

# 一种基于跨层负载感知和双向逐跳信息素更新的自组网蚂蚁算法<sup>\*</sup>)

郑相全<sup>1</sup> 郭伟<sup>2</sup> 葛利嘉<sup>1</sup>

(重庆通信学院数字通信与信号处理重点实验室 重庆 400035)<sup>1</sup>

(电子科技大学通信抗干扰技术国家级重点实验室 成都 610054)<sup>2</sup>

**摘要** 本文将跨层优化和蚂蚁优化方法结合起来解决自组网中的负载均衡问题,提出了一种基于跨层负载感知和双向逐跳更新信息素的蚂蚁优化路由协议(CLABHPU)。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,进行按需路由发现和维护;通过拥塞节点丢弃蚂蚁分组的方法减少了控制开销,增加了算法的可扩展性,较好地解决了自组网中现有基于蚂蚁算法的路由协议中普遍存在的拥塞问题和路由开销问题。同时,协议在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,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,提高了算法的可靠性和顽存性。仿真结果表明,CLABHPU 在分组成功递交率、路由开销以及端到端平均时延等方面具有优良性能,能很好地实现网络业务流负载均衡。

**关键词** 跨层负载感知,基于蚂蚁优化的路由协议,信息素更新,自组网

## A Cross-layer Load-aware Based Ant-colony Optimization Routing Protocol with Bi-directional and Hop-by-hop Pheromone Update for Ad hoc Networks(CLABHPU)

ZHENG Xiang-Quan<sup>1</sup> GUO Wei<sup>2</sup> GE Li-Jia<sup>1</sup>

(Key Laboratory of Digital Communication & Signal Processing, Chongqing Communication College, Chongqing 400035)<sup>1</sup>

(National Key Laboratory of Communication, UEST of China, Chengdu 610054)<sup>2</sup>

**Abstract** In this paper, we present a cross-layer load-aware based ant-colony optimization routing protocol with bi-directional and hop-by-hop pheromone update for ad hoc networks(CLABHPU). CLABHPU combines the total path average estimated delay with the buffer occupation ratio of MAC layer as the primary metric for route selection and route adjustment during the on-demand route discovery and route maintenance periods, and CLABHPU makes use of some data packets to implement route maintenance and allow the heavy load nodes to drop the ants. Thus it can decrease the route overhead, increase the scalability of the algorithm and improve the congestion problem commonly existed in existing ant-based routing protocols for ad hoc networks. At the same time, the intermediate nodes bi-directionally update the pheromone table hop-by-hop, which can accelerate the convergence rate of the algorithm and improve the speed of response to abnormal cases. Besides, CLABHPU depends on probability routing table, thus providing a number of redundant paths to the destination and helping increasing the reliability and survivability of the algorithm. The performance of the algorithm is measured. Simulation results show that CLABHPU performs well in the packet delivery rate and routing overhead as well as end-to-end delay.

**Keywords** Cross-layer load-aware, Ant-colony optimization routing protocol, Pheromone update, Ad hoc networks

## 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网络。网络拓扑结构随网络中终端节点的随意移动而频繁变化,整个网络没有固定的基础设施,可以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,设计时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。自组网中现有典型路由协议大都按照典型的分层体系结构进行设计,没有综合考虑不同协议层之间的信息共享和相互影响问题,难以实现网络性能的综合提高。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>[2,3]</sup>的组网方法进行路由协议设计,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,网络综合性能达到最优。另一方面,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,使控制中心成为网络业务流传输的瓶颈。基于蚂蚁算法的路由方法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。

本文将跨层设计思想和蚂蚁算法结合起来解决自组网中的负载均衡问题<sup>[4]</sup>,提出了一种基于跨层负载感知和双向逐跳信息素更新蚂蚁算法的路由协议(CLABHPU)。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

<sup>\*</sup>)国家自然科学基金(60472053)和重庆市科技计划项目(8030)资助。郑相全 博士,主要研究方向是自组网技术、软件无线电组网技术等。

