

# Ad hoc 网络寻路阶段的合作激励机制研究

黄 蕾 刘立祥

(中国科学院软件研究所综合信息系统技术国家级重点实验室 北京 100080)

**摘 要** 如何激励属于不同利益最大化实体的自私节点合作是当前 Ad hoc 网络研究中的一个热点问题. 现有的自私节点检测和激励机制主要针对数据传输阶段, 不能适应寻路阶段的特点. 文中基于邻居节点中继和生成的路由请求包之间的统计关系, 提出了一种适用于按需路由协议寻路阶段的自私行为检测和惩罚机制, 并利用博弈论工具将其建模为噪声环境下的重复囚徒困境博弈, 对算法激励合作的有效性进行分析. 理论分析和仿真结果显示, 该算法能够有效地惩罚寻路中的自私行为, 促进节点合作.

**关键词** Ad hoc 网络; 路由; 自私检测; 合作激励; 博弈论

**中图法分类号** TP393

## Study on Cooperation Stimulation Mechanism in Route Discovery of Ad hoc Networks

HUANG Lei LIU Li-Xiang

(National Key Laboratory of Integrated Information System Technology, Institute of Software,  
Chinese Academy of Sciences, Beijing 100080)

**Abstract** How to stimulate selfish nodes which belong to different utility-maximizing entities to cooperate is a hot topic in Ad hoc network research community. Current mechanisms proposed so far focus mainly on detecting selfish behavior and stimulating cooperation in data forwarding stage. They are not applicable in route discovery stage. Based on statistics relationship of route request packets relayed and generated by a neighbor node, this paper proposed an algorithm to detect and punish the selfishness in route discovery stage for on-demand routing protocols. The algorithm was modeled with the tool of game theory as the repeated prisoner dilemma in noisy environment, and its effectiveness to stimulate cooperation was analyzed with the model. Theoretic analysis and simulation results showed that our scheme could punish the selfishness in route discovery effectively and thus stimulate nodes to cooperate.

**Keywords** Ad hoc network; routing; selfishness detection; cooperation stimulation; game theory

## 1 引 言

Ad hoc 网络由一组移动或固定的无线节点组成, 信息交流等网络关键任务的实现需要各节点之间的相互协作, 这种合作性也是现有诸多路由协议

设计的一个基本假设前提. 但是当节点属于不同实体时, 其合作性缺乏内在的保证, 理性节点更倾向于采取能够使得自身利益最大化的行动, 而不是完全遵从协议. 由于无线传输需要耗费大量的能量, 因此理性的自私节点会尽量避免为其他节点中继数据, 从而导致网络性能下降, 合作用户利益受损.

Ad hoc 网络中自私节点的激励机制是当前的一个研究热点,提出的解决方案可分为三种类型<sup>[1]</sup>: 基于信用的方法(credit-based method)、基于声誉的方法(reputation-based method)和博弈论方法(game theory method). 基于信用的方法一般建立在虚拟货币机制的基础上,通过精心设计的支付方式,使得节点只有在合作的时候才能使自己的利益最大化<sup>[2-4]</sup>. 这种方法的缺陷在于作为其基础的虚拟货币管理系统需要篡改改硬件的支持<sup>[2]</sup>,或者需要集中的支付服务<sup>[4]</sup>,尚未有令人满意的解决方案;基于声誉的方法记录节点的过往行为,综合直接观察结果和第三方信息形成对节点合作性的判断,对不合作的自私节点以拒绝服务的方式进行惩罚,从而达到促进合作的目的<sup>[5-8]</sup>. 目前声誉系统采用基于 Watchdog<sup>[8]</sup>的隐式响应或基于 ACK 的显式响应作为监测的主要方式,文献[1,9-10]等指出现有监测机制的不准确性是声誉系统应用的主要障碍;博弈激励机制大多建立在“针锋相对(TFT)”策略及其变种的基础上,目前提出的方案多是各节点根据自己数据传输的成功率来调整为网络中其他节点中继的概率<sup>[11-13]</sup>. 这类文献重在纳什均衡的证明,使用了较强的假设条件,距离实际应用尚有一段距离.

上述文献提出的激励机制大多是针对数据传输阶段节点的自私行为而设计的,一个基本假设是节点在路由发现阶段采取合作策略,而只在数据传输阶段自私丢包<sup>[5,8,11,13]</sup>,显然这个假设不尽合理. 对于 Ad hoc 网络中通常采用的按需路由来说,节点在寻路阶段的自私行为可以使它免于后续的数据中继任务,“合法”地节省更多的能量,因此节点倾向于在寻路阶段采用自私策略,我们必须考虑相应的检测和合作激励机制. 寻路阶段的自私行为可以分为两大类型:

(1) 主动篡改路由控制包. 当自私节点接收到 RREQ(路由请求)、RREP(路由响应)等控制包时,它改变其中一些关键域(如 AODV 中的跳数、DSR)中的中间节点列表,从而使自己避免出现在源和目的均为其他节点的路径上,逃避中继任务. 这类自私行为的应对方式已经被广泛研究<sup>[7,14]</sup>.

