

# 一种基于跨层设计和蚁群优化的自组网 负载均衡路由协议

郑相全<sup>1,2</sup>, 郭 伟<sup>1</sup>, 葛利嘉<sup>2</sup>, 刘仁婷<sup>1</sup>

(1. 电子科技大学通信抗干扰技术国家重点实验室, 四川成都 610054)

2. 重庆通信学院数字通信与信号处理重点实验室, 重庆 400035)

**摘 要:** 针对大部分现有替代路径共同存在的替代路径老化和构建效率问题, 本文提出了一种基于跨层设计和蚁群优化的负载均衡路由协议 (CALRA), 利用蚁群优化算法特有的信息素挥发方法实现对替代路径的老化问题, 将蚁群优化和跨层优化方法结合起来解决自组网中的负载均衡问题, 通过双向逐跳更新的方式较好的解决了替代路径构建效率问题, 并将蚂蚁在所经过的各中间节点为路由表带来的信息素增量映射为蚂蚁离开源节点的距离、移动过程中所遇到的节点拥塞程度、节点当前信息素浓度和节点移动速度等各协议层的统计信息的函数, 通过对各种信息所对应的参数赋予不同加权值的方法对概率路由表进行控制, 改善了自组网中现有基于蚁群优化的路由协议中普遍存在的拥塞问题、捷径问题、收敛速度问题和引入的路由开销问题。仿真表明, CALRA 在分组成功递交率、路由开销、端到端平均时延等方面具有优良性能, 能很好地实现网络中的业务负载均衡。

**关键词:** 跨层设计; 蚁群优化; 负载均衡路由协议; 替代路径路由

**中图分类号:** TN 915 **文献标识码:** A **文章编号:** 0372-2112 (2006) 07-1199-10

## A Cross-Layer Design and Ant-Colony Optimization Based Load-Balancing Routing Protocol for Ad Hoc Networks (CALRA)

ZHENG Xiang-quan<sup>1,2</sup>, GUO Wei<sup>1</sup>, GE Lijia<sup>2</sup>, LIU Ren-ting<sup>1</sup>

(1 National Key Lab of Communication, UEST of China, Chengdu, Sichuan 610054 China; 2 Key Lab of Digital

Communication & Signal Processing, Chongqing Communication College, Chongqing 400035, China)

**Abstract** In order to reduce the flooding overhead of alternate path construction most existed alternate path routing protocols build their alternate path routes (APR) during route discover period which leads to two limitations. As the topology of the ad hoc network evolves, links will be created and destroyed, resulting in route reduction or route failure. Additionally, changes in network traffic (including the traffic introduced by the source itself) will have a significant effect on traffic-dependent route metrics. This motivates the need for a route-probing mechanism that could periodically re-assess the status of the APR route set. The results of those route probes could then be used to both construct a new APR route set and also adjust the route alternation policies. The alternate path route set is unidirectionally constructed from the source to the destination by flooding, which is inefficiency for bidirectional communication. In this paper, we present a cross-layer design and ant-colony optimization based load-balancing routing protocol for ad hoc networks (CALRA). In CALRA, ants deposit simulated pheromones as a function of multiple parameters corresponding to the information collected by each layer of each node visited, such as the distance from their source nodes, the congestion degree of the visited nodes, the current pheromones the nodes possess, the velocity the nodes move and so on, and provide the information to the visiting nodes to update their pheromone tables by endowing the above different parameters corresponding to different information with different weight values, which provides a new method to improve the congestion

收稿日期: 2004-12-07; 修回日期: 2006-04-12

基金项目: 国家自然科学基金 (No. 60472052, No. 10577007); 通信抗干扰技术国家级重点实验室基金 (No. 5143402010SZ S04); 重庆邮电大学移动通信技术重庆市信息产业部重点实验室开放课题基金和教育部分“新世纪优秀人才支持计划”项目

problem, the shortcut problem, the convergence rate and the heavy overheads commonly existed in existing ant-based routing protocols for ad hoc networks. With the pheromones evaporation mechanism, CALRA can age and re-assess the APR set. The performance of the algorithm is measured by the packet delivery rate, the goodput ratio (routing overhead), and the end-to-end delay. Simulation results show that CALRA performs well in decreasing the route overhead and balancing traffic load, as well as increasing the packet delivery rate, etc.

**Key words** cross-layer design; ant-colony optimization; load-balancing routing protocol; alternate path route

## 1 引言

自组网 (ad hoc networks)<sup>[1]</sup> 是一种移动通信和计算机网络相结合的无中心分布控制网络, 整个网络没有固定基础设施, 具有优良的自主构建、自主组织和自主管理等特性, 提供了廉价而快速部署网络的可能, 可以在不能利用或不便利用现有网络基础设施的情况下提供一种通信支撑环境和广泛应用于国防战备、灾难救助、偏远地区等无法得到有线网络支持或某些只是临时需要通信的环境。路由协议是自组网研究的重点和技术难点之一, 其设计需要考虑网络的移动性、能量受限、带宽有限和较高的误比特率等特征。在处理网络快速变化带来的影响的同时尽可能对网络的多个服务质量 (QoS) 参数进行优化, 其中最大程度减少控制分组传输的数量是非常关键的。替代路径路由协议通过建立替代路径的方法改善现有按需路由协议对节点移动的鲁棒性能, 以降低路由洪泛发生的频繁程度和洪泛开销, 但其研究大多集中在替代路径的构建和替代路径的切换时机问题上, 没有考虑替代路径的老化机制, 难以对路径质量进行评估。通过一次洪泛请求获得尽可能多的路由也是替代路径构建的一个考虑因素, 现有替代路径协议通常采用周期性交换 HELLO 消息的方式使每个节点维护  $k$  跳范围内的邻居信息表, 从而获得更多邻居范围内的链路状态。但是, 该方法也只是使从源节点到目的节点的单向替代路径数量增多, 无法实现高效的双向替代路径构建, 在双向通信的场合显得效率不高。现有替代路径路由协议对 QoS 的支持也比较少。另外, 在拓扑频繁变化的自组网中采用基于集中控制的路由机制会导致网络开销的增加和网络可靠性、可扩展性的降低, 会使控制中心成为网络业务流传输的瓶颈。基于蚁群优化方法的路由协议由于其优良的分布式特性, 能为网络提供大量的冗余路由和可靠连接, 可使系统具有抗控制中心故障和同时优化多个 QoS 参数的能力, 能自动配置和自动适应自组网的动态变化环境。当然, 蚁群优化算法是受到自然界中蚂蚁对食物的搜索方法的启发而产生的, 算法的处理开销比较大, 在将其引入自组网中的协议设计时必须考虑自组网特殊的网络应用环境。不过, 在器件制造水平从处理速度、存储容量到器件体积和功耗等方面都有了质的飞跃的今天, 相对节点的处理开销而言, 最大限度地降低链路通信开销是当今自组网设计中一个更为重要的问题。因此, 有必要针对自组网特殊的网络局限, 采用基于跨层设计<sup>[2-4]</sup> 的组网

方法, 综合考虑各协议层之间的信息进行路由协议设计以实现网络业务流负载的均衡传输, 以便在减少链路开销的同时充分利用网络资源, 使网络保持连续、高效、稳定的运行, 网络综合性能达到优化。