法综合考虑了各协议层之间的信息共享;通过拥塞节点的接纳控制、中间节点对信息素表的双向和逐跳更新等方法减少了协议的控制开销,增加了算法的可扩展性,较好地解决了自组网中现有基于蚂蚁算法的路由协议中普遍存在的拥塞问题和路由开销问题。

## 2 蚂蚁算法研究概述

蚂蚁算法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,可应用于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。

最早在有线网络如 ABC<sup>[6]</sup>、AntNet<sup>[6]</sup> 中引入的基于蚂蚁算法的路由方案难以快速适应动态网络拓扑的自组网应用,但其通过大量人工蚂蚁进行路由发现与维护的基本思想为后续基于蚂蚁算法的路由协议所继承。基于蚂蚁算法的自组网路由协议大多需要通过其它机制来解决对动态网络拓扑的适应问题,如 GPS/AL<sup>[7]</sup> 和 Ant-AODV<sup>[8]</sup> 采用类似于蚂蚁的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>[9]</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>[10]</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>[11]</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>[12]</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。PERA<sup>[13]</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。ARAMA<sup>[14]</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。可见,自组网中现有基于蚂蚁算法的路由协议没有综合考虑各层之间的信息共享和相互影响,协议开销较大,难以真正最大限度地综合利用网络资源和提高网络的整体性能,如路由发现时,需要源节点发送大量蚂蚁分组,实现路由的收敛,将带来大量的附加通信开销;没有考虑拥塞问题和算法的收敛速度问题,算法收敛时间较长的问题始终存在等。

## 3 基于双向逐跳信息素更新的蚂蚁算法和跨层负载感知的路由协议

### 3.1 节点负载状态的跨层感知

每个移动节点监视 MAC 层接口队列,在收到每个分组后都要对节点的平均估计时延进行计算。第  $m$  个分组成功发送后,节点  $i$  的平均估计时延值  $d_m^i$  为<sup>[15]</sup>:  $d_m^i = (1-w)d_{m-1}^i + w(s_m - a_m)$ ,  $m > 1$  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估计时延),  $\overline{D_p(n)} = \frac{D_p(n)}{n}$ 。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  $\eta = B_p(i)/B_c(i)$ 。节点  $i$  的负载状态通过本节点估计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流负载的处理速度、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流强度及本节点 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<sup>[16]</sup>:

### 3.2 CLABHPU 协议设计思想

CLABHPU 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,需要在解决拥塞问题的同时考虑算法的收敛速度和额外开销。CLABHPU 在路由发现阶段采用了前向蚂蚁(FANT)和后向蚂蚁(BANT)两种蚂蚁分组。当节点的业务流负载强度(包括源业务流和中继业务流)较大时,CLABHPU 通过将后续到达的 FANT 丢弃,避免节点再度成为新产生路由的中间节点,以接纳控制的方式缓解拥塞节点的拥塞状态和减少拥塞节点对 FANT 转发带来的附加控制开销。轻负载中间节点在收到蚂蚁分组后,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,降低后续路由由建立时的传输开销和时延,提高信息素表的更新效率。

算法在具体实现时,将影响网络性能的不同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给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,使算法适应网络的不同应用情况,在网络中找出某些约束条件下各节点之间的最佳路由。在 CLABHPU 中所考虑的参数主要有信息素当前浓度、蚂蚁分组从源节点到当前节点所经过路径的跳数、分组所

经过路径的总平均排队时延和信息素挥发强度,并采用按需发送 FANT 的方法控制蚂蚁数量,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。协议采用蚂蚁分组携带的路径列表解决控制分组的环路问题,采用路径类型和节点列表相结合的方式按需检查数据分组转发过程中可能出现的临时环路。

