第43卷 第8期 西 安 交 通 大 学 学 报 Vol. 43 №8
2009年8月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Aug. 2009

面向信誉的自组网节点合作方法

郭建立，刘宏伟，吴智博，杨孝宗
（哈尔滨工业大学计算机科学与技术学院，150001，哈尔滨）

摘要：针对移动自组网中自私节点出于节省资源而表现出不合作行为及丢弃待转发数据等问题，提出了一种面向信誉的自组网节点合作(CMC)方法，且第一次引入了共同邻居监听技术.看门狗在对下一跳转发节点进行监听的同时，可对周围不相关的数据流进行监听，从而加快了系统对不合作节点的检测速度.在路由发现过程中，CMC对路由控制消息进行了过滤，丢弃含有不合作节点的路由请求包和路由应答包，使源节点发现的路由能尽量绕过不合作节点.在ns-2下的仿真结果表明，当网络中存在10%～60%的不合作节点时，CMC能够使合作节点的吞吐率提高10%～40%.

关键词：移动自组网；节点合作；共同邻居监听；信誉
中图分类号：TP393 **文献标志码**：A **文章编号**：0253-987X(2009)08-0017-05

A New Reputation-Based Cooperation Enforcement Scheme for MANETs

GUO Jianli, LIU Hongwei, WU Zhibo, YANG Xiaozong
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: In mobile ad hoc networks, selfish nodes show non-cooperative behaviors due to discarding packets to save their resources (such as energy), which will seriously affect the network performance. A new cooperation enforcement scheme called CMC is proposed to mitigate this problem. In the CMC, the common neighbor monitoring technique is introduced to help the watchdog monitor all packets transmitting around it so that the system can detect the non-cooperative nodes more quickly. In the routing discovery phase, the route request packets and the route reply packets that contain non-cooperative nodes are dropped by CMC, so that the probability that a well-behaved node uses a bad route for data transmission is reduced. The ns-2 simulation results show that the CMC can improve the throughput of well-behaved nodes by 10%-40% in the presence of 10%-60% non-cooperative nodes.

Keywords: mobile ad hoc network; node cooperation; common neighbor monitoring; reputation

在移动自组网[1]中，节点的无线通讯范围是有限的，2个相距较远的移动节点的通信往往要借助其他节点的转发功能，并假设节点是合作的.近年来，在民用自组网中，各节点分别隶属于不同的个人或组织，缺乏共同的目的，节点间的合作将无法保证.自私节点为了节省能量，可能会丢弃那些需要转发的数据包，从而表现出不合作行为[2].文献[3]运用博弈论从理论上证明了，在移动自组网中不存在自发的节点合作.文献[4]指出，如果网络中存在10%～40%的自私节点，就会导致整个网络的性能下降16%～32%.

针对移动自组网中节点合作问题，研究者们提出了许多解决方案[5-6].文献[2,7-9]中提出了基于虚拟货币的方法，但其中存在着诸如节点间信息交换量大等问题.文献[4]中第一次提出使用看门狗技术检测不合作节点，之后又相继出现了采用二手信息计算节点信誉值法[10]、两跳ACK检测法[11].文献[12-13]对信誉值计算方法进行了分析和总结，并

收稿日期：2008-12-18. 作者简介：郭建立(1980—)，男，博士生；杨孝宗(联系人)，男，教授，博士生导师. 基金项目：国家高技术研究发展计划资助项目(2006AA01A103，2008AA01A201)；国家自然科学基金资助项目(60503015).

指出了使用二手信息计算节点信誉值时所存在的一些缺点,这使得基于直接信息的计算方法更适用于自组网,但其中也存在着不足,如对不合作节点的检测速度较慢.

为此,本文提出了共同邻居监听节点合作(Common-Neighbor Monitoring Enabled Cooperation Enforcement Scheme, CMC)的方法,以加快在使用直接信息计算节点信誉值时对不合作节点的检测速度.

1 CMC 方法

1.1 CMC 结构描述

CMC 基于动态源路由(DSR)协议,位于网络层和媒体访问控制(MAC)层之间,包含看门狗、过滤器、邻居管理器、信誉管理器和二次机会机制 5 个组件,以及黑名单和邻居列表 2 个重要数据结构,如图 1 所示.