评价网络性能的指标主要是吞吐率和网络传输时延<sup>[5]</sup>, 负载均衡技术同时具有提升这两方面的能力, 能够在提高系统吞吐率的基础上, 尽可能减少网络的端到端时延<sup>[6]</sup>。针对自组网特殊的应用环境, 为了充分利用电子技术快速发展为网络协议设计带来的便利, 本文将跨层设计思想和蚁群优化方法结合起来, 提出了一种分布式负载均衡概率路由算法 (CALRA), 利用蚁群优化算法特有的信息素挥发方式对节点所维护的替代路径进行老化处理, 通过每次路由发现和维护过程对信息素概率路由表进行双向和逐跳更新的方式提高替代路径构建效率, 使每个中间节点双向维护多条替代路径, 协议还通过概率路由维护交叉替代路径的方式为源和目的节点之间提供更多的路径冗余。重要的是, 建立多条替代路径并不会比只产生一条路由传输更多的控制消息, 不会增加额外的传输开销。为了将网络中链路的通信开销尽可能转换为节点的计算处理开销, 算法综合考虑了各协议层之间的信息共享, 将蚂蚁在所经过的各中间节点为路由表带来的信息素增量映射为蚂蚁离开源节点的距离、移动过程中所遇到的节点拥塞程度、节点当前信息素浓度和节点移动速度等各协议层的统计信息的函数, 通过对各种信息所对应的参数赋予不同加权值的方法对概率路由表进行控制, 较好的解决了自组网中现有基于蚁群优化的路由协议中普遍存在的拥塞问题、捷径问题、收敛速度问题和引入的路由开销问题, 并最终实现网络业务流负载在网络中的均衡传输。

## 2 蚁群优化算法研究概述

蚁群优化方法 (Ant Colony Optimization, ACO) 是一种性能优良的启发式随机优化算法, 采用正反馈机制实现分布式全局优化, 通过信息素的不断更新达到最终收敛于最优路径上<sup>[7]</sup>, 可应用于目前的串行计算机和将来的并行计算中进行单目标和多目标的全局优化以及通信网中路由的组合优化求解, 其固有的并行计算特性有利于实现分散控制, 能够大大提高系统的可靠性、鲁棒性和增强通信网适应诸如传输故障、突发业务等意外事件的能力。算法无需进行大量的概率计算或建立复杂的数学模型来进行系统预测等, 并不给网络信令系统增加很多负担, 很容易

实现. 尤其当网络的规模很大、系统运行动态多变、难以以为系统建立一个较准确的数学模型进行集中控制时, 蚁群优化的优点更加明显.

最早在有线网络如 ABC<sup>[8]</sup> 中引入的基于蚁群优化的路由方案难以快速适应动态网络拓扑的自组网应用, 但其通过蚂蚁进行路由发现与维护的基本思想为后续基于蚁群优化的路由协议所继承. 为解决对动态网络拓扑的适应问题, GPS/AL<sup>[9]</sup> 和 AntAODV<sup>[10]</sup> 采用移动代理收集所经过的网络节点中对路由有用的信息并进行传输, 但 GPSAL 需要固定网络基础设施提供位置信息. ARA<sup>[11]</sup> 按需广播前向蚂蚁, 并允许数据包对现有路由进行增强以减少发送蚂蚁带来的附加开销, 但采用 S-ACO<sup>[12]</sup> 进行路由维护会使得 ARA 在应用于节点数较多的网络时路径变得更不稳定. MABR<sup>[13]</sup> 利用蚂蚁确定网络所有节点的位置和完成位置信息向网络中所有节点的传输, 并依靠位置信息按照主动路由的方法实现网络中业务流的传输. ANSI<sup>[14]</sup> 在有限范围内采用本地主动蚂蚁进行广播以便在节点的本地邻近范围内建立路由, 在没有可用路由时才在整个网络中采用全局蚂蚁进行洪泛以发现新路由. ARAMA<sup>[15]</sup> 对路由跳数和节点电池使用的公平性进行了优化, 但采用只向一个邻居节点转发前向蚂蚁的方法来控制路由开销, 将造成网络中业务流负载的不均衡分布和网络瓶颈的出现. PERA<sup>[16]</sup> 采用主动路由方式, 在网络拓扑变化剧烈的情况下将引入大量的附加开销. 可见, 自组网中现有的蚁群优化算法各有优缺点, 同时还有不少问题需要解决, 如: 对建立的多条路径进行老化, 通过信息素浓度的挥发以及相应的信息素概率表更新予以实现, 需要在设计信息素浓度的挥发和更新方法时综合考虑适合自组网环境的多种影响参数; 相交叉的替代路径具有更多的冗余, 可以通过逐跳选路的信息素概率表实现, 但需要解决环路问题; 信息素浓度表的更新方法大多采用单向更新, 已有的双向更新也只是针对源和目的节点更新节点的信息素表项, 没有对从源节点和目的节点到本节点的路径中所有中间节点对应的信息素表项进行逐跳更新, 信息素表项的更新效率有待进一步提高; 需要解决现有蚁群优化算法中普遍存在的拥塞问题、捷径问题和收敛速度问题.

3 CALRA 协议

3.1 跨层负载感知及负载状态判断

为了使网络层在进行路由选择时能实现负载均衡, 需要对网络中各节点的业务负载状态进行感知, 节点的业务负载状态可以通过 MAC 层的分组传输时延进行表征. 通过 MAC 层提供的与节点负载状态 (业务负载分布) 相关的分组传输时延信息, 使网络层能够在路

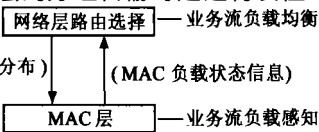


图 1 跨层负载感知与交互

由建立时对节点相对路径的状态进行判断, 并作为后续分组传输时路由选择的依据, 该方式反过来又将影响网络中业务负载的分布和 MAC 层的分组传输时延, 其跨层交互关系如图 1 所示.

具体实现时, 每个移动节点监视 MAC 层接口队列并对节点的平均分组传输时延进行计算. 第  $m$  个分组成功发送后节点  $i$  的平均分组传输时延值  $d_m^i$  为:  $d_m^i = (1 - w) \cdot d_{m-1}^i + w (s_m - a_m)$ ,  $m > 1$  且加权平滑参数  $w$  满足  $0 \leq w \leq 1$ ,  $a_m$ 、 $s_m$  分别为第  $m$  个分组到达节点  $i$  和被节点  $i$  成功发送时间. 从节点 1 到  $n$  的无环路径  $P(n) = \langle 1, \dots, i, \dots, n \rangle$  的路径总平均估计时延  $D_p(n)$  和节点平均估计时延  $\overline{D_p(n)}$  分别为:  $D_p(n) = \sum_{i=1}^n d^i$  ( $d^i$  为节点  $i$  在路由发现时的平均分组传输时延),  $\overline{D_p(n)} = \frac{D_p(n)}{n}$ . 设节点  $i$  的 MAC 层接口队列中当前缓存的分组数为  $B_p(i)$ , 节点  $i$  的 MAC 层接口队列缓存容量为  $B_c(i)$ , 则节点  $i$  的 MAC 层接口队列的当前占用率为  $\eta_i = B_p(i) / B_c(i)$ . 节点  $i$  的负载状态通过本节点平均分组传输时延与路径中节点的平均估计时延的可比性条件和节点 MAC 层接口队列缓存容量的占用百分比条件相结合进行判断, 前者反映本节点对业务负载的处理速度、处理能力和占用共享信道的公平性问题, 后者反映本节点的业务强度及本节点 MAC 层接口队列还可接纳的分组容量, 两者共同反映了节点对后续路由请求的接纳能力, 可表示为:

节点  $i$  相对路径  $p(n)$  处于捷径状态, 若  
 $(d^i < \overline{D_p(n)} (1 + \delta)) \cap (\eta_i < \alpha)$

节点  $i$  相对路径  $p(n)$  处于拥塞状态, 若  
 $(d^i \geq \overline{D_p(n)} (1 + \delta)) \cap (\eta_i > \beta)$ ,

其中,  $\delta > 0$  占用率门限  $\alpha, \beta \in (0, 1)$ .

