

Ad Hoc 网络 Q 学习稳定蚁群路由算法

王庆文^{1,2}, 史浩山², 戚 茜³

(1. 第二炮兵工程大学 空间工程系, 710025 西安; 2. 西北工业大学 电子信息学院, 710129 西安;
3. 西北工业大学 航海学院, 710072 西安)

摘 要: 针对 Ad Hoc 网络路由协议存在的对动态拓扑适应性差和链路不稳定问题, 提出了一种 Q 学习稳定蚁群路由算法 (SACRQ), 该算法综合了蚁群优化和 Q 学习算法的思想, 将信息素映射为 Q 学习算法的 Q 值, 增强节点对动态环境的学习能力. 在路由选择方面, 使用自适应伪随机比率选择下一跳节点, 避免算法陷入局部最优或是停滞; 提出了新的链路稳定度来衡量链路的鲁棒性, 结合鲁棒性和信息素强度两种因素选择下一跳链路. 该算法增加了链路的鲁棒性, 对 Ad Hoc 网络动态拓扑适应性强. 仿真结果表明, SACRQ 的路由发现数量、平均端到端延迟、冲突数量和每次路由发现吞吐量 4 种指标均优于 ARA 和 AODV.

关键词: Ad Hoc 网络; Q 学习; 蚁群; 路由算法; 鲁棒性

中图分类号: TP393

文献标志码: A

文章编号: 0367-6234(2012)07-0120-06

A stable ant colony routing algorithm based on Q-learning for Ad Hoc Networks

WANG Qing-wen^{1,2}, SHI Hao-shan², QI Qian³

(1. Dept. of Aerospace Engineering, The Second Artillery Engineering University, 710025 Xi'an, China;
2. School of Electronic Engineering, Northwestern Polytechnical University, 710129 Xi'an, China;
3. School of Marine, Northwestern Polytechnical University, 710072 Xi'an, China)

Abstract: To solve the problem of poor flexibility and frequent route breakage caused by dynamic topology in Ad Hoc network routing protocols, a stable ant colony routing algorithm based on Q-learning (SACRQ) is proposed, which synthesizes the Ant Colony Optimization and the Q-learning algorithm. The pheromone level is equal to the Q value to enhance the learning ability of nodes. To avoid local peak, SARCQ applies an adaptive pseudo random proportional action choice rule to select the next hop. A new robustness of the links metric is presented to calculate the probability of the route selection together with the pheromone level. The algorithm enhances the stability of the links and demonstrates high flexibility to the dynamic topology of the network. Simulation results show that SACRQ achieves better performance in terms of the number of the route discovery, the average end-to-end delay, the number of collisions and the average throughput per route discovery, which is respectively compared with the ARA and AODV.

Key words: Ad Hoc networks; Q-learning; ant colony optimization; routing algorithm; robustness

自组织网络 (Ad Hoc) 是一组带有无线收发装置移动节点组成的多跳的临时性自治系统^[1].

Ad Hoc 网络具有带宽受限、动态拓扑、多跳路由和能量受限^[2-4]等特点. 路由协议是 Ad Hoc 网络当前研究的热点之一, 典型的路由协议如 AODV、DSR、DSDV 和 OLSR 等, 对网络拓扑变化的适应性差. 蚁群算法以其正反馈、分布式计算和启发性搜索等特点, 被很多学者应用到 Ad Hoc 网络中来提高网络性能. 文献[5]提出了一种蚁群路由算法, 依据信息素强度来选择路由, 该算法实现简单, 但是并不能适应 Ad Hoc 网络拓扑的动态变

收稿日期: 2011-02-28.

基金项目: 教育部博士点基金资助项目 (20050699037);
国家自然科学基金资助项目 (60472074).

作者简介: 王庆文 (1982—), 男, 博士;
史浩山 (1946—), 男, 教授, 博士生导师.

通信作者: 王庆文, wqw013890@163.com.