DSR 协议的功能和代码保持不变,只有 FindRoute 函数被改写. FindRoute 函数根据黑名单在路由缓存中搜索可用路由,返回不含黑名单中节点的路由.

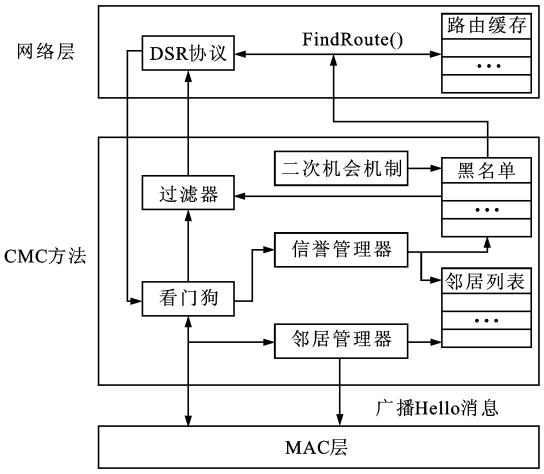


图 1 CMC 结构

1.2 看门狗

看门狗负责对邻居节点进行监听,但需要维护一个数据结构(数据包监听缓存),并在其中存放需要监听的数据包. DSR 协议发出的数据包只要其下一跳节点不是目的节点(目的节点不转发数据包,因此无法对其监听),就在监听缓存中对其保存一份拷贝.

接口在混杂模式下监听信道,并将接收到的数据包交给看门狗. 如果看门狗在监听缓存中找到该数据包,并且其未被篡改,就向信誉管理器发送一个正事件(Positive Event),增加转发节点的信誉值;如果直到监听缓存中的数据包超时,看门狗仍未观察到对该该数据进行转发,就向信誉管理器发送一个负事件(Negative Event),减小转发节点的信誉值.

1.3 共同邻居监听技术

在监听过程中,只有位于路由上的节点才能对下一跳节点的转发情况进行监听. 为了加快看门狗对不合作节点的检测速度,本文引入了共同邻居监听技术:当看门狗捕获到一个数据包时,如果其当前和下一跳转发节点都是自己的邻居,则把该数据包放入监听缓存中,监听其转发情况.

共同邻居监听如图 2 所示,节点 S 通过节点 A、B、C 向节点 D 发送数据,节点 M 则位于 B、C 的传输范围. 当 B 向 C 转发数据时, M 发现其当前转发节点(B)和下一跳转发节点(C)都是自己的邻居,就把它们放入监听缓存中,并监听 C 的转发情况.

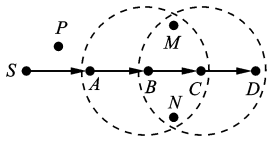


图 2 共同邻居监听

作为对不合作节点的惩罚,过滤器将丢弃所有由不合作节点发送的数据. 为了防止看门狗把这种惩罚行为误判为不合作行为,不要求对路由中第一个转发节点进行监听(图 2 中节点 P 不监听 A).

1.4 邻居管理器

从 1.3 节可以看出,在共同邻居监听中,如果维护的邻居列表不准确,将会严重影响看门狗的检测结果. 在邻居发现过程中,节点很容易遭受各种攻击,其中最著名的当属虫洞攻击,如图 3 所示,其中节点 F 通过重放 B 的 Hello 消息,使 A 把 B 误认为是邻居.

本文采用一种安全的邻居发现过程[14],借助节点的位置信息检测虫洞攻击,但要求网络中一小部分节点能够获得自己的精确位置信息,这些节点被称之为锚点.

假定节点在网络中以空间均一泊松过程模型(SHPPP)随机分布,其分布密度为 λ,锚点的分布密度为 λ^*,$\lambda^* \ll \lambda$;又设锚点的传输半径为 R,并且远远大于普通节点.