节点  $i$  相对路径  $p(n)$  处于可接纳状态, 其它

CALRA 中使用的分组格式定义如下:

flag	seq/src	dest	path.delay	path.node	data
------	---------	------	------------	-----------	------

其中, flag 为分组类型标识, 用于区别更新触发分组和普通的数据分组 (ORPKT); seq/src 在 flag 为蚂蚁分组时表示分组的序列号 seq, 为数据分组时表示源节点的标识 src; dest 为分组的目的节点; path.delay 为更新触发分组所历经路径的总平均估计时延  $\overline{D_p(n)}$ ; path.node 为更新触发分组所遍历  $P(n)$  中的节点集合域; data 为数据分组携带的数据净荷, 对于蚂蚁分组该域为空. 更新触发分组指路由发现阶段的前向蚂蚁 (forward ant FANT) 和后向蚂蚁 (backward ant BANT)、路由维护阶段的捷径正向增强蚂蚁 (enforce ant EANT) 和拥塞反向抑制蚂蚁分组 (antirant AANT)、路由维护阶段从需要发送的数据分组中每  $N$  ( $N$  为自然数) 个选择一个作为路由维护数据分组 (route maintain packet RM PKT).

### 3.2 协议设计思想

CALRA 算法的目标是在不出现拥塞的情况下尽量选择代价最小的路径, 包含两个方面的内容: 首先, 需要保证所选择的路由中不出现拥塞, 如果所选择的路由在会话过程中出现了拥塞, 则需要有一定的控制机制解决出现的拥塞问题, 并且算法收敛速度快, 需要重点解决拥塞问题; 其次, 在不出现拥塞的情况下尽量选择代价最小的路径, 以节约资源, 避免级联拥塞情况的出现, 使网络总体性能得到提高, 需要重点解决捷径问题. 上述两个方面的设计都必须考虑算法的收敛速度问题和引入的开销问题, 使算法在对不断变化的网络环境做出快速反应的同时引入的开销不大, 具有较强自适应能力和较高运行效率. CALRA 解决上述问题的基本思想是引入奖惩制度, 即在节点信息素浓度低于拥塞门限时, 加大单个更新触发分组对信息素浓度的影响权值, 随着节点信息素浓度的增加, 单个更新触发分组对信息素浓度的影响权值渐次减小, 在超过拥塞门限时甚至出现负增长, 从而体现低端正向加大奖励, 高端反向惩罚的原则.

具体实现时, CALRA 通过三种方式解决节点的拥塞问题: 通过相应的 QoS 参数以“低奖高惩”的方法影响信息素浓度的更新幅度, 在节点进入拥塞状态前就能逐渐对可能出现的拥塞进行适当的基于负载均衡的分流控制; 当节点处于拥塞状态时, 通过将后续到达的 FANT 丢弃, 避免节点再度成为后续新路由的中间节点, 以接纳控制的方式缓解拥塞节点的拥塞状态, 同时减少拥塞节点对前向蚂蚁分组转发带来的附加控制开销; 处于拥塞状态的节点还通过主动向拥塞节点的上游节点发送 AANT, 使上游节点尽量避免使后续数据分组的传输路径经过拥塞节点, 以缓解拥塞节点的拥塞状态. 对于捷径问题, CALRA 以跳数和路径时延为参数对捷径进行判断, 由发现捷径的节点主动向上游节点发送 EANT, 使上游节点尽量选择所发现的捷径传输后续分组, 将后续业务数据切换到新的低负载路径上, 以提高网络总体性能. 为了加速拥塞问题和捷径问题的收敛, CALRA 将单个 AANT 和 EANT 对信息素浓度增量的贡献设置为同等条件下单个路由发现蚂蚁对信息素浓度增量贡献的正整数倍.

### 3.3 协议中的网络环境参数

算法在具体实现时, 将影响网络性能的主要因素映射为控制信息素表更新的多个可变参数, 在对信息素表中信息素浓度值进行更新时对不同参数赋予不同的影响权值. 通过不同参数对路由选择调整的间接控制, 使算法针对网络不同应用情况找出某些约束条件下节点之间的最佳路由, 如避开重负载节点的路由. CALRA 中的网络环境因素及对应参数主要有以下几个方面:

①信息素当前浓度: 在相同条件下, 当节点的当前信息素浓度较小时, 单个信息素表更新触发分组对信息素浓度增量的贡献较大, 而当节点的信息素浓度较大时, 单个

信息素表更新触发分组对信息素浓度增量的贡献较小, 该参数体现渐进改善捷径问题和拥塞问题的“低奖高惩”信息素浓度更新方法.

②节点移动性因素: 在相同条件下, 节点运动速度  $v$  越大, 单个信息素表更新触发分组对信息素浓度增量的贡献越小, 可以使选择的路径具有更好的时间稳定性.

③路径老化因素: 更新触发分组从源节点到当前节点所经过路径的跳数  $h$  越多, 其对信息素浓度增量的贡献将相应地减小, 可以在不拥塞的情况下寻找出一些相对较短的路由.

④平均排队时延: 路径的总平均排队时延参数  $d$  来表征路由中各节点业务流负载状态和节点对数据业务流分组的处理速度, 更新触发分组所经过路径的总平均排队时延越大, 其对信息素浓度增量的贡献将相应地减小, 以此改善网络的时延特性.

⑤信息素挥发: 通过调整挥发强度参数  $\rho \in (0, 1)$  的大小来调整信息素的挥发速度, 使路由发现蚂蚁找到的路径有一个较恰当的生存时间.

另外, 蚂蚁数量也是该类算法的重要参数之一<sup>[7]</sup>. CALRA 采用按需发送 FANT、EANT、AANT 和路由维护阶段用部分数据分组 RM PKT 进行路由维护的方法对蚂蚁的数量进行控制, 可以在有效减少网络控制开销的同时提高协议的运行效率.

### 3.4 协议实现规则

令  $p_{ij}(dest, t)$  表示在  $t$  时刻节点  $i$  到目的节点  $dest$  的路由中, 选择以节点  $j$  为下一跳的链路  $L_{ij}$  的概率值,  $\phi_{ij}(dest, t)$  表示在  $t$  时刻节点  $i$  到  $dest$  的路由中, 以节点  $j$  为下一跳的链路  $L_{ij}$  的信息素浓度值,  $\Delta\phi_{ij}(dest, t)$  表示在  $t$  时刻节点  $i$  收到单个更新触发分组时为  $\phi_{ij}(dest, t)$  带来的增量值. 令  $\Delta\phi$  为网络初始化完成时单个路由发现蚂蚁分组从源节点  $src$  出发到达一跳邻近节点  $l$  时为节点  $l$  与  $src$  之间的链路带来的信息素浓度增量. 为实现 CALRA 算法, 采用以下操作规则:

规则 1 各节点信息素浓度的初值和终值: 网络初始化时各节点的信息素表为空; 长时间挥发的路径中当  $\phi_{ij}(dest) = 0$  时, 节点  $i$  将表项  $p_{ij}(dest, t)$  删除, 其中,  $j \in kb(i, dest)$ ,  $lab(i, dest)$  为节点  $i$  的所有一跳邻居节点中可以到达目的节点  $dest$  的所有节点构成的集合. 通过该规则, 使 CALRA 能很好的适应邻居节点的动态变化.

规则 2 准入控制规则: 在路由发现阶段, 当无环路径  $P(n) = \langle l_1, \dots, i, \dots, l_n \rangle$  中的节点  $i$  ( $2 \leq i \leq n$ ) 相对路径  $P(n)$  处于拥塞状态时, 节点  $i$  丢弃所收到的来自  $P(n)$  的路由请求分组.