(2) 被动丢弃. 自私节点丢弃所接收到的路由控制包,避免成为中继节点. 两种类型的路由控制包中,RREP 的性质与数据包类似,可采用 Watchdog 机制检测自私丢弃行为. 但是 RREQ 包为广播包,而且它的传输是有条件的,存在大量合法丢包,因此 Watchdog 机制不能有效检测被动丢弃 RREQ 包的

自私行为. 目前尚未有应对这类自私行为的有效方式.

本文主要研究 RREQ 丢弃的检测、惩罚和激励机制,因此后续章节所提及的自私行为将主要指 RREQ 的被动丢弃. 虽然 RREQ 的有条件传输和邻居集的不确定性使得基于单个包检测的 Watchdog 机制失效,但是,由于每一个 RREQ 都会被接收到的节点再次广播直到该节点拥有到目的节点的路径或者 TTL 超时,因此从统计角度来看,合作节点中继的 RREQ 与返回的 RREP 之和应该远大于自身所生成的 RREQ 数. 这一点可作为检测被动丢弃的基础. 本文提出一种被动丢弃行为的检测和激励机制,基本思想是节点监测过去一段时间内的邻居中继和生成的 RREQ 数量,如果两者比率超过一定门限,则认为邻居是合作节点,否则认为邻居是自私节点. 节点以一定的概率丢弃来自自私节点的包,作为对自私行为的惩罚,从而激励合作. 由于 Ad hoc 网络中节点移动模型、业务模型、通信模型等均不相同,包括 Watchdog 在内的各种检测算法都不可避免地存在一定的误判率,本文算法也不能例外. 我们建立简化的博弈模型对算法进行分析,研究误判率对合作激励的有效范围的影响. 分析结果表明,即使存在一定的误判率,本文算法也能够激励节点达成合作. 最后我们通过仿真对算法进行验证,仿真结果表明本文算法能够对自私节点进行有效的惩罚,从而激励合作.

## 2 节点寻路阶段自私行为的检测和惩罚算法

### 2.1 基本思想

本算法的出发点来自于这样一个观察: 由于 RREQ 的广播本质,对于业务量较均匀的网络来说,从统计上来看,合作节点自身所生成 RREQ 数应小于其中继的 RREQ 数和响应的 RREP 数之和. 在 Ad hoc 按需路由中,节点接收到 RREQ 后是否继续广播与 RREQ 的内容以及本地路由表有关. 例如,如果节点曾经接收过该 RREQ,那么这个包将被丢弃;如果节点具有到目的地的路径,这个包也不再前传,而是生成 RREP 返回源节点. 因此,一个节点自私与否无法逐包来判断. 但是,对按需路由协议来说,当节点有数据要发送且没有可用路由的时候,必须首先通过 RREQ/RREP 来建立数据传输路径. 这里的 RREQ 既可以是节点自己生成的,也可以是

中继其他节点的. 由于 Ad hoc 按需路由协议中的寻路都是通过广播实现的, 因此节点中继的 RREQ 数与响应的 RREP 数之和应大于它自身生成的 RREQ 数. 后续章节中我们将只考虑相应的 RREQ 数量, 这是因为除了节点数极少的网络外, RREQ 的数量将远大于 RREP.

我们通过仿真来验证这个观察. 仿真建立在 NS2 的平台上. 在  $1000\text{m} \times 1000\text{m}$  的区域内分别随机抛洒 30 和 10 个节点, 节点位置服从均匀分布. 每个节点从剩余节点中随机选择其目的地, 每一源-目的对建立一个 CBR 流, 包长度为 512 字节, 两包之间间隔为 1s. 采用 Random Way Point 的随机移动模型. 这种模型中, 节点随机移动到某一位置后停顿一段时间再开始新的移动. 设节点的移动速度在

$1\text{m/s} \sim 10\text{m/s}$  之间均匀分布, 停顿时间在  $10\text{s} \sim 50\text{s}$  之间均匀分布. 传输模型为 TwoRayGround, 节点通信半径约为 250m. 采用 DSR 作为路由协议. 仿真持续 1000s.

在每个节点的路由 Agent 中设置一张表, 记录过去 200s 内收到的邻居自己生成的 RREQ( $REQ_s$ ) 和中继的 RREQ( $REQ_r$ ) 的数量以及两者的比值  $\xi = REQ_r / REQ_s$ . 为了排除初始阶段不稳定性的影响, 我们的统计范围为  $REQ_s > 1$  的那些记录. 30 个和 10 个节点场景下  $\xi$  的分布如图 1 所示. 可以看出, 两个场景中  $\xi < 1$  的记录所占百分比都非常小, 10 个节点场景中约为 6%, 30 个节点场景中约为 1%. 随着节点数的增加, 峰值所对应的  $\xi$  值也随之增大.