### 3.3 CLABHPU 协议规则

CLABHPU 中使用的分组格式如下<sup>[17]</sup>:

| flag | seq/src | dest | path, delay/path, type | path, node | data |
|------|---------|------|------------------------|------------|------|
|------|---------|------|------------------------|------------|------|

其中,  $flag$  为分组类型标识,用于区别蚂蚁分组和数据分组;  $seq/src$  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  $src$ ;  $dest$  为分组的节点;  $path, delay/path, type$  为蚂蚁分组所历经路径的总平均估计时延  $D_p(n)$  或者标识当前数据分组的传输路径类型;  $path, node$  为分组所遍历  $P(n)$  中的节点集合域;  $data$  为数据分组携带的数据净荷。

令  $p_{i,j}(dest, t)$  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,  $\psi_{i,j}(dest, t)$  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,  $\Delta\phi_{i,j}(dest,t)$  表示在  $t$  时刻节点  $i$  收到单个蚂蚁分组时为  $\phi_{i,j}(dest,t)$  带来的增量值。令  $\Delta\psi$  为网络初始化完成时单个蚂蚁分组从源节点  $src$  出发到一跳邻近节点  $l$  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 CLABHPU 算法, 采用以下操作规则:

**规则 1** 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  $\phi_{i,j}(dest,t)=0$  时, 节点  $i$  将信息素表项  $P_{i,j}(dest,t)$  删除, 其中  $j \in lab(i, dest)$ ,  $lab(i, dest)$  为节点  $i$  的所有一跳邻近节点中可以到达目的节点  $dest$  的所有节点构成的集合。

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 | $\Delta\phi_{i,i-1}(dest,t)$ | $\phi_{i,i-1}(dest,t)$ | $h$   | $\rho$ | $d$                                 |
|--------|-------|------------------------------|------------------------|-------|--------|-------------------------------------|
| 1      | $i-1$ | $\Delta\phi_{i,i-1}(1,t)$    | $\phi_{i,i-1}(1,t)$    | $i-1$ | $\rho$ | $D_p(i-1)$                          |
| 2      | $i-1$ | $\Delta\phi_{i,i-1}(2,t)$    | $\phi_{i,i-1}(2,t)$    | $i-2$ |        | $D_p(i-1) - \bar{D}_p(i-1)$         |
| 3      | $i-1$ | $\Delta\phi_{i,i-1}(3,t)$    | $\phi_{i,i-1}(3,t)$    | $i-3$ |        | $D_p(i-1) - 2 * \bar{D}_p(i-1)$     |
| ...    | ...   | ...                          | ...                    | ...   |        | ...                                 |
| $i-2$  | $i-1$ | $\Delta\phi_{i,i-1}(i-2,t)$  | $\phi_{i,i-1}(i-2,t)$  | 2     |        | $D_p(i-1) - (i-3) * \bar{D}_p(i-1)$ |
| $i-1$  | $i-1$ | $\Delta\phi_{i,i-1}(i-1,t)$  | $\phi_{i,i-1}(i-1,t)$  | 1     |        | $D_p(i-1) - (i-2) * \bar{D}_p(i-1)$ |

**规则 4** 蚂蚁分组对信息素浓度的更新幅度规则: 在一次信息素更新过程中, 单个蚂蚁分组对链路  $L_{ij}$  产生的信息素增量  $\Delta\phi_{i,j}(dest,t)$  (简记为  $\phi_{i,j}$ ) 是节点当前的信息素浓度值  $\phi_{i,j}$ 、路径的总平均排队时延  $d$  以及蚂蚁分组从源节点到当前节点所经过的跳数  $h$  等参数的函数, 即

$$\Delta\phi_{i,j} = \begin{cases} f(\phi, d, h), & L_{ij} \in R_{src, dest} \\ 0, & L_{ij} \notin R_{src, dest} \end{cases}$$