如果 2 个节点能够互相收到对方的 Hello 消息,并且能收到来自 T 个共同锚点的 Hello 消息,

则它们互相把对方作为邻居. 实验中 T 为一常数，其值为

$$T = \left\lceil \frac{1}{3}R^2\pi\lambda^* \right\rceil + 1 \tag{1}$$

在邻居发现过程中，即使 2 个节点相邻，但如果它们的共同锚点数小于 T，这 2 个节点也不能成为邻居. 可见，在空间均一泊松过程模型中，面积为 ξ 的区域内有 k 个节点的概率为

$$p_\xi^k = \frac{(\lambda\xi)^k}{k!}\mathrm{e}^{-\lambda\xi} \tag{2}$$

令 p_{fail} 表示 2 个相邻节点间的共同锚点数小于 T 的概率，显然

$$p_{\mathrm{fail}} = \sum_{k=0}^{T-1} p_{\xi_{\mathrm{cmn}}}^k = \sum_{k=0}^{T-1} \frac{(\xi_{\mathrm{cmn}}\lambda^*)^k}{K!}\mathrm{e}^{-\xi_{\mathrm{cmn}}\lambda^*} \tag{3}$$

式中：ξ_{cmn} 为图 4 中阴影区域的面积. 由式(3)知，p_{fail} 的值只与 λ^* 和 ξ_{cmn} 相关，通过计算得知，当 $\lambda^* > 0.000\,8$ 时，无论 ξ_{cmn} 在$[0,\ \pi R^2]$范围取何值，p_{fail} 的值均小于 0.1. 显然，只要保持网络中锚点的分布密度，就能使 p_{fail} 的值很小，由此不会影响邻居发现结果.

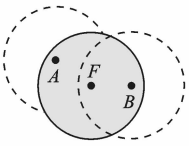


图 3　虫洞攻击

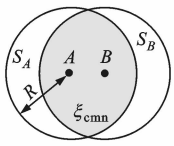


图 4　利用锚点检测虫洞攻击

1.5　信誉管理器

信誉管理器负责更新节点的信誉值. 节点初始信誉值 $\tau=0$，取值范围为$[-1,1]$. 定义阈值 $\tau_{\mathrm{Trust}}=0.8$、$\tau_{\mathrm{Faulty}}=-0.5$，并规定 $\tau>\tau_{\mathrm{Trust}}$ 的节点为可信节点，$\tau<\tau_{\mathrm{Faulty}}$ 的节点为自私节点. 如果收到正事件，信誉管理器根据式(4)增加转发节点的信誉值

$$\tau_{\mathrm{new}} = \begin{cases} \tau_{\mathrm{old}}\alpha_1 + \tau_{\mathrm{incr1}} & \tau_{\mathrm{old}} < \tau_{\mathrm{Faulty}} \\ \tau_{\mathrm{old}}\alpha_2 + \tau_{\mathrm{incr2}} & \tau_{\mathrm{old}} \geqslant \tau_{\mathrm{Faulty}} \end{cases} \tag{4}$$

式中：α 为贴现率；τ_{incr} 为信誉值增量. 实验中 $\alpha_1=0.99$，$\alpha_2=0.9$，$\tau_{\mathrm{incr2}}=0.1$，$\tau_{\mathrm{incr1}}=0.01$. 如果收到负事件，信誉管理器根据式(5) 减小转发节点的信誉值

$$\tau_{\mathrm{new}} = \begin{cases} \tau_{\mathrm{old}}\beta_1 - \tau_{\mathrm{decr1}} & \tau_{\mathrm{old}} \geqslant \tau_{\mathrm{Trust}} \\ \tau_{\mathrm{old}}\beta_2 - \tau_{\mathrm{decr2}} & \tau_{\mathrm{old}} < \tau_{\mathrm{Trust}} \end{cases} \tag{5}$$

实验中 $\beta_1=0.98$，$\tau_{\mathrm{decr1}}=0.02$，$\beta_2=0.9$，$\tau_{\mathrm{decr2}}=0.1$. 如果节点的信誉值小于 τ_{Faulty}，则认为其是不合作节点，放入黑名单中.