化. 文献[6]针对 Ad Hoc 网络能量受限等特点, 提出了一种简单的蚁群路由算法, 在路由发现阶段, 只允许邻居节点中的一个转播路由发现分组, 减少了路由消耗, 但是带来了较大的延迟. 文献[7]提出了一种基于链路稳定的蚁群路由算法, 选择稳定的链路进行分组的发送, 提高了发包率的同时减少了通信开销, 但是该算法的路由发现时间较长. 文献[8]提出了一种容错的蚁群路由算法, 避免有错误倾向的节点转发分组, 提高了发包率, 但是该算法延长了节点处理分组的时间. 文献[9]提出了一种 QoS 蚁群路由算法, 将节点的传输延迟、丢包率等因素综合起来进行路由选择, 提高了发包率, 但是增加了路由开销. 文献[10]根据目的节点的地理位置信息确定路径信息素的大小, 降低了路由发现时间和蚂蚁分组的数目, 但是该算法没有考虑路径的稳定性.

上述蚁群算法对动态网络拓扑适应性差, 算法容易陷入局部最优或是停滞, 从而限制了算法性能的优化, 本文提出了一种 Ad Hoc 网络 Q 学习稳定蚁群路由算法 (SACRQ). SACRQ 综合了蚁群算法和 Q 学习算法的思想, 增加了节点对动态环境的适应性. 在路由选择方面, 使用自适应伪随机比率选, 综合考虑信息素和链路稳定度两种因素, 避免出现局部最优或是停滞现象, 增加了链路的稳定性. 因而该算法对网络动态拓扑有更好的适应性. 仿真结果表明, SACRQ 减少了路由发现数量、平均端对端延迟和冲突数量, 提高了每次路由发现的吞吐量.

1 SACRQ 算法设计

1.1 信息素更新

SACRQ 将 Q 学习算法引入到传统的蚁群算法中, 充分利用学习机制, 强化最优机制的反馈, 从而增强对动态环境的学习能力. Q 学习是一种典型的强化学习算法^[11-12]. Q 学习算法不需要对所处的动态环境建模, 能接受并处理动态环境中的不完全、不确定信息, 产生最佳策略, 选择最优行动. Q 学习算法的本质是通过学习每个状态-动作的评价值 $Q(s, a)$, Agent 在每个时 t , 根据环境状态 s 选择动作 a , 观察收益 r_t 和新状态 s_{t+1} , 并按下式更新 Q 值:

$$Q_{t+1}(s, a) = (1 - \alpha)Q_t(s, a) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a)) \quad (1)$$

式中: α 学习因子, 且 $0 < \alpha < 1$; r_{t+1} 是 $t+1$ 时刻的奖赏; γ 是折扣因子, 且 $0 < \gamma < 1$.

SACRQ 的状态转移概率反映了蚁群算法与

Q 学习算法之间的内在联系. 其中, 信息素相当于 Q 学习算法中的 Q 值, 表示学习所得到的经验.

令 $\tau_{ij}(t)$ 表示 t 时刻链路 (i, j) 上的信息素, 同时也是 Q 学习算法的 Q 值. 网络初始时, 各路径上的信息素相等 $\tau_{ij}(0) = C$, C 为一个常数. 信息素在固定时间间隔 Δt 内, 按照如下规则进行更新:

$$\tau_{ij}(t) = (1 - \sigma)\tau_{ij}(t) + \sigma(\Delta\tau_{ij}(t) + \gamma \max(\tau_{jk}(t))) \quad (2)$$

式中: $k \in \phi_j$, ϕ_j 是节点 j 的一跳邻居集, σ 为挥发系数, $\Delta\tau_{ij}(t)$ 为 Δt 时间内的信息素增量, 可由公式(3) 计算:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^n \Delta\tau_{ij}^k(t) \quad (3)$$

式中: $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在本次循环中留在路径上的信息素强度, $\Delta\tau_{ij}^k(t)$ 表示本次循环中路径上的信息素强度的增量, 且

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q, & \text{若第 } k \text{ 只蚂蚁经过链路 } (i, j); \\ 0, & \text{其他.} \end{cases} \quad (4)$$

1.2 链路稳定度

SACRQ 结合节点的移动速度、移动方向、节点的通信半径和节点之间的距离等因素, 提出了新的链路稳定度量来衡量链路的鲁棒性.