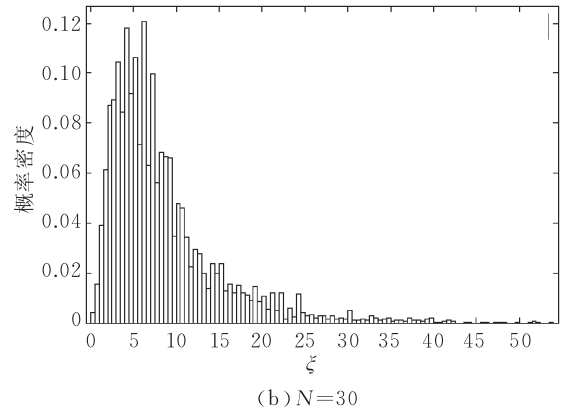
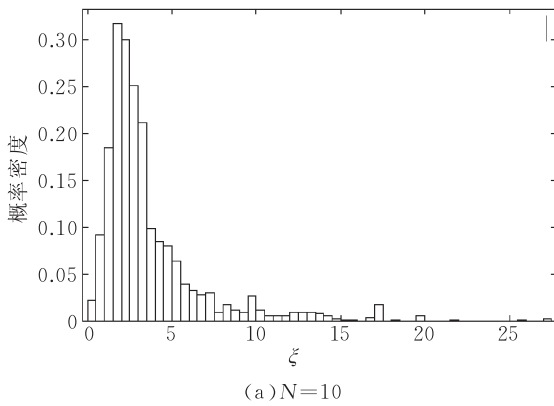


图 1 30 个节点和 10 个节点场景中  $\xi$  的分布情况

上述仿真结果证明了我们的观察在均匀业务下是成立的. 但是在业务极其不均匀的情况下我们的观察也可能不成立. 例如某一高速移动的节点有大量的数据要发送, 在此期间其他所有节点不需要寻路, 这时, 该节点的  $\xi$  将小于 1. 可以看出这种情况的发生概率是比较小的. 小概率的误判对合作激励机制的影响我们将在第 3 节讨论.

## 2.2 自私检测和惩罚算法

用 NB 来表示邻居节点, 用  $NB.REQ_r$  表示该节点中继的 RREQ 数,  $NB.REQ_s$  表示该节点生成的 RREQ 数. 为了减少误判, 同时提高算法的灵敏度, 我们设置两类门限:  $Th_i$  和  $Th_a, Th_s$ .  $Th_i$  是初始门限. 考虑一个刚刚加入网络的邻居, 节点对它的观察是  $NB.REQ_s = NB.REQ_r = 0$ , 不能判定该邻居是否自私, 这时它的寻路包应该得到中继. 只有在有明确的证据表明该邻居为自私时, 才对其进行惩罚. 因此当  $NB.REQ_s \leq Th_i$  时, 我们认为该邻居处于初始化阶段, 不对它进行惩罚.  $Th_a$  和  $Th_s$  代表了节点性质

的判断门限. 当  $\xi$  值小于  $Th_a$  时, 认为节点是自私的, 按照概率  $Th_p$  丢弃来自该节点的路由包作为惩罚. 当  $\xi$  值小于  $Th_s$  时, 认为节点是极度自私的, 以更严厉的丢弃数据包的方式对节点进行惩罚. 这两个门限值的选择与网络拓扑、业务模型等密切相关, 其关系待下一步工作讨论. 检测和惩罚算法如下:

```
OnRecvRequest(Packet RREQ)
/* Reverse Route Handling */
.....
/* Observation */
now = Scheduler::instance().clock();
Nb = Lasthop-of(RREQ);
RSrc = Route-SOURCE-of(RREQ);
if(Nb == RSrc)
    Nb.REQ_s_add(now);
else
    Nb.REQ_r_add(now);
/* Punishment for Selfish Node */
if(RSrc.REQ_s > Th_i)
```

```
ratio=RSrc.REQ_r/ RSrc.REQ_s
if(ratio<Th_a&&Random::uniform(0,1)<Th_p)
    drop(RREQ);
return;
/* Handling and Relaying of RREQ */
.....

OnRecvData(Packet DATA)
/* Punishment for Selfish Node */
DSrc=Route-SOURCE-of(DATA);
if(DSrc.REQ_s>Th_i)
    ratio=DSrc.REQ_r/DSrc.REQ_s
    if(ratio<Th_s&&Random::uniform(0,1)<Th_p)
        drop(DATA);
    return;
/* Handling and Relaying of DATA */
.....
```

当节点收到 RREQ 时,判断其路由源节点和上一跳节点. 如果两者相同,认为该包是邻居节点生成的,若两者不同,则认为是邻居节点中继的. 更改相应的  $REQ_r$  和  $REQ_s$  值. 需要注意的是这两个值均是过去一段时间内的统计值,时间段外的情况不在考虑范围内,时间段的推移在 *add* 函数中实现. 如果 RREQ 包是邻居生成的,计算  $\xi$ ,如果  $\xi > Th_a$ ,该包得到正常处理,如果  $\xi < Th_a$ ,以概率  $Th_p$  丢包. 当节点收到数据包 DATA 的时候,判断其源节点的  $\xi$ ,如果  $\xi < Th_s$ ,以概率  $Th_p$  丢弃.