( $R_{src, dest}$  为蚂蚁分组从  $src$  到  $dest$  所经过的路由)。令  $f(\phi, d, h) = (1 + \lambda_p \phi_{i,j})^{-\frac{1}{k}} \times (\lambda_d d + \lambda_h h)^{-m} \Delta\psi$ , 其中  $k, m \in N$ ,  $\lambda_p, \lambda_d, \lambda_h$  为大于 0 的常数。在单个蚂蚁分组触发信息素表更新时, 通过改变  $\lambda_p, \lambda_d, \lambda_h$  的取值来调整  $\phi_{i,j}, d$  以及  $h$  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, 以达到对某种网络问题满意的收敛速度<sup>[18]</sup>。从上式可知, 单个蚂蚁分组为节点信息素浓度带来的增量, 随着  $\phi_{i,j}, d$  和  $h$  的增加而减少, 即

$$\Delta\phi_{i,j} = \begin{cases} f(\phi, d, h), & L_{ij} \in R_{src, dest} \\ 0, & L_{ij} \notin R_{src, dest} \end{cases} = \begin{cases} \frac{\Delta\psi}{(1 + \lambda_p \phi_{i,j})^{1/k} (\lambda_d d + \lambda_h h)^m}, & L_{ij} \in R_{src, dest} \\ 0, & L_{ij} \notin R_{src, dest} \end{cases}$$

**规则 5** 每个节点的处理模块收到蚂蚁分组提交的信息后, 更新相邻节点间链路的信息素表规则:

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

**规则 6** 各节点对信息素浓度的挥发规则:

$\phi_{i,j}(dest, t + \tau) = (1 - \rho)\phi_{i,j}(dest, t)$ ,  $\tau$  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 7** 各节点对  $p_{i,j}(dest, t)$  的更新规则:  $p_{i,j}(dest, t) = \frac{\phi_{i,j}(dest, t)}{\sum_{k \in lab(i, dest)} \phi_{i,k}(dest, t)}, j \in lab(i, dest), \sum_{j \in lab(i, dest)} p_{i,j}(dest) = 1$ 。

### 3.5 协议操作步骤描述

假设系统中各节点都执行同样的路由选择算法, 协议按照规则 1 进行网络的初始化。

**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  $flag = FANT, seq = seq + 1, dest = dest$ ,

**规则 2** 准入控制规则: 在路由发现阶段, 当无环路径  $P(n) = \langle 1, \dots, i, \dots, n \rangle$  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  $P(n)$  的路由请求蚂蚁分组。

**规则 3** 蚂蚁分组对信息素表项的逐跳更新规则: 假设节点收到蚂蚁分组时  $path.node = \{1, 2, \dots, i-2, i-1\}$  则节点  $i$  中需要更新的信息素表项为  $p_{i,i-1}(dest, t), (dest \in path.node)$ , 即  $dest$  的取值遍历从 1 到  $i-1$  的所有节点, 如表 1 所示。

$path.delay = D_p(src), path.node = \{src\}$ 。

**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 3~7 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  $i$  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  $(dest \in path.node)$  的信息素概率值 (概率值按照规则 4~7 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 3~7 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 3~7 对信息素表进行更新, 之后源节点在信息素表中选择到目的节点  $p_{i,j}(dest, t)$  最大的下一跳节点  $j$  对数据分组进行转发。

**步骤 6** 中间节点收到数据分组后, 根据  $path.type$  判断

数据分组是否采用了冗余路径。如果采用了冗余路径,则在分组转发以前首先需要根据  $path\_node$  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,则首先需要对节点  $i$  和数据分组进行修改:  $\phi_{i,j+1}(dest, t) = 0, path\_node = \{src, \dots, i\}$ 。节点  $i$  再选择本节点信息素表中  $p_{i,j}(dest, t)$  最大的下一跳节点转发修改后的数据分组。如果  $path\_type$  表明数据分组没有采用冗余路径,则直接选择本节点信息素表中  $p_{i,j}(dest, t)$  最大的下一跳节点对该数据分组进行转发。