令 v_i, v_j 分别表示相邻节点 i, j 的移动速度; θ_{ij} 表示两个节点运动方向的夹角; D_{ij} 表示两个节点的距离; R 表示节点的传输半径, 则节点 i, j 之间链路稳定度定义如下:

$$LS_{ij} = \left(1 - \frac{\min(v_{\max}, |v_i - v_j|)}{v_{\max}}\right) \times \frac{1}{2}(\cos\theta_{ij} + 1) \times \left(1 - \frac{D_{ij}}{R}\right) \quad (5)$$

令 θ_i, θ_j ($0 \leq \theta_i, \theta_j < 2\pi$) 为相邻节点 i, j 的移动方向, 两个节点运动方向的角度差 θ_{ij} 的余弦值可以表示为

$$\cos\theta_{ij} = \cos\theta_i \times \cos\theta_j + \sin\theta_i \times \sin\theta_j \quad (6)$$

令 $(x_i, y_i), (x_j, y_j)$ 为相邻节点 i, j 的地理位置坐标, 那么两个节点之间的距离 D_{ij} 可以计算为

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (7)$$

对于两个能够相互通信的节点 i, j , 显然有 $0 < D_{ij} \leq R$.

LS_{ij} 的值在 0 与 1 之间, LS_{ij} 值越大, 表明链路越稳定. 当节点之间满足下面 3 个条件时, LS_{ij} 的值较大, 链路的鲁棒性较高: 1) 节点 i 和 j 的移动方向相似; 2) 节点 i 和 j 相互靠近; 3) 节点 i 和 j 的地理位置很近.

1.3 选择策略

SACRQ 的路由选择策略使用自适应随机比例规则,解决了传统的蚁群算法容易陷入局部僵局的问题.选择下一跳邻居节点时,既考虑到链路的信息素的强度,又考虑到链路的稳定性.引入一个参数 ν ,决定了利用先验知识和探索新路径之间的相对重要性.蚂蚁 k 选择下一跳节点的概率依据公式(8)和(9):

$$P_{ij}^k(t) = \begin{cases} \frac{\psi_{ij}(t)}{\sum_{s \in \phi_i} \psi_{is}(t)}, & j \in \phi_i; \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

$$s = \begin{cases} \max\{\psi_{ij}(t)\}, & j \in \phi_i, q \leq \nu; \\ S, & j \in \phi_i, q > \nu. \end{cases} \quad (9)$$

式中: q 是在 $[0,1]$ 区间均匀分布的随机数, ϕ_i 是节点 i 的一跳邻居集, ν 是一个参数 ($0 \leq \nu \leq 1$), S 为根据公式(1)给出的概率分布选出的一个邻居. $\psi_{ij}(t)$ 是信息素和链路稳定度的函数,表示如下:

$$\psi_{ij}(t) = (\tau_{ij}(t) + 1)^F \times e^{LS_{ij}}. \quad (10)$$

式中: $\tau_{ij}(t)$ 为信息素强度, F 为调节因子, LS_{ij} 为节点 i 和 j 之间的链路稳定度.收敛因子 F 的值与数据集中到一条路径或路径集密切相关.如果 F 值太大,则选择概率时,对信息素的依赖大,达不到考虑其他因素的目的.如果 F 值太小,则信息素的影响偏弱,减弱了蚁群算法收敛快的优点.

2 SACRQ 算法实现

2.1 链路稳定性维护

SACRQ 需要节点周期性的发送 Hello 分组,Hello 分组中携带节点的地理位置信息和移动信息,获得局部的链路的稳定信息.节点均维护一个路由表和一个邻居列表.当节点 i 接收到来自邻居节点 j 的 Hello 分组后,根据公式(5)计算出节点链路稳定度,更新自己的邻居列表,将邻居节点的链路稳定度和信息素的值添加到邻居列表中.

2.2 路由发现

源节点 s 想要发送数据分组到目的节点 d ,但是源节点没有到目的路由,那么源节点释放前向蚂蚁 FANT, FANT 的任务就是寻找到目的节点的路径并且更新经过路径上的信息素.

FANT 包含如下字段:

蚂蚁 ID、源节点 ID、目的节点 ID、上一跳节点的地理位置、上一跳节点的速度、访问过的节点列表.

步骤 1 初始化信息素,设置各链路信息素的初始值.

步骤 2 源节点产生 FANT 后,广播给其邻居节点.

步骤 3 中间节点接收到 FANT 后,如果接收到的分组非重复,更新 FANT 的相关字段,依据公式(2)在固定时间间隔更新链路信息素.在路由表中添加或更新到源节点的路由表项.

步骤 4 中间节点依据公式(8)和(9)选择下一跳的邻居节点,并将自己的 ID 加入到 FANT 的访问过节点列表中,转发 FANT.

