

# 基于蚁群算法的非结构化 P2P 搜索研究

赵 宏, 谢伟志, 张晨曦

(同济大学 电子与信息工程学院, 上海 201804)

**摘 要:**在非结构化 P2P 网络中,如何降低消息开销,提高搜索效率,是解决搜索问题的关键。引入人工智能领域的蚁群算法,针对其机制可能产生误导现象的问题,提出一种基于多种信息素的改进方法,使算法可以根据不同的文件类型,智能地选择路由方向。对算法建模和仿真,以控制变量的方法进行分析对比后显示,改进算法在平均查询次数及命中率方面比传统算法有较为明显的提高,更加适合被应用于目标文件所属类型不断变化的真实 P2P 网络。

**关键词:**蚁群算法;非结构化点对点网络;搜索算法

**中图分类号:**TP393.09

**文献标识码:**A

**文章编号:**1673-629X(2009)02-0031-04

## Research on Unstructured P2P Network Search Algorithm Based on Ant Colony Optimization

ZHAO Hong, XIE Wei-zhi, ZHANG Chen-xi

(Dept. of Electronics and Information, Tongji University, Shanghai 201804, China)

**Abstract:** It is important to reduce the message packets and improve searching efficiency in decentralized and unstructured P2P networks. Ant colony optimization (ACO) in artificial intelligence is introduced. An improved method in which message packages can be routed according to different types of files based on multiple pheromones is developed to reduce the number of misleading routing caused by ACO. After the modeling, simulation and also analyzing by controlling variables in the same environment, results demonstrate the new algorithm is better than the traditional one on average hops and hit rate per query, and is more suitable for the real P2P networks where the type of the file being requested may change.

**Key words:** ant colony optimization; unstructured P2P network; searching algorithm

### 0 引 言

随着网络规模的不断扩大,越来越多的应用已经从传统的 C/S 模式转向 P2P 系统。在 P2P 系统中,每个结点既充当服务器,为其他结点提供服务,同时也享用其他结点提供的服务。这使得 P2P 技术与传统分布式系统相比,在非中心化、可扩展性、健壮性、性价比、隐私保护和负载均衡等方面都具有无可比拟的优势。

对于不同的 P2P 网络,因为底层拓扑结构的不同,存在着不同的搜索算法。集中式拓扑结构中,资源的发现依赖于中心目录系统,容易造成服务器单点故障、访问的网络瓶颈和版权等相关问题。结构化拓扑受结点频繁加入和退出的影响较大。并且,DHT 精确关键字映射的特性决定了其对内容/语义等复杂查询

缺乏支持。非结构化系统则完全放弃了全局性的目录索引或规则,其结构简单、更容易实现,在当前应用较广。然而,由于非结构化网络中结点之间的链路没有遵循某些预先定义的拓扑来构建,没有可以依赖的全局信息来指导搜索的方向,在定位资源时,就不可避免会带有盲目性。这不但会导致搜索性能低下,大量产生的查询消息包还会严重降低网络性能。

文中将基于人工智能的蚁群算法应用到非结构化 P2P 网络的搜索中来,分析了直接使用传统的蚁群算法时存在的不足,提出了一种基于多种信息素调控机制的改进的蚁群算法。该方法不需要全局性的文件分类规则就可以根据被搜索文件的类型智能地进行路由,并且拥有较好的搜索效率。

### 1 相关工作

在非结构化的 P2P 网络中,泛洪(Flooding)方式将查询消息传播给所有的邻居节点,是最基本的,也是最盲目的搜索方式。其后出现了多种改进算法,如改进

收稿日期:2008-05-27

作者简介:赵 宏(1983-),男,上海人,硕士研究生,研究方向为 P2P 网络;张晨曦,博士,博士生导师,教授,研究方向为计算机体系结构、P2P 网络。

的 BFS(Modified BFS)<sup>[1]</sup>、有向 BFS(Directed BFS)<sup>[2]</sup>、迭代加深技术 (Iterative Deepening)<sup>[2]</sup>、随机漫步法 (Random Walker)<sup>[3]</sup>、范围扩张 (Expanding Ring)<sup>[3]</sup> 等。从效果上看,这些算法的确提高了搜索的性能,然而若从实现机制背后的原理来分析,改进的算法往往缺乏足够的理论依据,显得过于机械。