**步骤 7** 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,转发前需将数据分组的  $path\_type$  域修改为冗余路径标识。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 8** 如果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

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>[19]</sup>。对 CLABHPU 使用 OPENET 仿真软件进行性能验证的结果如图 1~2 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<sup>[18]</sup> 在  $1500m \times 300m$  的区域内移动,移动速度在  $0 \sim 10m/s$  间均匀分布,通过设置不同的停顿时间表示网络的不同移动性<sup>[8,9,12,15]</sup>。各节点无线覆盖范围为  $250m$ , MAC 层采用 IEEE802.11 DCF,信道容量为  $2Mbps$ 。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:  $\lambda_p, \lambda_d, \lambda_h$  和  $m$  均为 1,  $k=2, \rho=0.2, \tau=1s, w=0.6, a=0.4, \beta=0.8$ 。

图 1 所示为节点的数据分组到达间隔服从参数  $s=0.333s$  的指数分布时,节点状态判决函数中的关键参数  $\delta$ (对应图 1 中的变量  $delta$ ) 取值变化对协议性能影响。结果表明,当  $0 \leq \delta \leq 0.15$  时,协议能够获得分组成功递交率、端到端时延和路由开销等性能的综合优化。图 1 还给出了  $s=0.5s$  时协议的仿真结果,显示协议具有良好的适应网络业务流强度和规模扩展的能力。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,获得综合优化。