步骤 5 当目的节点 d 接收到 FANT 后,释放出后向蚂蚁 BANT.为了建立多条路径,所有的 FANT 都会被目的节点接收. BANT 沿着 FANT 的路径返回到源节点,并根据公式(2)更新路径上的信息素. BANT 和 FANT 的结构和功能基本相同,不同之处在于 BANT 不会被广播.如果 BANT 不能到达下一跳节点,那么这个 BANT 将被丢弃.

步骤 6 源节点接收到 BANT 后,开始发送数据分组.

2.3 路由修复

链路断开通常是由于节点关闭通信,无线通信覆盖失败或是拥塞引起的大量的冲突.在移动自组织网络中,链路断开现象很频繁,因而路由修复过程必须以少量路由消耗的情况下快速执行. SACRQ 采用多路径机制,只有当到目的节点的全部路径都断开时,才进行路由修复. SACRQ 的路由修复过程的步骤如下:

SACRQ 在路由修复过程中,释放两种蚂蚁:前向修复蚂蚁 R_FANT 和修复应答蚂蚁 R_BANT. R_FANT 和 R_BANT 的 TTL 值为 2. R_FANT 以广播的方式传播,当中间节点有到目的节点的路由时,给路由修复发起节点发送 R_BANT 分组, R_BANT 以单播的方式传播,返回给发起路由修复的源节点.举例如下:假设 S-A-C-E-F-D 的一条传输路径,节点 A 发现链路断开,开始路由修复过程,首先设置定时器 RRT,如果在定时器时间内, R_FANT 到达了节点 E,该节点拥有到目的节点 D 的路,节点 E 就单播 R_BANT 给节点 A,告知节点 A 路由修复已成功.如果定时器到期了,路由修复没有成功,节点 A 就将链路断开消息发送给源节点,通知源节点 S 重启路由发现过程.

3 算法复杂度分析

3.1 SACRQ 算法复杂度分析

定理 1 SACRQ 算法的时间复杂度为 $O(H)$, H 为网络中路径跳数值的最大值.

证明 SACRQ 算法成功建立起路由需要路由请求分组和路由回复分组经过路径. 设分组每经过一跳链路花费的时间开销是一个单位 $O(1)$, 因此路由请求分组和路由回复分组经过 H 跳的时间复杂度都为 $O(H)$, 所以 SACRQ 的时间复杂度为 $O(H)$.

定理 2 SACRQ 算法的消息复杂度为 $O(\Delta(H+n))$, H 为网络中路径跳数值的最大值, n 为网络中的节点总数, Δ 为节点的度即节点的邻居节点数量.

证明 SACRQ 路由发现过程中, 源节点有 Δ 个邻居节点处理 $FANT$, Δ 个邻居节点最多形成到目的节点的 Δ 条路径, 每条路径上至多有 $H-1$ 个节点对分组进行处理. 因此, SACRQ 路由发现部分的消息复杂度为 $O(\Delta H)$; SACRQ 需要周期性发送 Hello 分组, 因此每个节点需要接收 Δ 个邻居的 Hello 分组, 所以 n 个节点接收 Hello 分组的复杂度为 $O(\Delta n)$. 因此可得, SACRQ 算法的消息复杂度为 $O(\Delta(H+n))$.

3.2 3 种算法复杂度比较

同定理 1 和定理 2, 可得 ARA 和 AODV 的算法复杂度. SACRQ、ARA 和 AODV 3 种算法的复杂度如表 1 所示.

表 1 3 种算法的复杂度

算法	时间复杂度	消息复杂度
SACRQ	$O(H)$	$O(\Delta(H+n))$
ARA	$O(H)$	$O(\Delta H)$
AODV	$O(H)$	$O(H^2+n\Delta)$

通过表 1 可得:

- 1) SACRQ、ARA 和 AODV 3 种算法的时间复杂度一样.
- 2) SACRQ 的消息复杂度大于 ARA 算法, 是由于 SACRQ 需要周期性发送 Hello 分组维护稳定性.
- 3) SACRQ 与 AODV 的消息复杂度的优劣要根据具体条件而定: 当 $\Delta > H$ 时, SACRQ 的消息复杂度大于 AODV. 当 $\Delta < H$ 时, SACRQ 的消息复杂度小于 AODV. 当 $\Delta = H$ 时, SACRQ 的消息复杂度等于 AODV.