为了避免被检测出来,自私节点可能采取两种规避措施:(1)更改寻路包的源地址选项,使得该 RREQ 看起来是由该邻居节点中继的,从而欺骗检测机制.(2)变更自己的身份,如采用 Sybil 攻击<sup>[15]</sup>. 这时,由于邻居身份是虚假的,检测机制不能有效发挥作用.

第一个问题可采用文献[7]中应对主动篡改攻击的方式来解决. 在公钥管理系统的支持下,当节点发送 RREQ 时,必须利用自己的私钥对其进行签名. 邻居以及中间节点可以对 RREQ 包的发起方身份进行验证. 如果验证失败的话,丢弃该包. 这样自私节点不能改动寻路包的源地址项,从而解决了第一个问题.

Sybil 攻击是基于声誉的系统共同面对的一个问题. 文献[15]对此进行了深入的研究,并提出了多种解决方案,如无线资源检测、密钥验证等等. 这些方法的思路可用来解决第二个问题. 例如,可采用基于密钥预分配<sup>[15]</sup>或者基于公钥的身份认证体系<sup>[16]</sup>来保证节点身份的真实性. 这些方法在文献中都已

进行了深入的研究,本文不再进行详细讨论.

### 3 算法分析

算法的表现与  $Th_i$ ,  $Th_a$ ,  $Th_s$  和  $Th_p$  四个参数密切相关,其中  $Th_i$ ,  $Th_a$ ,  $Th_s$  确定了检测的误判率,  $Th_p$  决定了算法的惩罚力度. 这里所说的误判率不仅是指把合作节点误判为自私节点的概率,也包括把自私节点误判为合作节点的概率,即通常意义上的敏感度. 由于网络情况千变万化,不论  $Th_i$ ,  $Th_a$ ,  $Th_s$  取什么值,静态设定或者动态变化,都可能存在误判. 本节利用博弈论的工具对算法在一定误判率情况下合作激励的有效性进行分析.

#### 3.1 系统建模

我们用随机配对重复博弈来对网络的寻路过程进行抽象. 所谓随机配对博弈,即随机选择相互作用的两个节点,它们均需要对方作为中继才能到达各自的目的地. 为了分析的方便,我们把时间轴划分为离散的时槽. 假设一个时槽内网络拓扑保持不变,时槽之间拓扑随机变化. 每时槽内两节点均进行一次寻路和数据传输过程. 节点可以选择自私丢包或者合作中继的策略. 如果对方选择中继,则源节点可获得  $\alpha$  单位的收益. 两节点数据传输的开销均为  $\beta$  单位. 一般来说,  $\alpha$  应远大于  $\beta$ . 每时槽内两个节点的交互可建模为一次阶段博弈. 用  $A_i = \{\text{合作}(C), \text{自私}(D)\}$  来表示  $i$  节点的可选行动.  $a$  来表示博弈的两个节点的行动组合,其中  $a_i$  是  $i$  节点的行动,  $a_{-i}$  是除  $i$  节点外其他节点的行动.  $u_i$  来表示节点  $i$  在阶段博弈中获得的支付,则阶段博弈的支付矩阵如表 1 所示.

表 1 阶段博弈的支付矩阵

	合作(C)	自私(D)
合作(C)	$(\alpha - 2\beta, \alpha - 2\beta)$	$(-2\beta, \alpha - \beta)$
自私(D)	$(\alpha - \beta, -2\beta)$	$(-\beta, -\beta)$

可以看出阶段博弈的局势是一个典型的囚徒困境,其纳什均衡是(自私,自私),也就是说静态策略是无法促成合作的. 把本文的算法建模为下述动态策略:

$$\sigma_i : a_i^{(t+1)} = f(\epsilon, a^{(t)}, a^{(t-1)}, \dots),$$

其中,  $\epsilon = P(\hat{a}_{-i}^{(t)} = D/a_{-i}^{(t)} = C) = P(\hat{a}_{-i}^{(t)} = C/a_{-i}^{(t)} = D)$  为对手行为的误判率,  $\hat{a}$  为检测到的对手的行为,  $a$  为对手真正的行为. 尽管在实际应用中,这两个概率是不同的,并且随时间变化,但为了分析的方便,我

们假设两者相同且不随时间变化,均为 $\epsilon$ ,粗略地反应了本文算法中 $Th_i, Th_a, Th_s$ 的影响。 $f$ 函数对应着节点根据过往信息决定当前行为的方式:如果检测出对方自私的话,本节点以概率 $p$ 丢弃对方的包, $p$ 反应了本文算法中 $Th_p$ 的影响。