目前,针对该问题的一个研究趋势,是利用 P2P 资源搜索时存在的局部性原理,实现更加智能的搜索:基于 Reputation Learning<sup>[4]</sup> 的算法利用的是时间上的局部性,即用户的搜索请求往往与历史搜索有较大的相关性,某节点下一次查询的消息类型很有可能就在其历史查询所涉及的消息类型内;而基于 Interest - based Locality<sup>[5]</sup> 的算法则利用空间上的局部性,即拥有相同资源的结点,他们更加有可能处理彼此所发出的查询请求,因为相同的资源在某种程度上代表着他们有相同的兴趣。

不论是基于 Reputation Learning 的算法,还是基于 Interest - based Locality 的算法,其共同的实质是:各结点根据所知的历史与当前信息,选择更有可能处理查询请求的结点作为下一跳的路由。这就对算法提出了两个要求:一是尽量提高每一次选择的准确率从而避免盲目发包;另外,算法必须具有一定的学习能力,在做出不当选择后,能够根据反馈结果进行动态的优化。为解决以上两个问题,人工智能领域中的蚁群算法 (ACO)<sup>[6]</sup> 被引入进来。

## 2 传统的蚁群算法在 P2P 系统中的应用

蚁群算法是受自然界中蚂蚁搜索食物行为启发而提出的一种智能优化算法。仿生学家研究发现,蚂蚁能够在它所经过的路径上留下一一种被称为外激素的物质,而且蚂蚁在运动过程中能够感知这种物质的存在及其强度,并以此指导自己的运动方向,蚂蚁倾向于朝着该物质强度高的方向移动。因此,由大量蚂蚁组成的蚁群的集体行为便表现出一种正反馈现象:某一路径上走过的蚂蚁越多,则后来者选择该路径的概率就越大,蚂蚁个体之间就通过这种信息交流来搜索食物。

在基于蚁群算法机制的 P2P 搜索算法中,查询消息包可以看作是蚂蚁,搜索的目标视为食物,存在搜索目标的结点就是食物源。如图 1 所示,源结点 S 发出搜索请求时,就相当于派出蚂蚁到网络中寻找食物。网络中的结点都维护一张信息素表,结点 S 首先查找本地信息素表,选择其中信息素浓度 (Value) 最高的结点 (B) 作为下一跳结点。查询消息包被转发到 B 结点后,

B 结点首先查看本地是否拥有食物。如果没有,则将消息包中的 TTL(存活时间) 数值减 1,并根据本地信息素表选择合适的下一跳结点继续转发消息包。某一结点在收到消息包后,如果发现 TTL 为 0,则不再转发消息包。如果消息包在 TTL 减为 0 之前到达了拥有食物的目标结点 (T),就会返回一个命中消息包,这可以看成是找到食物的蚂蚁沿原路返回源结点,沿途释放信息素 (即修改结点的信息素表)。