规则 3 拥塞抑制和捷径增强规则: 在路由维护阶段, 无环路径  $P(n) = \langle l_1, \dots, i, \dots, l_n \rangle$  中的每个节点  $i$  ( $2 \leq i \leq n$ ) 在收到 RM PKT 后对本节点的业务负载状态及处理能力进行判断, 当节点  $i$  相对路径  $P(n)$  处于拥塞状态或捷

径状态时, 节点  $i$  产生对应于  $P(n)$  的 AANT 或 EANT. 其中,  $\text{flag} = \text{AANT 或 EANT}$ ,  $\text{seq} = \text{seq} + 1$ ,  $\text{dest} = \text{src}$ ,  $\text{path\_delay} = D_p(i)$ , 顺序提取 RM PKT 的  $\text{path\_node}$  中的所有节点, 得到 RM PKT 所经过的路径  $\text{src} \rightarrow \dots \rightarrow i$  并将其逆序后插入 AANT 或 EANT 的节点列表域  $\text{path\_node}$  中. AANT 或 EANT 根据  $\text{path\_node}$  按照源路由的方式选择下

表 1 蚂蚁分组对信息素表项的更新规则及各参数取值

dest	下一跳节点	$\Delta\phi_{i,i-1}(\text{dest}, t)$	$\phi_{i,i-1}(\text{dest}, t)$	v	h	$\rho$	d
1	$i-1$	$\Delta\phi_{i,i-1}(1, t)$	$\phi_{i,i-1}(1, t)$	$v_i$	$i-1$	$\rho$	$D_p(i-1)$
2	$i-1$	$\Delta\phi_{i,i-1}(2, t)$	$\phi_{i,i-1}(2, t)$		$i-2$		$D_p(i-1) - D_p(i-1)$
3	$i-1$	$\Delta\phi_{i,i-1}(3, t)$	$\phi_{i,i-1}(3, t)$		$i-3$		$D_p(i-1) - 2^* D_p(i-1)$
...	...	...	...		...		...
$i-2$	$i-1$	$\Delta\phi_{i,i-1}(i-2, t)$	$\phi_{i,i-1}(i-2, t)$		2		$D_p(i-1) - (i-3)^* D_p(i-1)$
$i-1$	$i-1$	$\Delta\phi_{i,i-1}(i-1, t)$	$\phi_{i,i-1}(i-1, t)$		1		$D_p(i-1) - (i-2)^* D_p(i-1)$

增量及各参数取值如表 1 所示.

轻负载中间节点在收到路由发现分组后, 在保证不出现环路的情况下对本节点的信息素表进行双向、逐跳更新, 可以充分利用蚂蚁分组所携带的信息为不同的源目的节点对产生更多的冗余路由, 降低后续路由建立时的传输开销和时延, 提高信息素表的更新效率.

规则 5 RM PKT 对信息素表项的更新规则: 在路由维护过程中, 节点  $i$  收到上游节点  $i-1$  转发的来自源节点  $\text{src}$  的 RM PKT, 并通过下游节点  $i+1$  将其向目的节点  $\text{dest}$  成功转发后, 节点  $i$  中需要更新的信息素表项为  $p_{i,i-1}(\text{src}, t)$  和  $p_{i,i+1}(\text{dest}, t)$ , 节点  $i$  在转发 RM PKT 前需要对其进行  $\text{path\_delay} = D_p(i-1) + d^i$  更新;

在路由建立之后进行正常数据传输的过程中, RM PKT 周期性的进行路由维护, 可以减少现有基于蚁群优化的自组网路由协议中各节点周期性向网络中不同目的节点发送路由维护蚂蚁所引入的额外开销和重新路由发现引入的时延.

规则 6 在一次信息素更新过程中, 单个更新触发分组对链路  $L_{ij}$  产生的信息素增量  $\Delta\phi_{ij}(\text{dest}, t)$  (简记为  $\phi_{ij}$ ) 分别由 EANT 产生的  $\Delta\phi_{ij}^s$ 、AANT 产生的  $\Delta\phi_{ij}^c$ 、FANT 和 BANT 产生的  $\Delta\phi_{ij}^d$  以及 RM PKT 产生的  $\Delta\phi_{ij}^f$  构成, 可表示为:

$$\Delta\phi_{ij} = \begin{cases} p[r\Delta\phi_{ij}^s + (1-r)\Delta\phi_{ij}^c] + (1-p)[(1-r) \\ \cdot \Delta\phi_{ij}^d + r\Delta\phi_{ij}^f], L_{ij} \in R_{\text{src,dest}} \\ Q L_{ij} \notin R_{\text{src,dest}} \end{cases} \quad (1)$$

在式 (1) 中,  $R_{\text{src,dest}}$  为更新触发分组从源节点  $\text{src}$  到目的节点  $\text{dest}$  所经过的路由. 上式中, 对于不同的更新触发分组, 参数  $p$ 、 $r$  的取值不同. 对于路由发现蚂蚁 FANT 和 BANT 有  $p=r=Q$  对于 RM PKT 有  $p=Q$   $r=1$  对于 AANT 有  $p=1$   $r=Q$  对于 EANT 有  $p=1$   $r=1$ . 可见, CALRA 的实质是通过 EANT、AANT、路由发现蚂蚁和 RM PKT 实现对信息素浓度的控制, 分别对应式 (1) 中的  $\Delta\phi_{ij}^s$ 、 $\Delta\phi_{ij}^c$ 、 $\Delta\phi_{ij}^d$  和  $\Delta\phi_{ij}^f$  四个分量, 前三个分量都是节点当前的信息素

一跳节点向  $\text{src}$  转发.

规则 4 路由发现蚂蚁、AANT 和 EANT 对信息素表项的逐跳更新规则: 假设节点  $i$  所收到的蚂蚁分组经历的路径  $\text{path\_node} = \{1, 2, \dots, i-2, i-1\}$ , 则节点  $i$  中需要更新的信息素表项为  $p_{i,j-1}(\text{dest}, t)$ , ( $\text{dest} \in \text{path\_node}$ ), 即  $\text{dest}$  的取值遍历从 1 到  $i-1$  的所有节点, 相应信息素浓度

浓度值  $\phi_{i,j}$ , 节点的移动速度  $v$ 、路径的总平均排队时延  $d$  以及更新触发分组从源节点到当前节点所经过的跳数  $h$  等参数的函数. 考虑到第四个分量仅用于正常路由维护的需要, 为简便, 可以取定值为  $\Delta\phi$ . 因此, 对 FANT 和 BANT 有  $\Delta\phi_{ij} = \Delta\phi_{ij}^d = f(\phi, v, d, h)$ , 对 RM PKT 有  $\Delta\phi_{ij} = \Delta\phi_{ij}^f = \Delta\phi$ ; 对 AANT 有  $\Delta\phi_{ij} = \Delta\phi_{ij}^c = l_c f(\phi, v, d, h)$ ,  $l_c$  为整数; 对 EANT 有  $\Delta\phi_{ij} = \Delta\phi_{ij}^s = l_s f(\phi, v, d, h)$ ,  $l_s$  为整数. 令  $f(\phi, v, d, h) = (1 + l_p \phi_{ij})^{-(1/k)} \times (l_v v + l_d d + l_h h)^m \Delta\phi$  其中  $k, m \in N$ ,  $l_p, l_v, l_d, l_h$  为大于 0 的常数. 通过改变整数  $l_s, l_c$  的值, 可以调整单个捷径蚂蚁和拥塞控制蚂蚁对信息素表的影响程度. 为简便, 可取  $l_s = -l_c = l > 1$  另外, 在单个路由发现蚂蚁触发信息素表更新时, 通过改变  $l_p, l_v, l_d, l_h$  的取值, 可以调整节点当前的信息素浓度值、节点的移动速度、节点排队处理时延以及更新分组从源节点到当前节点所经过的跳数等参数对信息素表的影响程度, 并通过  $k$  和  $m$  决定上述各种参数的相对重要性. 可见, 单个信息素表更新触发分组为节点信息素浓度带来的增量, 随着节点当前信息素浓度的增加、节点移动速度的增大、节点队列时延的增加、信息素表更新触发分组从源节点到当前节点距离的增加而减少. 由式 (1) 有

$$\Delta\phi_{ij} = \begin{cases} \{p[r l - (1-r) l] + (1-p)(1-r)\} \cdot \\ f(\phi, v, d, h) + (1-p)r \Delta\phi, L_{ij} \in R_{\text{src,dest}} \\ Q L_{ij} \notin R_{\text{src,dest}} \\ \begin{cases} \frac{(2r-1)l_p + (1-p)(1-r)}{(1+l_p \phi_{ij})^{1/k} (l_v v + l_d d + l_h h)^m} + \\ (1-p)r \} \Delta\phi, L_{ij} \in R_{\text{src,dest}} \\ Q L_{ij} \notin R_{\text{src,dest}} \end{cases} \end{cases} \quad (2)$$