4 仿真分析

4.1 仿真环境

仿真环境是带有 CMU 无线扩展的 NS-2 (version2.31), 仿真场景为 $1\,000\text{ m} \times 1\,000\text{ m}$, 节点的传输半径为 200 m , 网络带宽为 2 M bit/s , 仿真时间为 60 s , 节点的暂停时间为 0 s . 仿真协议:

路由层(1)SACRQ(2)ARA(3)AODV; MAC 层为 IEEE 802.11. 数据流选用与文献[6]一样的 TX-CBR, 其与 CBR 最大的不同在于, 目的节点接收到数据分组后, 将分组再次发送给源节点. SACRQ 算法的各个参数取值为: $F=1$, $\nu=0.3$, $\sigma=0.4$, $C=30$, $Q=0.5$, $\Delta t=1$. 仿真节点数量为 104, 其中 4 个节点为目的节点, 在整个仿真过程中处于静止状态, 地理位置坐标分别为 $(200, 200)$ 、 $(200, 800)$ 、 $(800, 800)$ 和 $(800, 200)$, 其他 100 个节点随机移动. 源节点发送的分组大小为 512 比特, 每秒发送 4 个分组. 仿真是在没有噪声干扰的环境下进行的, 仿真实验结果均为 10 次实验的平均值. 节点最大移动速度为 6 m/s , 仿真时间为 19.26136 s 的网络拓扑如图 1 所示, 其中标注 SRC 的节点为源节点, 标注 DST 的节点为目的节点, 黑色圆代表节点正发送无线电波.

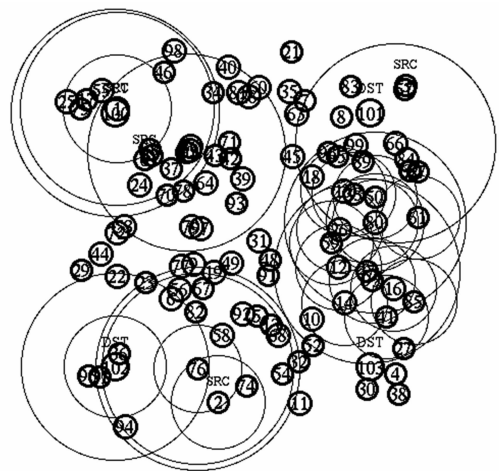


图 1 仿真时间为 19.26136 s 时的网络拓扑

4.2 评价指标

1) 路由发现数量 R_{FT}

$$R_{FT} = \sum_{i=1}^s S_{RQ}(i). \quad (11)$$

式中: s 表示网络中源节点总数, $S_{RQ}(i)$ 表示节点 i 在仿真时间内发送的非重复的前向蚂蚁或是路由请求分组的总数.

2) 平均端对端延迟 $Avdelay$

$$Avdelay = \frac{1}{N} \sum_{i=0}^N (R_{time}(i) - S_{time}(i)). \quad (12)$$

式中: N 表示到达目的节点的数据分组数量, $R_{time}(i)$ 表示第 i 个分组到达目的节点的时间, $S_{time}(i)$ 表示第 i 个分组的发送时间.

3) 冲突数量 N_{col}

$$N_{col} = \sum_{i=1}^n C(i). \quad (13)$$

式中: n 表示网络中节点总数, $C(i)$ 代表节点 i 在

仿真过程中 MAC 层产生的冲突数量.

4) 每次路由发现吞吐量 F_{th}

$$F_{th} = \frac{8}{R_{FT}} \sum_{i=1}^N R_{bytes}(i). \quad (14)$$

式中: N 表示到达目的节点的数据分组数量, $R_{bytes}(i)$ 代表分组 i 的字节数, R_{FT} 表示网络中的路由发现数量.

4.3 仿真结果

图 2 显示了 3 种算法的路由发现数量与节点移动速度的关系. 从图中可以看出, 3 种算法的路由发现数量随着节点移动速度增加而增加, SACRQ 路由发现数量明显小于 ARA 和 AODV, 这种趋势随着节点的移动速度的增加而增加. 原因是, 在路由建立过程中, SACRQ 采用强化学习机制增加了对动态拓扑的适应度, 选择稳定的链路进行数据分组的转发, 在节点高速移动的情况下, 减少了链路断开的次数. 另一方面, SACRQ 采用多路径机制, 只有当全部路径断开时并且路由修复不成功, 才重启路由发现, 这样, 就大大减少了路由发现次数.