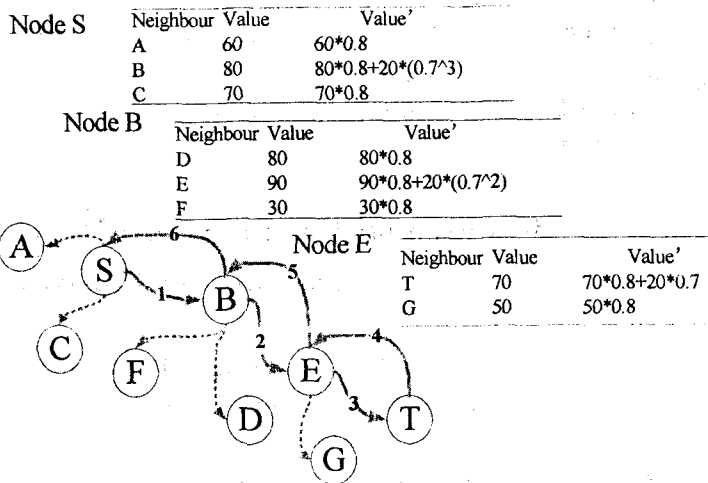


图 1 传统蚁群算法机制图

在返回命中消息包时,沿途结点的信息素表更新规则如下:

$$\alpha = (1 - \rho) * \alpha + \Delta\alpha \quad (1)$$

其中  $\rho$  是挥发系数,通常设置  $\rho < 1$ ,用来避免信息素的无限增加。 $\Delta\alpha$  为信息素的增量,这里根据下式得出:

$$\Delta\alpha = Q * (\mu * St_i) \quad (2)$$

$Q$  是增加的信息素原始总量,为常数。 $\mu$  为结点浓度差,是一介于 0 和 1 之间的调节因子。 $St_i$  为目标结点  $t$  到结点  $i$  的跳数,这样就保证了在每次增加信息素浓度时,离目标结点越近的结点增量越多,其增量随着当前结点与目标结点距离的增大而递减。每隔一段时间,信息素会自动更新,形成信息素的挥发现象,这时,  $\Delta\alpha$  取值为 0。

在图 1 中,假设返回消息包的途中,各结点都经过一次挥发,挥发系数  $\rho$  取 20%,增加的信息素原始总量  $Q$  为 20,结点浓度差  $\mu$  取 70%,则命中消息包到达结点 B 时,其信息素表中,邻居 D 和 F 对应的信息素都经过了一次挥发,因而新的浓度 (Value') 分别变为  $(1 - 20\%) * 80$  和  $(1 - 20\%) * 30$ 。对于邻居 E 的信息素浓度,虽然其经过一次挥发后降到了  $(1 - 20\%) * 90$ ,但是因为沿该路径成功搜索到了食物,所以须对其进行增加, B 结点与目标结点 T 之间距离两跳,新增加的信息素浓度就是  $20 * (0.7^2)$ 。这样,经过此次查询后, B

结点的信息素表中,  $D$  和  $F$  两项对应的信息素浓度降低,  $E$  项对应的信息素浓度升高。下次, 当有新的查询包到达  $B$  结点时, 因为  $E$  结点拥有更高的信息素浓度, 代表历史上经过该结点找到文件的概率更高, 所以就会将  $E$  结点选作下一跳结点。

对上法进行分析, 信息素实际上是一种与本结点相关的历史搜索结果的统计数据, 算法根据该数据结合一定的启发式机制来选择下一跳的路由, 在一定程度上可以减少盲目性。比如, 当局部性现象表现为用户的搜索请求总是局限于同一类文件时, 该算法就可以获得不错的效率。然而, 真实的网络中, 用户的搜索请求往往并非仅限于一类。这时, 使用单一信息素作为判断依据的传统蚁群算法, 因为机制只能反映哪些结点上存在文件的可能性更大, 却不能反映这些结点上的文件是否真正符合本次查找的要求, 所以就可能出现如下的误导现象: 假设某结点  $N_d$  拥有大量类型为  $A$  的文件, 而没有任何类型为  $B$  的文件。如果在前一次查询中, 在  $N_d$  上成功地找到了类型为  $A$  的文件, 那么, 在下次查找中, 如果要搜索  $B$  类文件, 则先前关于搜索类型  $A$  文件时产生的信息素将会引导蚂蚁走向完全没有  $B$  类型文件的  $N_d$ 。由此, 我们认为, 对传统蚁群算法进行适当改进, 减少误导现象发生的频率, 将有助于进一步减少选择下一跳路由时的盲目性, 从而提高算法的准确性与收敛速度。