当网络中具有高信息素的路径由于某种原因出现状态的恶化或故障, 可以通过改变  $l, l_p, l_v, l_d, l_h, k$  和  $m$  等参数的值来调整算法对拥塞状态、捷径、当前信息素浓度、移动速度、节点处理时延和路由所经过跳数等不同环境情况的敏感度, 以达到对某种网络问题满意的收敛速度<sup>[17]</sup>.

规则 7 每个节点的处理模块收到更新触发分组提交

的信息后,更新相邻节点间链路的信息素表规则:

$$\phi_{ij} = \phi_{ij} + \Delta\phi_{ij}, L_{ij} \in R_{src, dest}$$

规则 8 在路由发现和路由维护过程中各节点对信息素浓度的挥发规则:

$\phi_{ij}(dest, t+t) = (1-\rho)\phi_{ij}(dest, t)$ ,  $t$  为信息素浓度挥发的更新间隔,  $\rho \in (0, 1)$  为挥发强度参数.

规则 9 在路由发现和路由维护过程中各节点对  $p_{ij}(dest, t)$  的更新规则:

$$p_{ij}(dest, t) = \frac{\phi_{ij}(dest, t)}{\sum_{k \in lab(i, dest)} \phi_{ik}(dest, t)},$$

$$j \in lab(i, dest), \sum_{j \in lab(i, dest)} p_{ij}(dest) = 1$$

规则 10 数据传输期间的噪声规则: 在数据传输阶段, 为了快速解决捷径问题, 使信息素表中的其他路径在成为捷径时拥有被尽快发现的概率, 位于节点  $i$  上的数据分组选择邻居节点  $j$  作为下一跳的概率  $p_{ij}(dest, t)$  如下:

$$p_{ij}(dest, t) = \begin{cases} \text{如果 } \mu \leq \mu_0 \\ \left\{ \frac{\phi_{ij}(dest, t)}{\sum_{k \in lab(i, dest)} \phi_{ik}(dest, t)}, \quad \forall j \in lab(i, dest), \right. \\ \quad \left. Q, \quad \forall j \notin lab(i, dest) \right. \\ \text{如果 } \mu > \mu_0 \\ \left\{ \frac{1}{|lab(i, dest)|}, \quad \forall j \in lab(i, dest), \right. \\ \quad \left. Q, \quad \forall j \notin lab(i, dest) \right. \end{cases}$$

其中:  $|lab(i, dest)|$  为节点  $i$  的一跳邻居节点中可以到达目的节点  $dest$  的所有节点构成的集合包含的元素个数;  $\mu_0 \in [0, 1]$  是系统定义的常数, 随机数  $\mu \in [0, 1]$ , 满足概率  $p(\mu \leq \mu_0) \gg p(\mu > \mu_0)$ .

### 3.5 协议操作步骤

假设系统中各节点都执行同样的路由选择算法, 协议包括路由发现和路由维护两过程.

#### 3.5.1 路由发现过程

步骤 1 源节点  $src$  有数据发送而又没有到目的节点  $dest$  的路由时, 通过按需的方式向网络中发送 FANT 进行路由发现, 其中  $flag = :FANT$ ,  $seq = :seq + 1$ ,  $dest = :dest$ ,  $path.delay = :D_p(src)$ ,  $path.node = : \{src\}$ .

步骤 2 中间节点  $i$  在收到来自节点  $i-1$  的 FANT 后, 根据 FANT 分组的  $path.node$  判断是否出现环路, 将  $path.node$  中已经包含了节点  $i$  的 FANT 判定为出现环路而丢弃. 对于没有出现环路的 FANT, 按照规则 2 进行准入控制. 如果节点  $i$  为可接纳状态, 则根据  $seq$  判断收到的 FANT 分组是否为来自不同路径的重复分组. 对于重复分组, 则进行丢弃处理. 对于非重复分组, 则将 FANT 在进行  $path.delay + d^i$  和  $path.node \cup \{i\}$  更新后转发, 转发时, 如果节点  $i$  中有到目的节点  $dest$  的路由表项, 则节点  $i$  对于 FANT 的后续转发按照单播方式选择信息素表中信息素概率值最大的表项所对应路径进行转发. 否则, 继续进行

广播. 在对重复分组进行丢弃或对非重复分组进行转发之前, 都需要按照规则 4 规则 6~9 修改节点  $i$  的信息素表, 即对于当前信息素表中没有的信息素表项进行增加, 已有的则直接修改对应的表项值, 使修改后的节点的信息素表中包括如下表项: 目的节点  $dest (dest \in path.node)$  的标识、下一跳节点  $i-1$  的标识、节点  $i$  通过节点  $i-1$  到  $dest (dest \in path.node)$  的信息素概率值 (概率值按照规则 6~9 进行更新).

步骤 3 FANT 到达目的节点后, 目的节点根据收到的 FANT 所包含的整个路由发现信息创建 BANT:  $flag = BANT$ ,  $seq = seq + 1$ ,  $dest = src$ ,  $path.delay = D_p(dest)$ , 顺序提取 FANT 的  $path.node$  中的所有节点, 得到 FANT 所经过的路径  $src \rightarrow \dots \rightarrow dest$  并将其逆序后插入 BANT 的节点列表域  $path.node$  中. BANT 根据节点列表域按照源路由的方式选择下一跳节点向源节点进行转发, FANT 死亡;

步骤 4 中间节点  $i$  收到来自节点  $i+1$  的 BANT 后对信息素表中各个表项的信息素概率值按照规则 4 规则 6~9 进行更新, 之后中间节点  $i$  再对 BANT 进行  $path.delay + d^i$  更新, 并选择 BANT 的  $path.node$  中的下一跳节点对该 BANT 进行转发;

步骤 5 源节点收到第一个 BANT 后, 按照规则 4 6~9 对信息素表进行更新之后, 源节点到目的节点的路由发现过程完成, 进入路由维护阶段.

#### 3.5.2 路由维护过程

步骤 1 源节点  $src$  每发送  $N$  个普通数据分组就从中选择一个成为 RM PKT, 并按照规则 10 在信息素表中选择到目的节点的下一跳节点对数据分组进行转发.