行为策略的整体收益通过贴现平均准则来计算:

$$U_i = (1-\delta) \sum_{t=1}^{\infty} \delta^{t-1} u_i^{(t)},$$

其中贴现因子 $0 \leq \delta < 1$ 代表了对节点耐心和远见的一种度量,或者是博弈在下一回合仍然继续的概率。它在 Ad hoc 网络中的意义有不同的解释,文献[12]把 $1/(1-\delta)$ 解释为会话持续时间,文献[17]把 $\delta$ 解释为两节点再次相遇的概率。除了上述两个因素外,本文用 $\delta$ 来反应节点能量限制的影响。这是因为,当能量即将耗尽的时候,节点不再关心未来的收益, $\delta$ 趋近于0;而能量充足且需要使用网络的时间较长时,节点比较关注未来收益, $\delta$ 趋近于1。能够促成合作的 $\delta$ 值的下界用 $\delta^*$ 来表示。 $\delta^*$ 代表了一个行为策略促成合作的有效范围。 $\delta^*$ 越大,有效范围越小; $\delta^*$ 越小,则有效范围越大。

动态行为策略 $\sigma$ 是一个均衡的充要条件是对每个局中人 $i$ 和每个行为策略 $\sigma'_i$ ,使用 $\sigma$ 所获得的贴现收益要大于等于使用 $\sigma'_i$ 所获得的贴现收益[18]:

$$(1-\delta) \sum_{t=1}^{\infty} \sum_{a \in D} P^{(t)}(a/\sigma_{-i}, \sigma_i, \epsilon) \delta^{t-1} u_i^{(t)}(a) \geq (1-\delta) \sum_{t=1}^{\infty} \sum_{a \in D} P^{(t)}(a/\sigma_{-i}, \sigma'_i, \epsilon) \delta^{t-1} u_i^{(t)}(a) \quad (1)$$

其中, $P^{(t)}(a/\sigma, \epsilon)$ 表示若每个局中人 $i$ 在每个回合都使用行为策略 $\sigma_i$ 并存在 $\epsilon$ 的误判率时, $t$ 时槽局中人所采用的行动组合为 $a$ 的概率。

### 3.2 合作性分析

我们假设节点当前的行为只与上一时槽的观测值相关, $p = P(a_i^{(t+1)} = C/\hat{a}_i^{(t)} = D)$ ,这时整个过程可建模为一个具有四个状态 $(C, C), (C, D), (D, C), (D, D)$ 的马尔可夫链,节点当前阶段行动与上一阶段对手行动的条件概率为

$$\tau_{11} = P(a_i^{(t+1)} = C/\hat{a}_i^{(t)} = C) = (1-\epsilon) + (1-p)\epsilon,$$

$$\tau_{12} = P(a_i^{(t+1)} = C/\hat{a}_i^{(t)} = D) = (1-\epsilon)(1-p) + \epsilon,$$

$$\tau_{21} = P(a_i^{(t+1)} = D/\hat{a}_i^{(t)} = C) = p\epsilon,$$

$$\tau_{22} = P(a_i^{(t+1)} = D/\hat{a}_i^{(t)} = D) = (1-\epsilon)p.$$

用 $\mathbf{P}$ 来表示马尔可夫链的转移矩阵,用 $\mathbf{B}$ 来表示阶段博弈的支付向量, $\mathbf{U}^{(t)}$ 来表示从 $t$ 时槽起的贴现支付:

$$\mathbf{P} = \begin{bmatrix} \tau_{11}^2 & \tau_{11}\tau_{21} & \tau_{11}\tau_{21} & \tau_{21}^2 \\ \tau_{11}\tau_{12} & \tau_{21}\tau_{12} & \tau_{11}\tau_{22} & \tau_{21}\tau_{22} \\ \tau_{11}\tau_{12} & \tau_{11}\tau_{22} & \tau_{21}\tau_{12} & \tau_{21}\tau_{22} \\ \tau_{12}^2 & \tau_{12}\tau_{22} & \tau_{12}\tau_{22} & \tau_{22}^2 \end{bmatrix},$$

$$\mathbf{B} = (u_i(C, C), u_i(C, D), u_i(D, C), u_i(D, D))^T,$$

$$\mathbf{U}^{(t)} = (U_i^{(t)}(C, C), U_i^{(t)}(C, D), U_i^{(t)}(D, C), U_i^{(t)}(D, D))^T.$$

那么可通过下式计算出各状态的贴现收益:

$$\mathbf{U}^{(t)} = (1-\delta)\mathbf{B} + \delta\mathbf{P}\mathbf{U}^{(t+1)} \quad (2)$$

由于无限重复博弈中从各时槽开始的博弈结构是一样的,因此我们可以省略掉时间上标 $t$ ,从而得到各状态的贴现收益为

$$\mathbf{U} = (1-\delta)(1-\delta\mathbf{P})^{-1}\mathbf{B}.$$

由于存在着误判率,因此系统不可能总保持在 $(C, C)$ 的状态,这种情况下合作的含义就是节点遵循行为策略时的收益应大于其偏离该策略的收益,如式(1)。具体来说,就是在策略要求节点合作时,节点采用合作行动的收益应大于采取自私行动的收益,即

$$U_i(C, *) > U_i(D, *) \quad (3)$$

式中, $*$ 为 $C$ 或者 $D$ 。我们对 $\alpha, \beta, \epsilon, p$ 等参数取不同的值,以观察策略促成合作的有效范围。 $\delta^*$ 与各参数之间的关系如图2所示。