### 3 基于多种信息素的蚁群算法在 P2P 系统中的应用

考虑扩展以上的蚁群算法中的信息素调控机制, 使蚂蚁在搜索过程中, 可以针对不同的目标文件所属类别选择更合适的路径, 从而使搜索的方向朝着更有可能拥有文件的结点进行路由。定义了两种类型的信息素: 同类最优路径信息素 ( $\alpha$  信息素) 和分类相关性信息素 ( $\beta$  信息素)。 $\alpha$  信息素用于表示与本结点中某文件类别对应的下一跳各邻居结点的优先度。 $\beta$  信息素表示本结点中各分类与上一跳邻居结点相应分类的相关度。如图 2 所示, 假设结点  $S$  上存在文件类型:  $T1$ ,  $T2$ ,  $T3$ , 当源结点  $S$  发起查询消息包时, 除了指定目标文件名外, 还需要给出该文件在本地分类中所属的类型(假设为  $T2$ )。根据结点  $S$  的本地  $\alpha$  信息素表, 结点  $C$  被选取作为下一跳结点。当消息包转发到  $C$  结点时, 如果发现本地没有目标文件, 必须首先为其指定一种本地的文件类型。做法是根据本地  $\beta$  信息素, 选择一种与该文件在上一跳中所属类型最相似的本地文件类型( $C: T3$ ), 然后再根据该本地类型通过本地  $\alpha$  信息素表选择下一跳结点( $T$ )。

结点S的 $\alpha$ 信息素表		结点C的 $\beta$ 信息素表		结点C的 $\alpha$ 信息素表	
T1	Neighbour Value	S: T1	Native Category Value	T1	Neighbour Value
A	60	T1	60	B	60
B	80	T2	70	C	80
C	70	T3	80	D	70
T2 Neighbour Value		S: T2		T2 Neighbour Value	
B	60	T1	60	C	60
C	80	T2	70	D	80
D	70	T3	80	E	70
T3 Neighbour Value		S: T3		T3 Neighbour Value	
C	60	T1	60	D	60
D	80	T2	70	T	80
E	70	T3	80	E	70

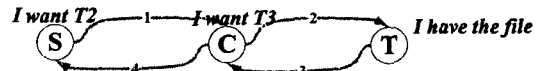


图2 改进后的蚁群算法机制图

当有命中消息包从某个邻居返回时, 结点对该邻居相应的两种信息素同时进行本地更新, 更新规则与使用单一信息素的蚁群算法一致。另外, 为了加快算法的收敛速度, 在根据信息素浓度进行选择时, 算法默认挑选信息素浓度最大的项, 同时也允许其以某一小概率  $b$  突变选择其他的候选项。

## 4 实验仿真

### 4.1 实验方法

文中实验使用的 peersim 模拟器, 是一个开源的、基于 JAVA 的大型 P2P 模拟器框架。其支持两种模拟方式: 基于事件驱动的模拟和基于周期驱动的模拟。实验假设网络中存在  $N$  个不同文件, 以从 1 开始的整数进行标识。初始化网络时, 每个结点从中随机选取  $M$  个文件作为本地文件, 并对其分类(本实验采用对文件标识以  $B$  为基数取模的方式进行分类, 实验中  $N$  取 100,  $B$  取 10), 初始化时各结点的本地  $\alpha$  信息素表中各项为本地文件所对应的类型, 对应浓度为 0。本地  $\beta$  信息素表为空(路由过程中, 若  $\beta$  信息素表中没有所查询项, 则自动增加, 初值为 0)。本实验采用基于周期驱动的模拟方式, 当每个周期到来时, 各结点的两种信息素都自动进行一次挥发, 同时在网络中随机选择一个结点作为查询请求的发起者, 并在  $N$  个文件范围内随机选择一个作为需要查询的文件, 进行一次搜索操作。