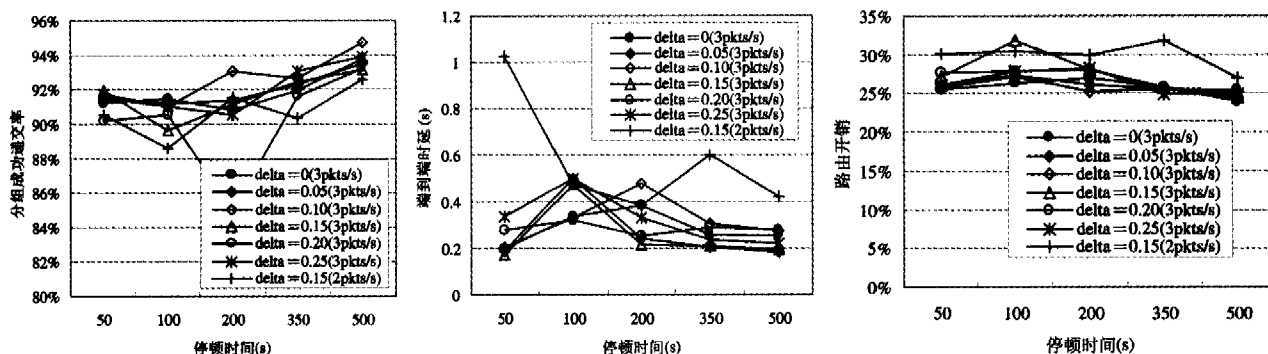


图 1 不同业务流强度情况下  $\delta(delta)$  的变化对 CLABHPU 性能的影响

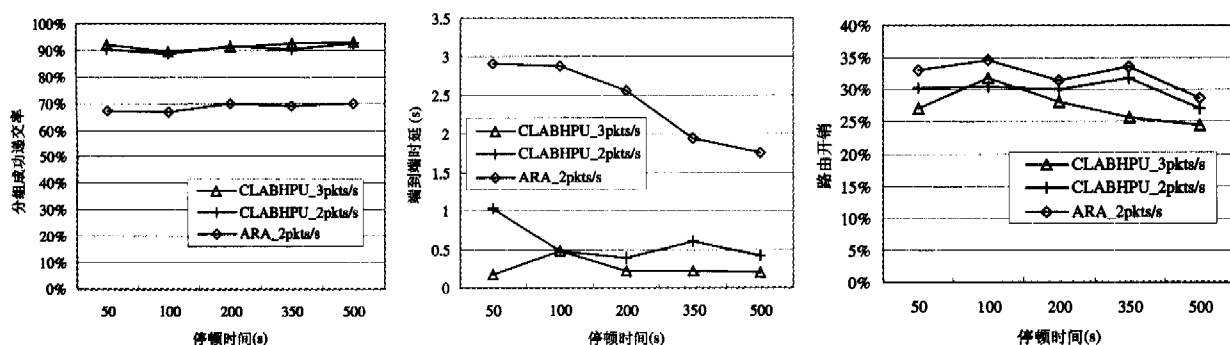


图 2 CLABHPU 与 ARA 的性能比较

图 2 为  $\delta=0.5, s=0.5s$  和  $0.333s$  时协议与 ARA 进行比较的仿真结果。与文[9]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,能更好地反映协议对于网络动态拓扑和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。图 2

表明,CLABHPU 和 ARA 的分组成功递交率都随着节点移动性的降低而上升,且前者比后者具有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以及相应的路由重建和分组重传的次数随之减少,使 CLABHPU 和 ARA 的时延都随停顿时间的增加而下降。但由于 ARA 没有业务流负载的准入控制机制,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 具有大于 CLABHPU 的端到端时延。另外,CLABHPU 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,并可以使 CLABHPU 获得低于 ARA 的控制开销。

**结束语** 本文提出了一种基于跨层优化和蚂蚁算法的自组网路由协议 CLABHPU,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的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,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,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。下一步需要研究  $\lambda_p$ 、 $\lambda_d$ 、 $\lambda_h$ 、 $m$ 、 $k$  等参数对协议性能的影响,并考虑支持多径 QoS 路由问题、链路质量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 CLABHPU 和使其进一步走向实用化。

## 参考文献

- 1 IETF. Mobile ad hoc networks charter[EB/OL]. <http://www.ietf.org/html.charters/manet-charter.html>
- 2 Safwati A, Hassanein H, Mouftah H. Optimal cross-layer designs for energy-efficient wireless ad hoc and sensor networks[C]. In: proceedings of the 2003 IEEE International Conference on Performance, Computing, and Communications Conference, 9-11 April 2003, 123~128
- 3 Yuen W H, Lee H, Andersen T D. A simple and effective cross layer networking system for mobile ad hoc networks[C]. In: the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Sep 2002, 4: 1952~1956
- 4 郑相全, 郭伟. 自组网中的负载均衡路由协议. 计算机科学, 2004, 31(11): 40~45
- 5 Schoonderwoerd R, Holland O, Bruten J, et al. Ant-based load balancing in telecommunications networks[J]. Adaptive Behavior, 1997, 169~207
- 6 Caro G D, Dorigo M. AntNet: A Mobile Agents Approach to Adaptive Routing[R]. [Technical report IRIDIA/97-12.] Univer-

