s,i}$ , 因为  $Path'_{s,i}$  经过了节点  $v_i$ .

因此, RREQ 中包含的信息 ( $Path'_{s,i}$  和  $Cost(Path'_{s,i})$ ) 既不能帮助  $v_i$  成为 LCP 上的节点, 又对  $Cost(LCP)$  和  $Cost(LCP^{-i})$  的计算没有帮助, 从而不会影响节点  $v_i$  的支付.

在移动自组网中, 节点转发数据消耗的能量要远大于计算过程中所消耗的能量, 而模型中又假定节点是理性的, 因此转发节点  $v_i$  总会选择丢弃该 RREQ. 证毕.



**GUO Jian-Li**, born in 1980, Ph. D. candidate. His current research interests include ad hoc networks, mobile computing, etc.

**WU Zhi-Bo**, born in 1954, Ph. D., professor, Ph. D. supervisor. His current research interests include computer architecture, fault tolerance computing and mobile computing, etc.

Background

In ad hoc network, nodes cooperate by forwarding packets for each other to allow them to communicate beyond direct wireless transmission range. All the mature routing protocols, such as DSR and AODV, assume that nodes are cooperative, and they are willing to forward packets for other nodes. In recent years, with the progress of technology, it is becoming possible to deploy mobile ad hoc networks for civilian applications as well. In these networks, each node belongs to different individuals or organizations, and the cooperation among the nodes would not be guaranteed. In order to save resources (such as energy), some nodes may show selfish behavior, discarding all passing by packets.

How to stimulate selfish nodes to cooperate became a hot spot in ad hoc network research community recently.

ting, etc.

**DONG Jian**, born in 1978, Ph. D., associate researcher. His current research interests include fault tolerance computing, mobile computing, etc.

**YANG Xiao-Zong**, born in 1939, professor, Ph. D. supervisor. His current research interests include mobile computing, fault tolerance computing, etc.

**LIU Hong-Wei**, born in 1971, Ph. D., associate professor. His current research interests include fault tolerance computing, mobile computing, etc.

Current studies proposed a lot of schemes to encourage selfish nodes to take part in network cooperation, such as Ad hoc-VCG or LOTTO. But there still exist some drawbacks in these protocols: source node can increase its payment by cheating; relay nodes may drop RREQs, which is important for destination node; there are lots of RREQs need to be exchanged in route discovery phase. This paper proposes a new cooperation protocol named LMOCP, which solves those shortcomings. The authors also prove that LMOCP is Ex Post Nash implemental.

This research is partially supported by the High Technology Research and Development Program (863) of China under grant No. 2006AA01A103 and the National Natural Science Foundation of China under grant No. 60503015.