图 3 显示了 3 种算法的平均端对端延迟与节点移动速度的关系. 从图中可以看出, 3 种算法的平均端对端延迟随着节点移动速度增加而增加, SACRQ 平均端对端延迟明显小于 ARA 和 AODV, 这种趋势随着节点的移动速度的增加而增加. SACRQ 选择稳定的链路进行数据分组的转发, 减少了因链路断开和路由修复带来的延迟. 当节点的最大移动速度是 6 m/s 时, ARA 的延迟性能比 AODV 差是网络拓扑的随机性造成的.

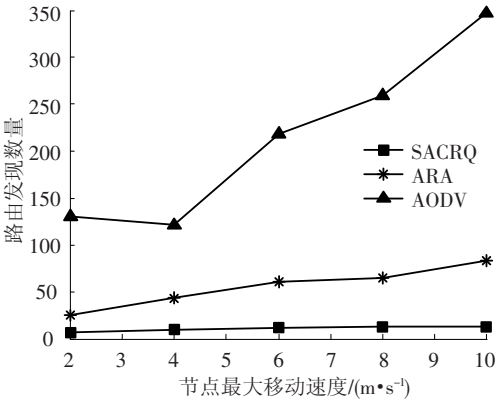


图 2 路由发现数量

图 4 显示了 3 种算法产生的冲突数量与节点移动速度的关系. 从图中可以看出, 3 种算法的产生的冲突数量随着节点移动速度增加而增加, SACRQ 产生的冲突数量明显小于 ARA 和 AODV, 这种趋势随着节点的移动速度的增加而增加. 原因是, SACRQ 在路由建立过程中, 依概率

选择稳定邻节点进行分组的转发, 减少了因链路断开而导致的路由修复和路由重启的次数, 也就减少了产生大量的控制分组的数量, 因而减少了产生冲突的数量.

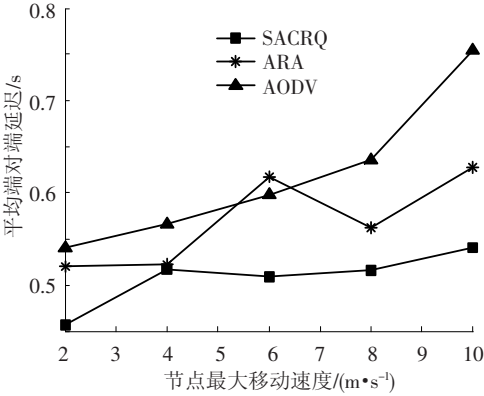


图 3 平均端对端延迟

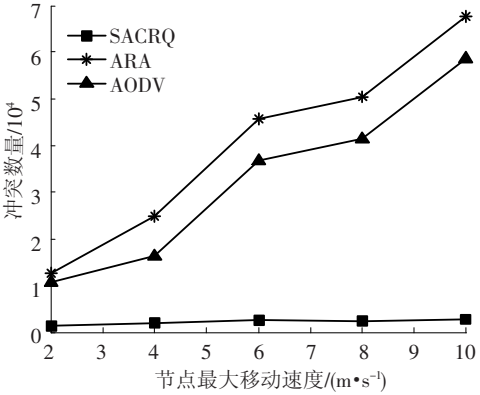


图 4 冲突数量

图 5 显示了 3 种算法的每次路由发现的吞吐量与节点移动速度的关系. 从图中可以看出, 3 种算法的每次路由发现吞吐量随着节点移动速度增加而减少, SACRQ 每次路由发现吞吐量明显多于 ARA 和 AODV, 这种趋势随着节点的移动速度的增加而区域平缓. 原因是, SACRQ 在路由建立过程中, 利用强化学习机制增加了对网络拓扑的适应度, 选择稳定链路进行通信, 减少了链路断开的概率, 每次路由发现建立的路由质量好、效率高, 所以每次路由发现获得的吞吐量明显好于 ARA 和 AODV.