### 4.2 实验结果

在结点数为 5000, TTL 为 10, 结点浓度差 90%, 每个结点的邻居结点数为 5, 网络中不同文件数 100, 结点上随机分配 20 个文件的情况下, 对 P2P 网络搜索中的关键衡量标准平均查询次数与命中率进行统计得到以下实验结果。

如图 3、图 4 所示, 改进的算法不仅在平均查找次数上比泛洪明显减少, 而且在平均查找次数与命中率上均优于传统的蚁群算法。特别是, 随着查询次数的增加, 这种优势也越发明, 而当算法经过足够长时间

的学习而趋于稳定后,两者之间的差距也趋于稳定。我们认为,由于试验中采取的是随机选择查询文件的方式,被选到的文件所属的类型就可能在多个不同类型之间发生切换。在这样的情况下,新算法有效减少了原算法所无法避免的误导现象,因而获得了更好的效率。

查询次数与平均查询次数关系图

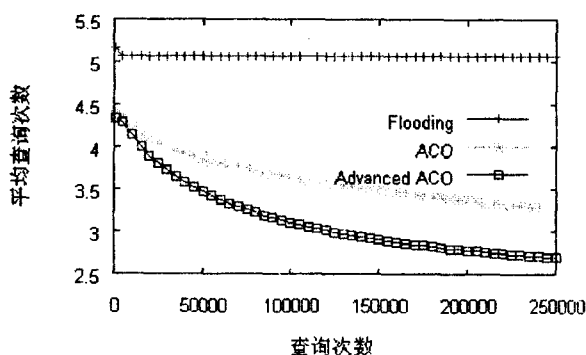


图 3 查询次数与平均查询次数关系图

查询次数与命中率关系图

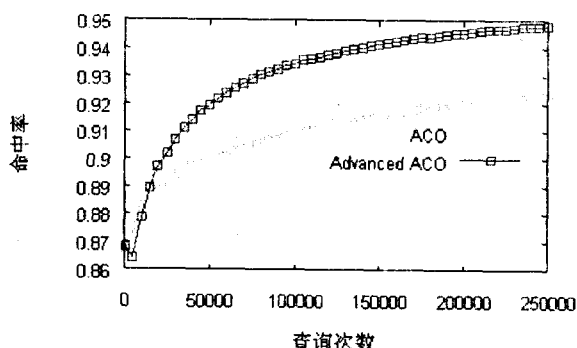


图 4 查询次数与命中率关系图

#### 4.3 实验参数分析

由于实验过程中预先设定的条件较多,为了使结果更有说明力,以下采用控制变量法,对实验中的参数进行进一步分析。

表1、图5、图6是对结点浓度差进行单独调整后的实验结果。应当看到,当结点浓度差取值较小时,蚂蚁返回源结点途中所释放的信息素过小,会导致算法丧失其正反馈的特性,进而无法对下次搜索进行有效的指导,此时两种算法的性能大致相同。而在其他情况下,改进的算法性能均优于传统的蚁群算法。另外,对于TTL、结点数、结点邻居数、结点文件数等参数一一采用类似的方法进行实验,也都得到了改进算法优于传统算法的结果。