- site Libre de Bruxelles, Belgium, 1997
- 7 Camara D, Loureiro A A F. A GPS/Ant-Like Routing Algorithm for Ad Hoc Networks[C]. IEEE Wireless Communications and Networking Conference (WCNC'00), Chicago, IL, September 2000
- 8 Marwaha S, Tham C K, Srinivasan D. Mobile Agents based Routing Protocol for Mobile Ad hoc Networks[C]. IEEE Global Telecommunications Conference (GLOBECOM'02), Taipei, Taiwan, November 17-21, 2002
- 9 Gunes M, Sorges U, Bouazizi I. ARA - The Ant-Colony Based Routing Algorithm for MANETs[C]. In: International Conference on Parallel Processing Workshops (ICPPW'02), Vancouver, B C, Canada, August 2002, 79~85
- 10 Dorigo M, Caro D G. The Ant Colony Optimization Meta-Heuristic, New Ideas in Optimization[M]. Corne D, Dorigo M Glover F editors, McGraw-Hill, 1999
- 11 Heissenbttel M, Braun T. Ants-Based Routing in Large Scale Mobile Ad-Hoc Networks[R]. [Technical report]. University of Bern, Kommunikation in verteilten System(KiVS03), 2003
- 12 Rajagopalan S, Jaikao C, Shen C C. Unicast Routing for Mobile Ad hoc Networks with Swarm Intelligence[R]. [Tech Rep 2003-07]. University of Delaware, May 2003
- 13 Baras J S, Mehta H. A Probabilistic Emergent Routing Algorithm for Mobile Ad hoc Networks[C]. WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, March 2003
- 14 Hussein O, Saadawi T. Ant routing algorithm for mobile ad-hoc networks(ARAMA)[C]. In: Proc. of the 2003 IEEE International Conference on Performance, Computing, and Communications Conference, April 2003, 281~290
- 15 Song Joo-Han, Wong V, Leung V C M. Load-aware on-demand routing(laor) protocol for mobile ad hoc networks[C]. In: VTC 2003-spring, April 2003, 3: 1753~1757
- 16 郑相全. 基于负载均衡的无线自组网关键技术与算法研究[D]. 成都: 电子科技大学, 2005
- 17 Zheng Xiangquan, Guo Wei, Liu Renting. An ant-based distributed routing algorithm for ad hoc networks[C]. ICCAS2004, Chengdu, China, June 2004, (vol. I): 412~417
- 18 Jennings B, Brennan R, Gustavsson R, et al. FIPA-compliant agents for real-time control of intelligent network traffic[J]. Computer Network, 1999, 31(19): 2017~2036
- 19 Corson S, Macker J. Mobile Ad hoc Networking(MANET): Routing Protocol Performance Issues and Evaluation Considerations (RFC 2501)[EB/OL]. <http://www.ietf.org/rfc/rfc2501.txt>
- 20 Johnson D B, Maltz D A. Mobile Computing[M]. Kluwer Academic Publishers, 1996

(上接第 34 页)

- 4 Tan G, Miu A, Guttat J, et al. Forming Scatternets from Bluetooth Personal Area Networks[A]. MIT-LCS-TR-826[R]. 2001
- 5 Basagni S, Pereoli C. Multihop Scatternet Formation for Bluetooth Networks[A]. In: IEEE Vehicular Technology Conference (Vol. 1)[C], 2002. 424~428
- 6 Záruba G V, Basagni S, Chlamtac I. Bluetrees - Scatternet Formation to Enable Bluetooth-Based Ad Hoc Networks[A]. In: IEEE International Conference on Communications (vol. 1) [C], Finland, Helsinki, 2001. 273~277
- 7 Salonidis T, Bhagwat P, Tassiulas L, et al. Distributed Topology Construction of Bluetooth Personal Area Networks[A]. In: IEEE Infocom (vol. 3)[C]. USA: Anchorage, 2001. 1577~1586
- 8 Wang Z, Thomas R J, Haas Z J. Bluenet - a new scatternet forma-

- tion scheme[C]. 35th Hawaii International Conference on System Science(HICSS-35), Big Island, Hawaii, 2002
- 9 Foo Chun-Choong, Chua Kee-Chaing. BlueRings - Bluetooth Scatternets with Ring Structures[A]. IASTED International Conference on Wireless and Optical Communication(WOC 2002)[C]. Canada: Banff, 2002
- 10 Law C, Siu K Y. A Bluetooth Scatternet Formation Algorithm [A]. In: Proceedings of the IEEE Symposium on Ad Hoc Wireless Networks[C]. USA: San Antonio, 2001. 2864~2869
- 11 林鸿, 丁文芳, 高强. 一种蓝牙分散网拓扑结构创建和网络路由分布式算法. 计算机研究与发展, 2003, 40(2): 201~206
- 12 任秀丽, 全成斌, 等. 一种改进的蓝牙分散网形成算法. 小型微型计算机系统, 2003, 25(11): 1942~1945