1.6　过滤器

过滤器主要是根据黑名单列表对经过节点的路由控制消息和数据包进行过滤. 对于每个路由请求消息(RREQ)或路由应答消息(RREP)，如果其中含有黑名单中的节点，就认为当前正在发现的路由是“坏”的，丢弃它. 另外，过滤器还对所有需要转发的数据包进行过滤，作为对不合作节点的惩罚，丢掉所有源节点位于黑名单中的数据包. 过滤算法如图 5 所示.

```
1  收到一个数据包ρ
2  if ρ是RREP或RREQ，并且其中含有
     黑名单中的节点 then
3          镇压ρ，并返回
4   else if ρ是数据包，并且源节点是
     黑名单中的节点 then
5          镇压ρ，并返回
6  end if
7  把ρ交给DSR协议继续处理
```

图 5　过滤算法

在路由发现过程中，位于源和目的节点之间的所有节点均进行路由过滤. 因此，源节点所发现的路由能够最大程度地绕开不合作节点，从而增加了数据发送的成功率.

1.7　二次机会机制

文献[4,12,14]认为有多种原因会影响看门狗的检测结果，如信号冲突、网络拥塞等，这些可能会导致看门狗把合作节点错误地标记为不合作节点. 另外，检测出的不合作节点，也可能会在后期表现出合作行为. 为了让被隔离的节点能够重新返回网络，CMC 引入了二次机会机制，即所有被放入黑名单中的节点，经过一段设定的时间后，将从黑名单中释放出来，它们的信誉值保持不变. 此目的是，一旦这些节点继续表现出不合作行为，能够快速地再次被放入黑名单中.

2　性能分析

将网络中节点看作集合 $s=\{1,2,\cdots,n\}$，并假定自私节点集合为 $s'\subset s$. 有 2 个因素会导致数据包丢失：节点不合作和网络环境因素(如网络拥塞等). 本文采用以下 2 个指标来评估不合作节点对网络性能的影响.

(1)吞吐率 r_{T}：合作节点发出的能够顺利到达目的节点的数据包数与合作节点发出的总数据包数的比值，即

$$r_{\mathrm{T}}=\frac{\sum\limits_{S\notin(s/s')\wedge D\notin(s/s')\wedge\rho_{S,D}^{\mathrm{num}}\in\Sigma_D}\rho_{S,D}^{\mathrm{num}}}{\sum\limits_{S\notin(s/s')\wedge D\notin(s/s')}\rho_{S,D}^{\mathrm{num}}} \tag{6}$$

式中：$\rho_{S,D}^{\mathrm{num}}$表示由节点$S$发送到$D$的编号为num的数据包；$\Sigma_D$表示$D$收到的数据包集合.

(2)转发吞吐率r_{FT}：去掉直接发送的数据包(不需要转发的数据包)后的吞吐率，即

$$r_{\mathrm{FT}}=\frac{\sum\limits_{S\notin(s/s')\wedge D\notin(s/s')\wedge\rho_{S,D}^{\mathrm{num}}\in\Sigma_D\wedge l(S,D)>1}\rho_{S,D}^{\mathrm{num}}}{\sum\limits_{S\notin(s/s')\wedge D\notin(s/s')\wedge l(S,D)>1}\rho_{S,D}^{\mathrm{num}}} \tag{7}$$

式中：$l(S,D)$表示节点S与D之间的路由长度，$l(S,D)=1$跳表示S与D之间不需要转发节点.

3 仿真及结果分析

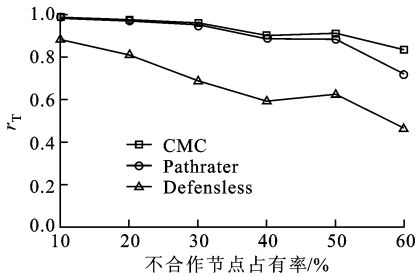
本文采用ns-2对CMC方法进行了验证，并观察不合作节点所占比率从10%增加到60%时，其对网络性能的影响，仿真基本参数设置如表1所示，仿真结果取每个组合运行10次的平均值.