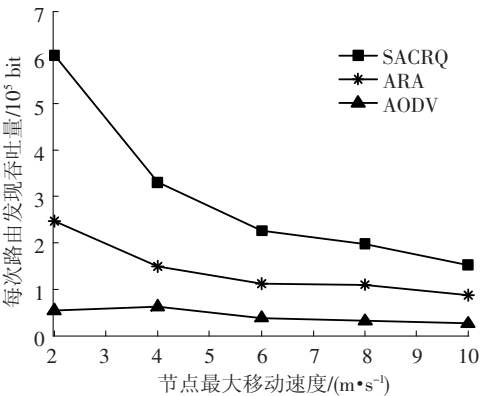


图 5 每次路由发现的吞吐量

5 结 语

本文提出了一种 Ad Hoc 网络 Q 学习稳定蚁群路由算法(SACRQ). SACRQ 将 Q 学习算法融入蚁群算法中,将蚁群算法的信息素值映射为 Q 学习算法的 Q 值,增强节点对动态环境的学习能力. 提出了一种新的链路稳定度定义,在路由选择策略方面,使用自适应伪随机比率,避免算法陷入局部最优或是停滞;综合考虑信息素强度和链路稳定度两种因素,增加了对网络动态拓扑的适应性. NS-2 仿真结果表明,SACRQ 的路由发现数量、平均端到端延迟、冲突数量和每次路由发现吞吐量四种指标均明显优于 ARA 和 AODV. 下一步的工作是将能量均衡的思想引入到算法中,进一步提高算法的性能.

参考文献:

- [1] HANASHI A M, SIDDIQUE A, AWAN I, *et al.* Performance evaluation of dynamic probabilistic broadcasting for flooding in mobile ad hoc networks[J]. *Simulation Modelling Practice and Theory*, 2009, 17 (2): 364-375.
- [2] GENÇ Z, ÖZKASAP O. Epidemic-based reliable and adaptive multicast for MANETs[C]//*Proceedings of the Wireless Communications and Networking Conference*, Hong Kong, China, IEEE, 2007:4395-4400.
- [3] RAJU G V, REDDY T B, MURTHY C S R. On supporting robust voice multicasting over Ad Hoc wireless Networks [C]//*Proceedings of the Communications*, Glasgow, Scotland, IEEE, 2007:1636-1641.
- [4] 李新, 孙丹丹, 周立刚, 等. 基于时延和能耗的 Ad Hoc 网络路由选择算法[J]. *哈尔滨工业大学学报*, 2008, 40 (5): 763-766.
- [5] GUNES M, SORGES U, BOUAZIZI I. ARA-the ant-colony based routing algorithm for MANETs[C]//*Proceedings of the Parallel Processing Workshops*. Vancouver, British Columbia, Canada, IEEE, 2002:79-85.
- [6] CORREIA F, VAZÃO T. Simple ant routing algorithm strategies for a (Multipurpose) MANET model[J]. *Ad Hoc Networks*, 2010, 8 (8): 810-823.
- [7] KADONO D, IZUMI T, OOSHITA F, *et al.* An ant colony optimization routing based on robustness for ad hoc networks with GPSs[J]. *Ad Hoc Networks*, 2010, 8 (1): 63-76.
- [8] MISRA S, DHURANDHER S K, OBAIDAT M S, *et al.* A low-overhead fault-tolerant routing algorithm for mobile ad hoc networks: A scheme and its simulation analysis[J]. *Simulation Modelling Practice and Theory*, 2010, 18 (5): 637-649.
- [9] COBO L, QUINTERO A, PIERRE S. Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics[J]. *Computer Networks*, 2010, 54 (17): 2991-3010.
- [10] KAMALI S, OPATRYN J. A position based ant colony routing algorithm for mobile Ad-hoc networks[C]//*Proceedings of the Wireless and Mobile Communications*. Washington, DC, USA: IEEE, 2007:21-21.
- [11] JUANG Chia-feng, LU Chun-ming. Ant colony optimization incorporated with fuzzy Q-learning for reinforcement fuzzy control[J]. *Systems, Man and Cybernetics, Part A: Systems and Humans*, IEEE Transactions on, 2009, 39 (3): 597-608.
- [12] OUZECKI D, JEVTIC D. Reinforcement learning as adaptive network routing of mobile agents[C]//*Proceedings of the MIPRO*. Opatija, Croatia: IEEE, 2010:479-484.

(编辑 苗秀芝)