表 1 结点浓度差对实验结果的影响

TTL	查询次数	结点文件数	结点数	邻居数	网络文件数
10	100000	20	5000	5	100

平均查询次数与结点浓度差关系图

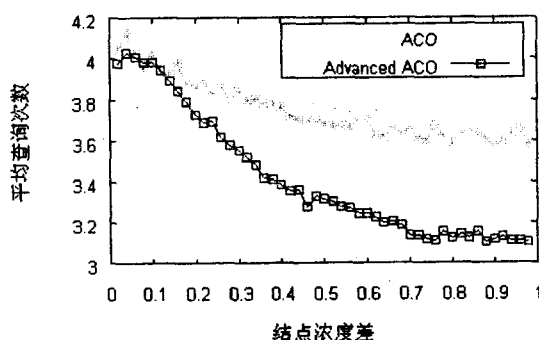


图 5 平均查询次数与结点浓度差关系图

命中率与结点浓度差关系图

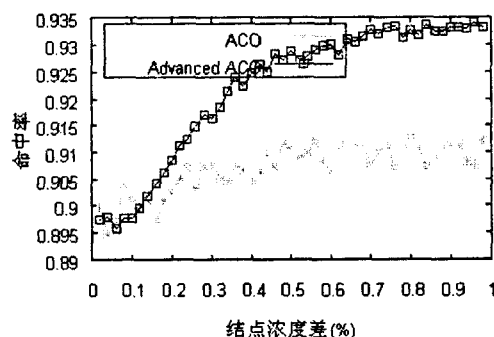


图 6 命中率与结点浓度差关系图

## 5 结束语

文中对于蚁群算法在 P2P 网络中应用时存在的缺陷进行了分析,提出了一种新的蚁群算法,该方法可以根据不同的目标文件所属类别选择更合适的路径进行路由。仿真实验结果证明新方法在平均查询次数及命中率方面比传统蚁群算法有较为明显的提高,是一种有效的改进方法。我们认为,对于节点每次发起的查询,假设其目标文件总是局限于同一类型,这是不符合真实用户的行为特征的。新算法的机制保证了其在目标文件所属类型发生变化时,能更好地工作,因此相对于传统的蚁群算法更加适用于真实的 P2P 网络。

#### 参考文献:

- [1] Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks[C]// In: Proc. of the 11th Int'l Conf. on Information and Knowledge Management. New York: ACM Press, 2002:300-307.
- [2] Yang B, Garcia-Molina H. Efficient Search in Peer-to-Peer Networks[C]// In: Proc. of the 22nd IEEE ICDCS'02. Vienna: IEEE Computer Society, 2002:5-14.
- [3] Lv Q, Cao P, Cohen E, et al. Search and Replication in Unstructured Peer-to-Peer Networks[C]// In: Proc of the

(下转第 38 页)

## 2.2 实验比较

现在来对图 2 和图 3 中的数据进行分析,它们都是对 1000 个语音包进行采样。采用文献[4]算法所得到的丢包率都很低,在 $\leq 0.3\%$ 浮动,主要原因是此算法在发送的 1000 个语音包中选择了第 4 大延迟作为 max,结果会把延迟最大的至多 3 个包丢弃(有可能第 2 大、第 3 大的延迟与 max 相等,所以至多有 3 个包丢失),从丢包率看,此算法明显比文中算法要好,但在文献[5]中曾指出语音包的时延不能超过 400ms。文献[4]的最大缺点也正是在此。相比之下,文中的算法是维持时延在合理范围之内对丢包率进行最大程度上的降低。通过 10 次采样,缓冲区的大小平均维持在 150ms 左右,丢包率在 4%~10% 左右浮动,所以,延时要明显好于上述算法。其实,运用文中算法时,要是不考虑 400ms 这个阈值的话,在时延接近 700ms 时,丢包率不一定会比文献[4]差。

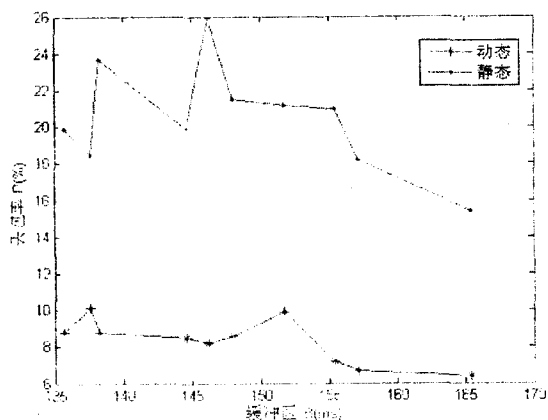


图 3 比较图

## 3 结束语

针对语音接收端如何动态设置抖动缓冲区大小而提出的一种思路,在文中算法中,是在保证时延的前提下,而尽可能地减小丢包率,但是,算法还存在一些明显的不足:首先,设置发送完 20 个语音包时,调整一次抖动缓冲区大小,算法只能保证在每次发送的这 20 个语音包中最大可能消除抖动现象,但前 20 个包中的最后一个语音包与后 20 个包中第一个语音包,却不可避