表1 仿真基本参数

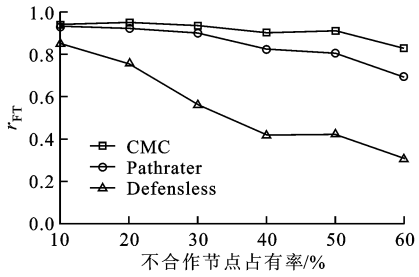
参数	设定值
仿真持续时间/s	1 000
节点传输半径/m	250
节点接收半径/m	250
载波监听半径/m	550
最大停留时间/s	100
数据包类型	恒定比特率(CBR)
数据包大小/B	512
发送速率/包·s^{-1}	5

首先，研究静态网络中合作节点的吞吐率. 选用的仿真区域为670 m×670 m，50个节点在仿真区域中随机分布，连接数为50，仿真结果如图6a所示. 可以看出，相对于Pathrater方法，CMC方法的吞吐率并未得到明显改善，只有当不合作节点比率增加到50%以上时，才略显优势. 主要原因是：在静态网络中，Pathrater方法最终也能发现所有不合作节点(只是发现速度稍慢)，并在以后的数据发送过程中绕开了不合作节点. 合作节点的转发吞吐率如图6b所示，可以看出，CMC方法相比于Pathrater方法性能有所改进.

其次，查看动态网络中合作节点的吞吐率. 节点运动模型为Random Waypoint，最大运动速度为10 m/s，最大停留时间为100 s. 仿真区域为670 m×670 m，节点数为50，CBR连接数为50，仿真结果



(a)吞吐率

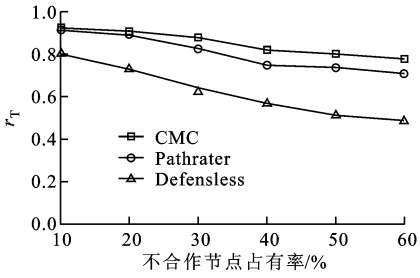


(b)转发吞吐率

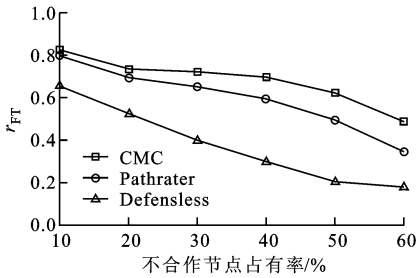
Defenseless：不使用节点合作方法；Pathrater：文献[4]方法

图6 静态网络合作节点吞土率(670 m×670 m，50个节点)

如图7所示. 可以看出，当网络中不合作节点所占比率大于30%时，CMC方法的性能要明显优于Pathrater方法.



(a)吞吐率



(b)转发吞吐率

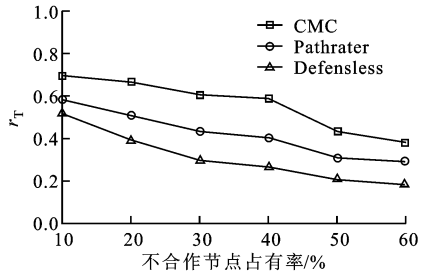
Defenseless：不使用节点合作方法；Pathrater：文献[4]方法

图7 动态网络合作节点吞土率(670 m×670 m，50个节点)

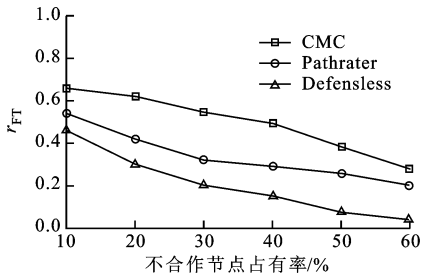
最后，增大网络规模，使用100个节点，同时将仿真区域扩大到1 200 m×1 200 m，仿真结果如图8所示. 可以看出，CMC方法明显优于Pathrater方法，这主要是由于网络规模扩大，路由的平均长度增

大所致. 图 9 所示为数据包所经过的平均路由长度. 可以看出，CMC 方法的平均路由长度大于 3.6 跳，而 Pathrater 方法的平均路由长度大于 3.4 跳，这意味着在平均情况下，每对连接大约需要 2.5 个转发节点.

在 Pathrater 方法中，源节点只能发现 1 跳范围内的不合作节点，对于 2 跳或者多跳的节点，源节点将不能判断其行为，因此很可能会选用含有不合作节点的路由，而 CMC 方法通过过滤路由控制消息，使源节点能够尽量绕开网络中的不合作节点.