步骤 2 中间节点  $i$  收到来自节点  $i-1$  转发来的分组后, 需要判断分组类型. 如果是数据分组则转到步骤 3 如果是控制蚂蚁分组则转到步骤 4

步骤 3 如果为普通数据分组, 直接按照规则 10 进行转发. 如果是 RM PKT, 则首先根据  $path.node$  判断分组所经过的路径是否在节点  $i$  处出现环路. 如果出现环路, 假设  $path.node = \{src, \dots, i, i+1, \dots, i-1\}$  且从  $i$  到  $i-1$  经过了  $m$  跳转发, 则首先需要对节点  $i$  和 RM PKT 进行如下修改:

$\phi_{i,i+1}(dest, t) = Q$ ,  $path.node = \{src, \dots, i\}$ ,  $path.delay = :path.delay - m * \frac{D_p(i-1)}{D_p(i-1)}$ . 在上述修改完成之后, 节点  $i$  再根据修改后的 RM PKT 中各域包含的值, 依次按照规则 3 进行拥塞状态、捷径状态的判断和处理. 在节点状态判断完成之后再按照规则 5~9 所描述方法对节点所维护的信息素表中各信息素概率值进行更新, 最后按照规则 10 选择下一跳节点对修改后的 RM PKT 进行转发. 对于收到的无环路 RM PKT, 则直接在进行拥塞状态、捷径状态的判断和处理之后按照规则 5~9 更新节点  $i$  所维护的信息素表和按照规则 10 选择下一跳节点对进行  $path.delay = :path.delay + d^i$  更新之后的 RM PKT 进行转发.

步骤 4 对收到的 AANT 和 EANT, 中间节点首先按照规则 4 规则 6~9 更新节点所维护的信息素路由表, 之后再对分组的路径时延按  $\text{path\_delay} + d^i$  进行更新, 最后再按照源路由方式选择  $\text{path\_node}$  中的下一跳节点对更新后的分组进行转发.

步骤 5 当中间节点  $i$  发现到下游节点  $i+1$  断链或收到断链消息时, 首先删除出现断链的路径对应的信息素浓度表项, 再查找本节点是否还记录有到目的节点的另一信息素表项对应的冗余路由. 如果有其他冗余路由, 则将选择信息素浓度最大的路径作为后续分组转发的路径, 并按照规则 6~9 对本节点的信息素表进行更新, 此时  $\phi_{ij}(\text{dest}) = (1 - \rho) \phi_{ij}(\text{dest})$  ( $j \in \text{lab}(i, \text{dest})$  且  $j \neq i+1$ ),  $\phi_{i, i+1}(\text{dest}) = 0$  如果本节点没有其他冗余路由, 则继续向本节点的上游节点发送断链信息.

步骤 6 源节点收到断链消息, 或在数据传输完成前发现了断链且没有其他冗余路由可选用时, 则重新进行新的路由发现以获得新的路由, 完成后续数据的传输.

## 4 协议性能分析与仿真结果

### 4.1 协议复杂度

时间复杂度定义为执行一次协议操作所需要运行的步数, 通信复杂度是指执行协议操作所需要传送的信息数, 这里都指最差情况下的复杂度. CALRA 在路由发现过程中通过源节点发出 FANT 分组, 目的节点返回一个 BANT 分组, 来回穿越网络两次, 时间复杂度是  $O(D)$  ( $D$  为网络直径), 路由的查询是在各个节点分布式同时进行的, 通信复杂度为  $O(2N)$  ( $N$  为网络中节点个数). CALRA 采用替代路径机制, 保持着到达同一个节点的多条路由, 当一条路由出现问题时, 用其他路由替代, 所以大多数情况下故障后的时间复杂度为 0 在负载均衡路由协议 ABR 中, 目的节点的返回路由信息可以通过其他稳定路由返回, 即使链路出现故障后, 还可从故障点进行路由选择, 所以时间复杂度和通信复杂度都有所不同. TORA 使用的是

路由反转算法, 通过产生有向无圈图建立路由, 路由失效后, 只在局部进行路由调节, 路由信息不在全网分发, 所以通信复杂度为  $O(2n)$  ( $n$  为受影响的节点数). 如表 2 所示为 CALRA 与三种典型按需路由协议 (AODV、DSR 和 TORA) 和一种负载均衡路由协议 ABR 的复杂度比较, 其中  $d$  为网络拓扑改变时网络中受影响部分的半径,  $m$  为应答分组传输经过的节点数,  $z$  为应答分组传输的半径.

表 2 几种按需路由协议复杂度比较表

参数	AODV	DSR	TORA	ABR	CALRA
时间复杂度 (初始化)	$O(D)$	$O(D)$	$O(D)$	$O(D+z)$	$O(D)$
时间复杂度 (故障后)	$O(D)$	$O(D)$	$O(D)$	$O(d+z)$	$O(d)$
通信复杂度 (初始化)	$O(2N)$	$O(2N)$	$O(2N)$	$O(N+m)$	$O(2N)$
通信复杂度 (故障后)	$O(2N)$	$O(2N)$	$O(2n)$	$O(n+m)$	$O(n)$

### 4.2 仿真结果及分析

路由协议性能主要从分组成功递交率、平均端到端时延和路由附加开销等方面进行评价<sup>[9]</sup>. 对 CALRA 使用 OPNET 仿真软件进行性能验证的结果如图 2~5 所示. 仿真中假设 40 个节点 (其中 20 个为源节点) 采用典型的 Random Waypoint 移动模型在  $1500\text{m} \times 300\text{m}$  的区域内移动, 移动速度在  $(0 \sim 10)\text{m/s}$  间均匀分布, 通过设置不同的停顿时间分别表示网络不同的移动性<sup>[10-14]</sup>. 各节点无线覆盖范围为  $250\text{m}$ , MAC 层采用 IEEE802.11 DCF 信道容量为  $2\text{M bps}$  每个节点的数据分组到达间隔服从参数为  $s$  的指数分布, 每个分组的数据净荷为  $512$  字节, 目的节点从网络的其余节点中随机选取. 各参数取值如下:  $l_p, l_v, l_a, l_h$  和  $m$  均为 1,  $l = 3, k = 2, \rho = 0.2, t = 1, s_w = 0.6, \alpha = 0.4, \beta = 0.8$ .

图 2 为  $\delta = 0.15, h_0 = 0.75$  改变分组产生间隔  $s$  和 RM PKT 间隔的  $N$  取值所得到的仿真结果. 结果表明,  $N = 1$  时网络具有最优分组成功递交率和端到端时延性能, 但引入的路由开销也最大. 随着  $N$  的增大, 分组成功递交率和端到端时延性能相应下降, 而路由开销减小. 当  $N > 3$  后, 网络的分组成功递交率和端到端时延性能随取值变化