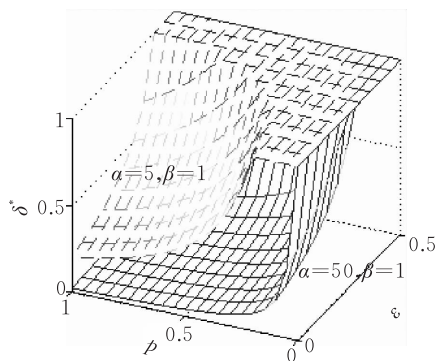


图2 合作范围与 $\epsilon, p$ 的关系

从图2中我们可以观察不同参数对合作范围的影响:

(1)  $\delta^*$ 随着 $\alpha, \beta$ 的比率的增大而降低,其原因是数据成功传输所获得的收益越大,节点就越倾向于合作。

(2)  $\delta^*$ 随着 $p$ 的增加而降低,其原因可解释为惩罚越严厉,节点偏离既有策略所获得的收益越小,节点越倾向于合作。

(3)  $\delta^*$ 随着 $\epsilon$ 的增加而增加,当 $\epsilon=0.5$ 时,不论其他参数为何值,节点均不能达成相互合作。其原因

是,误判率越大,节点所获得的对对手的信息越少,就越倾向于采用安全的自私策略以保证自身利益的最大化.

可以看出,当 $\epsilon$ 较小, $p$ 较大时,不等式(3)对于大部分 $\delta$ 值均成立,系统达到合作的均衡状态.这说明本算法可有效实现合作激励的功能.

### 4 仿真验证

我们仍采用第 2 节中 30 个节点的仿真场景对算法进行仿真验证.算法参数选择为  $Th_i=1, Th_a=$

4,  $Th_s=1$  和  $Th_p=1$ . 节点的初始能量为 5,发送和接收能量配置分别为 0.1 和 0.025,仿真时间设为 2000s.

首先我们考察不同自私节点数时算法的表现.设置三种场景,基线场景:所有节点均合作;场景 1:自私节点丢弃 RREQ 包,但是没有启用本文算法;场景 2:自私节点丢弃 RREQ,但是合作节点启用了本文算法.各场景分别设 3,6,9,12,15 个节点为自私节点.对比自私节点和合作节点在不同场景下的吞吐量,如图 3 所示.

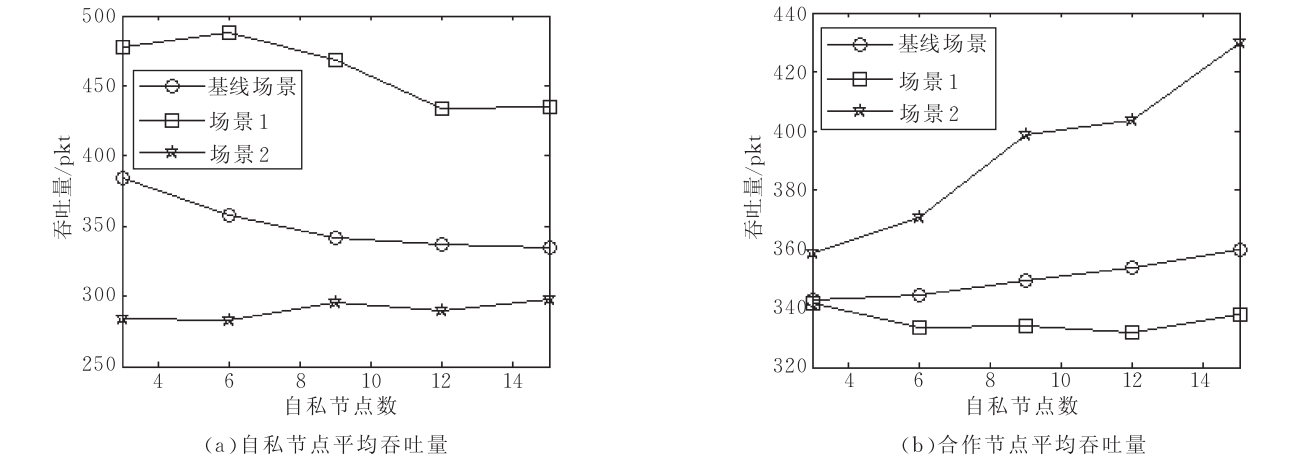


图 3 不同场景下吞吐量与自私节点数目的关系

首先观察自私节点吞吐量的变化规律.对比基线场景和场景 1 的吞吐量变化,可以看出,节点自私丢弃 RREQ,逃避中继任务,节省了能量,从而使得自己的吞吐量上升.场景 2 本文算法启用之后,对自私节点进行有效检测和惩罚,使得自私节点吞吐量大幅度下降,低于其合作状态,因此理性的自私节点将倾向采用合作策略,从而实现对自私节点的激励.

合作节点的吞吐量变化规律与此相反.自私节点丢弃 RREQ 包使得合作节点需要花费更多的能

量中继包,导致自己的吞吐量下降.算法启用后,合作节点不再中继自私节点的包,从而可以节省能量用于中继合法包.自私节点数目越多,节省的能量越多,吞吐量提高的幅度就越明显.