免地产生了抖动;其次是语音包到达缓冲区时,由于网络情况的不稳定,到达缓冲区的语音包会出现失序的情况,这必然又会产生不必要的丢包情况。综上所述,下一步的工作主要集中在上面几点,要通过进一步的研究和学习,最大限度地解决上述问题。

## 参考文献:

- [1] Khawam K, Kofman D. Flow Size - Aware Proportional Fair Scheduler[C]//Next Generation Internet Design and Engineering, 2006. NGI '06. 2006 2nd Conference. [s. l.]: [s. n.], 2006: 350 - 357. <http://ieeexplore.ieee.org/iel5/11058/34932/01678261.pdf?tp=&arnumber=1678261&isnumber=34932>.
- [2] Levy H, Zlatokrilov H. The Effect of Packet Dispersion on Voice Applications in IP Networks[J]. IEEE/ACM Transactions on Networking, 2006, 14(2): 277 - 288.
- [3] 姚玉坤, 刘合武. IntServ 与 DiffServ 在 VoIP QoS 中的应用分析[J]. 计算机与数字工程学报, 2007, 35(4): 55 - 58.
- [4] 周康, 裴楷, 吴宇红. VoIP 系统中消除时延抖动的研究与实现[J]. 电子科技, 2007(1): 51 - 54.
- [5] 王培明, 施寅. VoIP 中丢包隐藏技术研究[J]. 计算机技术与发展, 2006, 16(7): 26 - 28.
- [6] Hassan M, Alekseevich D F. Variable Packet Size of IP Packets for Voip Transmission[C]//Proceedings of the 24th Iasted International Multi-Conference Internet and Multimedia Systems and Applications. Innsbruck, Austria: [s. n.], 2006: 136 - 141.
- [7] 陈明义, 钱敏, 谭勇. 提高 IP 电话 QoS 的综合去抖动同步策略研究[J]. 计算机工程与设计, 2006, 27(23): 4991 - 4993.
- [8] 祝宇成, 周应权. 一种实时语音传输中冗余控制算法的改进[J]. 科学技术与工程, 2007, 7(17): 4476 - 4479.
- [9] 付强. RTP 协议及其 QoS 研究[D]. 南京: 南京邮电大学, 2006.
- [10] Augé J, Roberts J. Buffer Sizing for Elastic Traffic. Next Generation Internet Design and Engineering[C]//NGI '06. 2006 2nd Conference. [s. l.]: [s. n.], 2006: 33 - 40. <http://ieeexplore.ieee.org/iel5/11058/34932/01678220.pdf?tp=&arnumber=1678220&isnumber=34932>.

(上接第 34 页)

16th international conference on Supercomputing (ICS'02). New York: ACM Press, 2002: 84 - 95.

- [4] Joseph S, Hoshiai T. Decentralized meta-data strategies: Effective peer-to-peer search[J]. IEICE Transactions on Communications, 2003; 86(6): 1740 - 1753.
- [5] Sripanidkulchai K, Maggs B, Zhang H. Efficient content Location Using Interest-Based Locality in Peer-to-Peer Sys-

tems[C]//In Proceedings of IEEE INFOCOM. San Francisco, CA: IEEE Infocom on Computer and Communications Societies, 2003: 2166 - 2176.

- [6] Colomni A, Dorigo M, Maniezzo V. An investigation of some properties of an ant algorithm[C]//In: Proc. of the Parallel Problem Solving from Nature Conference (PPSN'92). Brussels, Belgium: Elsevier Publishing, 1992: 509 - 520.