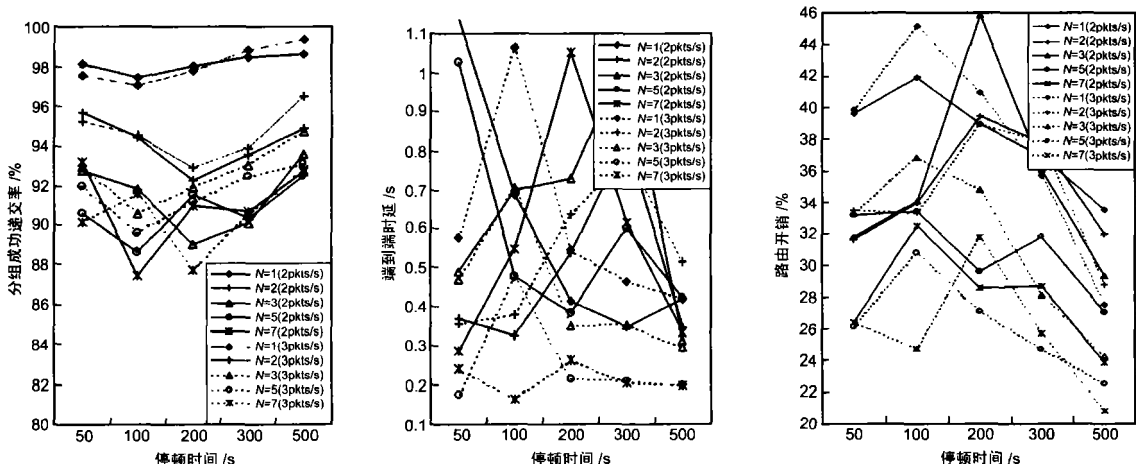
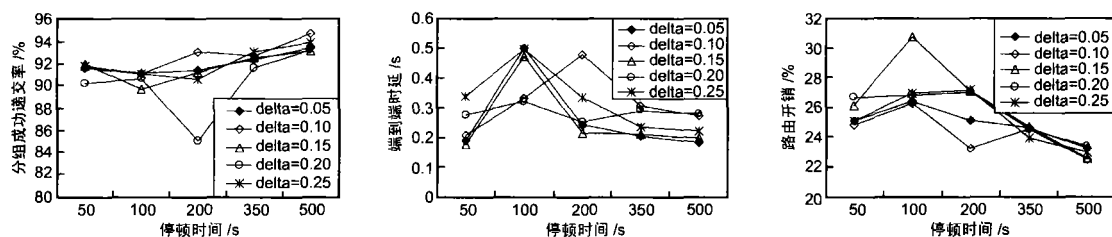
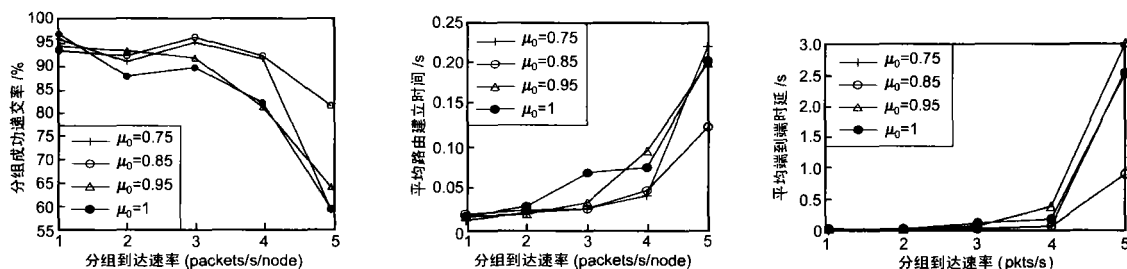


图 2 在分组产生率为 2 分组/秒和 3 分组/秒时  $N$  的取值对 CALRA 性能的影响



图3 在分组产生率为 3packets/s 时 ( $\delta$ ) 的变化对 CALRA 性能的影响图4 在分组产生率为 3packets/s 时  $\mu_0$  对 CALRA 性能的影响

的影响就很小了, 趋于稳定, 路由开销则随  $N$  的增加而较显著下降, 尤其是在网络业务流强度较大的情况下, 显示协议具有良好的适应网络业务流强度和网络规模扩展的能力. 可见, 通过引入 RMPKT, 可以在路由开销增加不大的情况下, 获得协议性能的明显改善. 另外, 即使在大移动性情况下 (如停顿时间为 50s 时), 仍然具有优良的分组成功递交率和端到端时延性能, 表明协议对于节点的移动性具有很强的适应能力. 在移动性较低 (如停顿时间大于 300s) 时, 协议能够获得分组成功递交率、端到端时延和路由开销的综合优化.

根据图 2 选择综合性能较好的结果  $N=5$  对节点状态判决中的关键参数  $\delta$  进行研究, 节点的数据分组到达间隔服从参数  $s=0.333s$  的指数分布时的仿真结果如图 3 所示. 结果表明  $\delta$  (对应图 3 中的变量  $\delta$ ) 取值的变化对协议的性能有比较大的影响, 在业务流强度较大时  $\delta$  的取值比较关键, 当  $0 < \delta < 0.15$  时, 协议的综合性能最优.

为了考查引入噪声对解决捷径问题的改善情况, 本文在固定  $\delta=0.20$  停顿时间为 0 的情况下, 改变噪声门限  $\mu_0$  的取值, 考查网络中分组成功递交率、路由建立的平均时间和分组的平均端到端传输时延性能随网络业务流强度变化的情况, 仿真结果如图 4 所示. 仿真结果表明, 噪声门

限  $\mu_0$  的取值将对网络的分组成功递交率、路由建立的平均时间和分组的平均端到端传输时延性能产生影响. 当  $\mu_0 > 0.85$  时, 随着  $\mu_0$  的增大, 分组的平均端到端传输时延和路由建立的平均时间也相应增加, 而分组的成功递交率下降. 当  $\mu_0 < 0.85$  时, 噪声的减少对路由建立的平均时间影响不大, 但分组的成功递交率和分组的平均端到端传输时延性能则有所恶化. 上述结果表明, 在一定取值范围内, 噪声门限的增加会导致噪声的减少, 使得捷径被发现的时间增加, 从而使得网络中更长路由的出现, 最终使得捷径问题恶化、网络中分组成功递交率和分组的平均端到端时延性能下降. 可见, 噪声的引入能够在一定程度上改善捷径问题的解决. 但是, 当噪声门限过低时, 过多噪声的引入不会带来捷径问题的进一步改善, 反而会导致分组递交率和分组平均端到端传输时延的性能下降. 因此, 需要针对具体应用对噪声门限的选择进行认真权衡.

根据图 2、3、4 的结果, 为了较全面的反映协议性能, 选择  $\delta=0.15$ ,  $\mu_0=0.85$ ,  $s=0.5s$ ,  $N$  分别为 2、3、5 等多种情况将协议与 ARA 进行比较 (如图 5). 与文献 [11] 中采用 10 个源和目的节点对固定的 CBR 连接的仿真环境相比, 本文采用源和目的节点对在全网的 40 个节点中动态随机选取的方法, 能更好的反映协议对于网络动态拓扑和

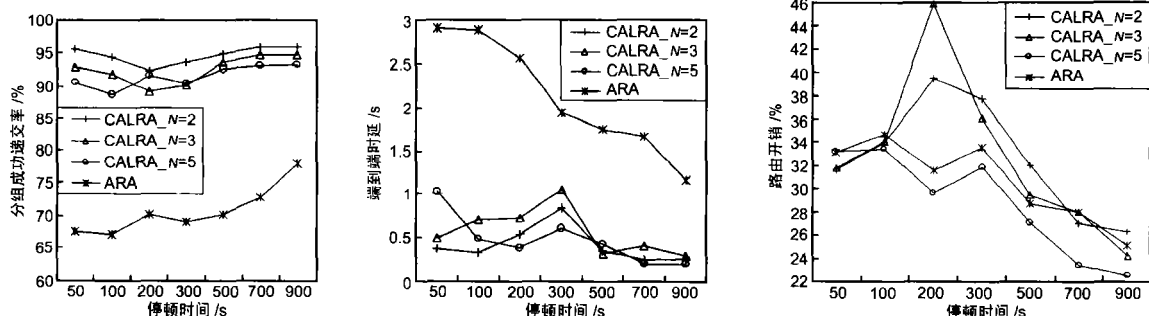


图5 在分组产生率为 2packets/s 时 CALRA 与 ARA 的性能比较



动态业务流负载的适应能力. 因为一旦源节点有数据发送而又没有到目的节点的路由, 就需要重新发起路由发现过程, 导致路由开销和端到端时延的增加. 从图 5 可看出, ARA 和 CALRA 的分组成功递交率都随着节点移动性的降低而上升, CALRA 具有比 ARA 更好的适应动态网络的稳健性, 其受网络拓扑动态变化的影响小于后者. 随着移动性的降低, 由网络拓扑变化造成的路由错误和相应的路由重建和分组重传的次数随之减少, 导致 CALRA 和 ARA 的时延都随停顿时间的增加而下降. 但由于 ARA 没有业务流负载的平衡机制, 使得当路由出现拥塞导致时延上升时, 难以选择业务流负载较轻的路由进行后续数据业务流的传输. 因此, 同等条件下 ARA 的端到端时延大于 CALRA 的端到端时延. 另外, CALRA 在路由发现阶段采用双向逐跳的信息素表更新方式, 通过一个蚂蚁就可以产生更多的路由表项, 使全网路由发现收敛时间缩短、数据分组具有更小的等待时延的同时, 可对向网络中发送的蚂蚁分组进行数量控制, 在收敛速度和控制开销之间进行权衡. 图 5 表明, CALRA 具有与 ARA 相近的路由控制开销, 在  $N$  较小 (加快收敛速度) 时, CALRA 引入的控制开销略高于 ARA, 在  $N > 5$  时, CALRA 可获得低于 ARA 的控制开销.