(a)吞吐率



(b)转发吞吐率

Defenseless：不使用节点合作方法；Pathrater：文献[4]方法

图 8　大规模动态网络合作节点吞土率
(1200 m×1200 m，100 个节点)

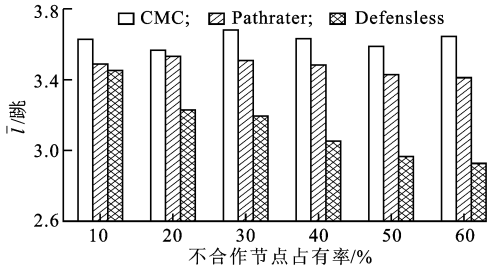


图 9　大规模动态网络的平均路由长度
(1 200 m×1 200 m，100 个节点)

4　结　论

本文提出的 CMC 方法能够快速检测出移动自组网中的不合作节点，并对其进行隔离，由此提高了网络的性能. 通过利用共同邻居监听技术，加快了看门狗对不合作节点的检测速度，使得系统可以快速地进行隔离. 在 CMC 方法中，源和目的节点之间的所有节点，都可对不合作节点的 RREQ、RREP 消息进行镇压，进一步减小了不合作节点对网络性能的影响. 仿真结果显示，当网络中存在不合作节点时，CMC 能够明显提高合作节点的吞吐率.

参考文献：

[1] CONTI M B S, GIORDANO S, STOJMENOVIC I. Mobile ad hoc networking [M]. Piscataway, NJ, USA: IEEE, 2004.

[2] DORSEY J G. Game-theoretic power management in mobile ad hoc networks[D]. Pittsburgh, Pennsylvania, USA: Carnegie Mellon University. Department of Electrical and Computer Engineering, 2004.

[3] FELEQYHAZI M, HUBAUX J P, BUTTYAN L. Nash equilibria of packet forwarding strategies in wireless ad hoc networks [J]. IEEE Trans on Mobile Computing, 2006, 5(5): 463-476.

[4] MARTI S, GIULI T, LAI K, BAKER M. Mitigating routing misbehavior in mobile ad hoc networks [C]// Proceedings of MOBICOM 2000. New York, USA: ACM, 2000: 255-265.

[5] MARIAS G F, GEORGIADIS P, FLITZANIS D, et al. Cooperation enforcement schemes for MANETs: a survey [J]. Wireless Communications and Mobile Computing, 2006, 6(3): 319-332.

[6] YOO Y, AGRAWAL D P. Why does it pay to be selfish in a MANET? [J]. IEEE Wireless Communications, 2006, 13(6): 87-97.

[7] WANG Yongwei, SINGHAL M. On improving the efficiency of truthful routing in MANETs with selfish nodes[J]. Pervasive and Mobile Computing, 2007, 3(5): 537-559.

[8] EIDENBENZ S, RESTA G. The COMMIT protocol for truthful and cost-efficient routing in ad hoc networks with selfish nodes [J]. IEEE Trans on Mobile Computing, 2008, 7(1): 19-33.

[9] 郭建立，吴智博，董剑，等. 基于机制设计理论的自组网节点合作协议 [J]. 计算机学报，2009，32(3)：483-492.
GUO Jianli, WU Zhibo, DONG Jian, et al. A cooperation protocol for ad hoc networks with selfish nodes based on mechanism design [J]. Chinese Journal of Computers, 2009, 32(3): 483-492.

[10] BUCHEGGER S, BOUDEC J Y L. Self-policing mobile ad hoc networks by reputation systems [J]. IEEE

Communications Magazine, 2005, 43(7): 101-107.

[11] LIU K J, DENG J, VARSHNEY P K, et al. An acknowledgment-based approach for the detection of routing misbehavior in MANETs [J]. IEEE Trans on Mobile Computing, 2007, 6(5): 536-550.

[12] BUCHEGGER S, MUNDINGER J, BOUDEC J Y L. Reputation systems for self-organized networks [J]. IEEE Technology and Society Magazine, 2008,27(1): 41-47.

[13] HU J Y. Cooperation in mobile ad hoc networks [EB/OL]. [2007-11-12]. http://www.cs.fsu.edu/research/reports/TR-050111.pdf.

[14] BUTTYAN L, HUBAUX J P. Security and cooperation in wireless networks[M]. Cambridge, UK: Cambridge University Press, 2007.

(编辑 苗凌)