下面我们考虑本文算法对选择性丢包的应对能力.设自私节点数目为 6,即 20% 的节点自私丢包,丢包概率分别为 50%,60%直到 100%.两种类型的节点在不同场景下的平均吞吐量如表 2 所示.

表 2 不同场景下不同类型节点的平均吞吐量

自私节点丢包概率/%	场景 1:不采用本文算法		场景 2:采用本文算法	
	合作节点平均吞吐量/包	自私节点平均吞吐量/包	合作节点平均吞吐量/包	自私节点平均吞吐量/包
100	342.625	388.833	383.708	257.5
90	350.917	330.833	374.125	294.667
80	344.5	321.833	362.167	318
70	343.75	314.667	362.333	333.167
60	348.167	325	360.042	339.5
50	344.375	315.333	363.25	348.5
0(合作状态)	354.708	316.667	360	348.833

从表 2 可以看出,在合作状态,本文算法并不会给网络性能带来负面影响.在场景 1 本文算法没有启用的时候,自私节点只有在丢包率较高( $>80\%$ )

时,吞吐量才能得到显著的增加.场景 2 中合作节点启用本文算法之后,丢包率越大,遭受的惩罚力度越大,自私节点的吞吐量越小.当丢包率大于 80% 时,

其吞吐量远小于合作状态的 348.833,因此本文算法激活后,理性节点将选择合作而不是自私丢包,从而实现对自私节点的激励。

## 5 相关工作

文献[8]是 Ad hoc 网络自私行为检测和激励领域的开创性文献。文献[8]关注的是数据传输阶段的自私丢包问题,并提出了后续文献中广泛应用的 Watchdog 检测方案。Watchdog 建立在 DSR 的基础上,并设定节点工作在混杂侦听模式下。节点发包之后,侦听下一跳节点的通信。如果在设定时间内,没有听到该包被继续传送到路径上的下一节点,那么认为下一跳节点自私丢包。当节点检测到其下一跳自私丢包的比率超过一定门限时,它将通知源节点。源节点中的 Pathrater 部件在选择路径时,避开丢包的自私节点。

从 Watchdog 的机理中可以看出它并不适合于寻路阶段 RREQ 的丢包检测;首先,RREQ 是广播包,链路层不提供应答,因此节点不能确定此时邻居域内哪些节点接收到 RREQ,哪些没有。不能确定需要监测的节点,也就无法对节点是否中继 RREQ 包做出判断。其次,由于一个 RREQ 可能被多个节点广播,因此为了降低寻路开销,节点对同一个 RREQ 只作一次响应,后续收到的重复 RREQ 将被丢弃,因此存在大量合法丢包。这两个方面使得 Watchdog 无法用于寻路阶段自私丢包的检测。

文献[19]提出了一种基于 ACK 的自私丢包检测方式 2ACK,其核心思想是传输路径上的节点接收到数据包后返回 ACK 给两跳前的节点,从而可以对中间节点的传输行为进行判定。2ACK 方法也建立在 DSR 的基础上。寻路过程中节点邻居集未知、存在合法丢包的特点同样使得 2ACK 不能正常发挥作用。尽管节点收到下两跳的 ACK 时能够确定下一跳节点是合作的,但是没有收到 ACK 时,却不能判定下一跳自私丢包。因此这种方式也不能用于寻路过程中。

文献[20]提出一种基于虚拟支付的紧凑激励机制来促使节点在寻路阶段合作。收到 RREQ 后再次广播的节点可以得到小额的支付,对最终构成源目的路径的节点则可以获得大额支付。参与路由越多的节点获得的支付越多,从而使节点愿意合作。但是文献[20]中只是简单地描述了这种思想,对于支付如何实现、支付的数量等细节问题均缺乏进

一步的说明。

## 6 结 论

对于采用按需路由的 Ad hoc 网络来说,自私节点丢弃寻路包从而合法逃避为其他节点中继数据的责任是节约自己能量的最好方式。由于路由控制包的有条件中继模式和邻居的不确定性,这类自私方式很难通过传统的 Watchdog 或基于 ACK 的方式来检测。本文基于寻路包的统计特性,提出了一种被动丢弃寻路包的行为检测和惩罚算法,使得自私节点的收益大幅度下降。本文建立博弈模型对算法的有效性进行分析,结果表明即使存在一定的误判率,我们的算法仍能够促使节点达成合作。仿真结果同样也证明了我们的算法能够有效惩罚自私节点,从而激励节点合作。