## 5 结束语

本文提出了一种基于跨层优化和蚁群优化方法的自组网路由协议 CALRA, 协议将整个路径中各节点 MAC 层的总平均估计时延和节点队列缓存的占用情况结合起来共同作为路由选择和路由调整的重要依据进行按需路由发现. 通过拥塞节点丢弃蚂蚁分组、借助部分兼具蚂蚁功能的数据分组实现正常路由表的维护等方法减少了控制开销, 增加了算法的可扩展性. 路由发现阶段通过中间节点对信息素表进行双向和逐跳更新、路由维护阶段通过捷径增强蚂蚁和拥塞抑制蚂蚁在各个节点中以一种分布式的方式完成对捷径问题、拥塞问题的处理, 提高了算法的收敛速度和对异常情况的反应速度. 通过概率选路提供到目的节点的大量冗余路由, 大多数路由分组都集中在最优路径附近, 提高了算法的可靠性和顽存性. 根据蚂蚁分组携带的反映自组网特殊环境特征的各种信息共同控制信息素浓度的更新幅度, 使算法能对分组成功递交率、端到端时延等多个 QoS 参数进行优化.

CALRA 是一种简单有效、收敛速度快、实现难度低的自组网路由协议, 具有较高的实用价值. 下一步需要改变  $l_p$ 、 $l_v$ 、 $l_d$ 、 $l_h$ 、 $m$ 、 $k$  等的取值进行详尽的仿真和理论研究, 并考虑支持多径 QoS 路由问题、引入链路质量参数对协议的性能影响问题和节点的相对运动对链路稳定性的影响问题, 以进一步完善 CALRA 设计, 使其进一步走向实用化.

## 参考文献:

- [1] 郑相全等编著. 无线自组网技术 [M]. 北京: 清华大学出版社, 2004: 6  
Zheng Xiangquan (Ed.). Wireless Ad Hoc Technologies [M]. Peking Tsinghua University Press, 2004: 7 (in Chinese)
- [2] 郑相全, 郭伟. 双向路径重选的自组网负载均衡路由协议 [J]. 计算机研究与发展, 2006, 43(2): 218–223  
Zheng Xiangquan, Guo Wei. A bidirectional path Re-selection based load-balanced routing protocols for ad hoc networks [J]. Journal of Computer Research and Development, 2006, 43(2): 218–223 (in Chinese)
- [3] Wing Ho Yuen, Heung-no Lee, T D Andersen. A simple and effective cross layer networking system for mobile ad hoc networks [A]. In Proc of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications [C]. IEEE Press, Lisbon, Portugal, 2002: 1952–1956
- [4] 郑相全. 基于负载均衡的无线自组网关键技术及算法研究 [D]. 成都: 电子科技大学, 2005  
Zheng Xiangquan. Research on Load-Balance Based Key Technologies and Algorithms for Ad hoc Networks [D]. Chengdu University of Electronic & Science Technology of China, 2005 (in Chinese)
- [5] S Corson, J Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations (RFC 2501) [EB/OL]. <http://www.ietf.org/rfc/rfc2501.txt>, 2004: 12
- [6] 郑相全, 郭伟. 自组网中的负载均衡路由协议 [J]. 计算机科学, 2004, 31(11): 40–45  
Zheng Xiangquan, Guo Wei. Load-balanced routing protocols in Ad Hoc networks [J]. Journal of Computer Science, 2004, 31(11): 40–45 (in Chinese)
- [7] M Dorig, E Bonabeau, T Guy. Ant algorithms and stigmergy [J]. Future Generation Computer System, 2000, 16(9): 851–871
- [8] R Schoonderwoerd, O Holland, J Bruten, L Rothkrantz. Ant-based load balancing in telecommunications networks [J]. Adaptive Behavior, 1997, 5(2): 169–207
- [9] D Camara, Antonio Alfredo F Loureiro. A GPS/Ant-Like Routing Algorithm for Ad Hoc Networks [EB/OL]. <http://www.csie.nctu.edu.tw/~ccyang/WirelessNetwork/Papers/MANET/AdHocUnicast18.pdf>, 2004: 12 (20).
- [10] S Marwaha, C K Tham, D Srinivasan. Mobile Agents Based Routing Protocol for Mobile Ad Hoc Networks [EB/OL]. <http://citeseer.ist.psu.edu/than02mobile>

hm J 2004, 12( 17).

- [ 11] Mesut Gunes Udo Sorges and Ined Bouazizi ARA-the ant colony based routing algorithm for MANETs[A]. In International Conference on Parallel Processing Workshops ( CPWP 02) [ C]. Vancouver B C, Canada, 2002 79– 85.
- [ 12] D Com   M Dorigo F Glover The Ant colony Optimization Meta-Heuristic: New Ideas in Optimization[M]. Ed London, UK: McGraw-Hill 1999, 11– 32.
- [ 13] M Heissenb  ttel T Braum Ants based routing in large scale mobile ad-hoc networks[A]. In Proceedings of the 13th ITG /G I-Fachtagung Kommunikation in verteilten Systemen ( K  S 2003) [ C]. Leipzig Germany, February 25– 28, 2003 181– 190.
- [ 14] S Rajagopalan, C Jakkae, C C Shen Unicast Routing for Mobile Ad hoc Networks with Swarm Intelligence [ EB / OL]. <http://www.cis.udel.edu/~rajagopa/ansi-unicast-udcistr-2003-07-dt-05-01-2003.pdf> 2004-10.
- [ 15] O Hussein, T Saadawi Ant routing algorithm for mobile ad-hoc networks ( ARAMA) [ A ]. In Proc of the 2003 IEEE International Conference on Performance, Computing and Communications Conference[ C]. Phoenix, Arizona

zona, 2003 281– 290.

- [ 16] John S Baras Harsh Mehta A probabilistic emergent routing algorithm for mobile ad hoc networks[A]. In W Opt 03 Modeling and Optimization in Mobile Ad Hoc and Wireless Networks Sophia-Antipolis[ C]. France March 3–5, 2003, 120– 125.
- [ 17] Br  ndan Jennings Rob Brennan Rune Gustavsson, et al. FIPA-compliant agents for real time control of intelligent network traffic[ J]. Computer Network: The International Journal of Computer and Telecommunications Networking 1999, 31( 19): 2017– 2036.

#### 作者简介:

郑相全 男, 1972年 1月出生于四川省资中县, 博士, 副教授, 主要研究方向是无线自组网技术、软件无线电组网技术、认知无线电技术等. E-mail: zxqx69@ hotmail. com

郭 伟 男, 1964年 6月出生于四川省达县, 教授, 博士生导师, 主要从事网络优化设计及仿真技术、网络可靠性、自组网技术、信号检测与处理技术等方面的研究. E-mail: guowe@ uestc. edu. cn

葛利嘉 男, 1957年 11月出生于四川省大邑县, 教授, 博士生导师, 主要从事超宽带无线电技术、军事通信网络技术、认知无线电技术、信号检测与处理技术等方面的研究. E-mail: gelijiayd@ yahoo. com. cn