## 参 考 文 献

- [1] Yoo Y, Agrawal D P. Why does it pay to be selfish in a MANET? IEEE Wireless Communications, 2006, 13(6): 87-97
- [2] Buttyan L, Hubaux J-P. Stimulating cooperation in self-organizing mobile Ad hoc networks. Mobile Networks and Applications, 2003, 8(5): 579-592
- [3] Wang W, EideNBenz S, Wang Y, Li X-Y. OURS: Optimal unicast routing systems in non-cooperative wireless networks//Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM). Los Angeles: ACM Press, 2006: 402-413
- [4] Zhong S, Chen J, Yang R. Sprite: A simple, cheat-proof, credit-based system for mobile Ad-hoc networks//Proceedings of the INFOCOM. San Francisco: IEEE Press, 2003: 1987-1997
- [5] Buchegger S, Boudec J-Y L. Performance analysis of the CONFIDANT protocol cooperation of nodes fairness in dynamic Ad hoc networks//Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). Lausanne: ACM Press, 2002: 226-236
- [6] Buchegger S, Boudec J-Y L. Self-policing mobile Ad hoc networks by reputation systems. IEEE Communications Magazine, 2005, 43(7): 101-107
- [7] He Q, Wu D, Khosli P. A secure incentive architecture for Ad-hoc networks. Wireless Communications and Mobile Computing, 2006, 6(3): 333-346
- [8] Marti S, Giulì T J, Lai K, Baker M. Mitigating routing misbehavior in mobile Ad hoc networks//Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM). Boston: ACM Press, 2000: 255-265

- [9] Huang E, Crowcroft J, Wassell I. Rethinking incentives for mobile Ad hoc networks//Proceedings of the ACM SIGCOMM 2004 Workshops. Portland; ACM Press, 2004; 191-196
- [10] Carruthers R, Nikolaidis I. Certain limitations of reputation—based schemes in mobile environments//Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems. Montreal; ACM Press, 2005; 2-11
- [11] Srinivasan V, Nuggehalli P, Chiasserini C F, Rao R R. Co-operation in wireless Ad hoc networks//Proceedings of the INFOCOM. San Francisco; IEEE Press, 2003; 808-817
- [12] Milan F, Jaramillo J J, Srikant R. Achieving cooperation in multihop wireless networks of selfish nodes//Proceedings of ACM Workshop on Game Theory for Communications and Networks. Pisa; ACM Press, 2006
- [13] Felegyhazi M, Buttyan L. Nash equilibrium of packet forwarding strategies in wireless Ad hoc networks. IEEE Transactions on Mobile Computing, 2006, 5(5): 463-475
- [14] Zapata M-G, Asokan N. Securing Ad hoc routing protocols//Proceedings of the 3rd Workshop on Wireless Security. New York; ACM Press, 2002; 1-10
- [15] Newsome J, Shi E, Song D, Perrig A. The Sybil attack in sensor networks: Analysis and defenses//Proceedings of the International Symposium on Information Processing in Sensor Networks (IPSN). New York; ACM Press, 2004; 259-268
- [16] Li R-D, Li J, Liu P, Chen H-H. On-demand public-key management for mobile Ad hoc networks. Wireless Communications and Mobile Computing, 2006, 6(3): 295-306
- [17] Urpi A, Bonuccelli M, Giordano S. Modeling cooperation in mobile Ad hoc networks: A formal description of selfishness//Proceedings of Modeling and Optimization in Mobile, Ad hoc and Wireless Networks. Sophia-Antipolis; IEEE Press, 2003
- [18] Myerson R B. Game Theory, Analysis of Conflict. Beijing; Chinese Economy Press, 2001
- [19] Liu K, Deng J, Varshney P K, Balakrishnan K. An acknowledgment-based approach for the detection of routing misbehavior in MANETs. IEEE Transactions on Mobile Computing, 2007, 6(5): 488-501
- [20] Zhu H-F, Bao F, Li T-Y. Compact stimulation mechanism for routing discovery protocols in civilian Ad hoc networks//Proceedings of the IFIP International Federation for Information Processing. LNCS 3677. 2005; 200-209



**HUANG Lei**, born in 1972, Ph.D. candidate, senior engineer. Her research interests include cooperation stimulation in Ad hoc networks, congestion control in satellite network, etc.

**LIU Li-Xiang**, born in 1973, Ph.D., associate researcher. His research interests include Ad hoc network, routing protocol in satellite network, novel network architecture, network control, etc.

## Background

Mobile Ad-hoc networks are basically peer to peer multi-hop wireless networks. Key networking tasks must be carried out on the base of mutual cooperation among these nodes. If these nodes belong to different profit-maximizing entities, for example, in a commercial scenario, cooperation assumption doesn't hold. To save the resources such as energy, a selfish node may refuse to relay the packets for others.

How to stimulate selfish nodes to cooperate became a hot spot in Ad hoc network research community recently. Current studies proposed a lot of schemes to detect selfish dropping in data forwarding stage, such as watchdog or 2ACK. But it is easy to escape the detection of these schemes

by just silent dropping control packets in route discovery stage. Because of the uncertainty in route discovery, it is hard to detect such kind of selfishness. This paper proposed a detection mechanism for silent dropping in route discovery based on the statistics of route control packet. Authors proved the effectiveness of the proposed algorithm to stimulate cooperation with the tools of game theory and by simulation.

This research is partially supported by the advanced research foundation of Chinese Academy of Sciences under grant No. 9140A150301.