面向时延优化的 Overlay 路由策略研究

叶 枰[1),2)] 李益忠[1),2)] 夏 勤[1),2)]

[1)](东南大学计算机网络和信息集成教育部重点实验室 南京 210096)

[2)](东南大学计算机科学与工程学院 南京 210096)

摘 要 路由扩展性是 Overlay 网络的研究热点,其中网络时延作为 Overlay 路由性能的一个关键指标已成为重点研究内容之一.文中对 Overlay 路由及其扩展性问题进行描述,并在此基础上进行数学建模和分析.针对时延优化目标,提出一种基于蚁群算法的单跳路由路径策略来处理邻居节点集合维护的问题:通过蚂蚁爬行的过程建立邻居节点集合,并根据设定的质量评估函数进行修剪来控制每个节点上维护的邻居节点集合大小.通过仿真实验对 Overlay 路由性能进行的分析,证实了 Overlay 路由在时延优化方面的可行性,同时实验结果表明了 ACOHPR 在对平均时延影响很小的情况下能够有效地降低每个节点的邻居节点集合大小.

关键词 时延优化;单跳路径路由;Overlay 路由;蚁群算法;邻居节点集合

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2010.00036

Delay Optimization Based Overlay Routing Strategy

YE Ping[1),2)] LI Yi-Zhong[1),2)] XIA Qin[1),2)]

[1)](*Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing* 210096)

[2)](*School of Computer Science and Engineering, Southeast University, Nanjing* 210096)

Abstract The routing scalability of Overlay network has become a research focus. The network delay is one important branch of the scalability field as a key parameter. This paper gives a description of Overlay routing and its scalability problem, on the basis of which the mathematical modeling are established. An improved one-hop path routing algorithm based on ant colony optimization (ACOHPR) is employed to solve the problem of maintaining neighborhood set in allusion to delay optimization. The neighborhood set is formed in the progress of ant crawl, which is pruned according to the quality evaluating function in order to control the size of neighborhood set in every node. By simulation the performance of Overlay routing is analyzed and its feasibility in delay optimization is validated. The simulation results also show that ACOHPR can effectively reduce the size of neighborhood set with little influence in average delay.

Keywords delay optimization; one-hop path routing; Overlay routing; ant colony optimization; neighborhood set

1 引 言

Overlay 思想在计算机网络发展过程中起到了极为重要的作用,从通信协议的分层设计到互联网在各种异构网络上的部署,这种“下层为上层提供服务,上层使用下层服务”的思想有效地降低了网络设计、部署的复杂性,为互联网上各种应用的顺利展开

收稿日期:2009-05-08;最终修改稿收到日期:2009-10-22 本课题得到国家自然科学基金重大研究计划项目(90604003)、国家自然科学基金项目(60603067)资助.**叶 枰**,男,1984 年生,博士研究生,主要研究方向为 Overlay 应用及 P2P 安全. E-mail: yping@seu.edu.cn. **李益忠**,男,1984 年生,硕士,主要研究方向为 Overlay 应用. **夏 勤**,男,1957 年生,高级工程师,主要研究方向为计算机网络、无线协议和 IPv6 技术.

提供了有力的保障. 近年来网络研究人员进一步拓展了 Overlay 的概念，通过在现有网络的应用层构建 Overlay 网络来实现新应用，满足用户的新需求. 这些系统的快速发展和普及，不仅改变了网络应用的形式和内容，同时也促进了互联网本身的变化和发展.

Overlay 路由是 Overlay 网络研究的重要组成部分，通过在现有网络的基础上组织起虚拟的通信结构可以有效地增强现有网络路由的可靠性，提高网络对不同应用需求的服务能力，文献[1-3]通过对 Overlay 网络拓扑连通性及节点之间的通信距离的建模说明了 Overlay 路由相对于原有 IP 路由的优势. Overlay 路由的性能问题更是研究重点，文献[4]提出了一种基于代价的模型来评估 Overlay 网络中资源使用代价，包括时延、带宽和节点计算能力等，并且为一些针对该网络的拓扑结构进行评定，而文献[5]则对网络中的可用带宽实时监控以实施动态的 Overlay 路由. 作为考察 Overlay 路由性能的一个重要方面，网络时延参数[6-7]被定义为数据从源端经过网络到达目的端所花费的时间. Detour[8] 研究人员已经发现：网络通信过程中，在网络中找到一个或者多个节点来转发数据，实现通信对端的数据“绕行”，可以获得比 IP 路由提供路径更好的通信效果. 研究结果发现通过“绕行”的方式，约 15% 的通信链接可以缩短时延达 25%，大于 15% 的链路可以将通信时延降低 25ms. 此外，这种工作方式还对通信吞吐率、丢包率等性能有明显的提高. 文献[9]对从 Planetlab 上收集的时延数据进行分析比较，结果显示 Overlay 方式最短路径路由策略可以将现有网络中不可达节点对比例从 9% 减少到 2%，同时 77% 的节点对能够改善其平均时延，从 220ms 减少到 181ms. 随后的研究中，人们开始希望 Overlay 网络可以在通过“绕行”提供更好通信性能同时为不同应用提供不同的性能满足支持. MIT 研究人员开发的 RON[10] 系统是对 Overlay 路由系统的最早实现方案，它主要通过实时的链路状态检测与状态信息维护，动态地在 Overlay 网络中找到一个中间节点转发数据，实现系统对网络链路失效的快速的检测和“绕行”.

在选取中间转发节点的过程中，路由表项的组织和查询会对系统工作效率产生直接的影响. 最短路径路由(Overlay based Shortest Path Routing, OSPR)[11-12] 策略给出了一个理想情况下的 Overlay 路由方案，系统中的每个节点通过交互获取相关的路由信息，计算整个网络中点对之间最短路径，构建自己的路由表. 该策略虽然能够保证获得整个网络节点对之间最优时延，但对网络本身提出了很高要求：节点之间有很好的同步能力，系统在特定时刻只对单一节点发出的消息进行处理；链路有较强的容错能力，不会突然失效或长时间出错. 这两个条件由于在网络环境下很难满足，极大地限制了该算法在 Overlay 网络中的应用.

本文首先对 Overlay 路由及其扩展性问题进行描述，并在此基础上进行数学建模和分析. 其次提出一种改进的蚁群算法来处理单跳路由转发策略中邻居节点维护的问题，通过在单跳路由系统中控制每个节点上维护的邻居节点集合大小来降低整个 Overlay 网络路由的管理开销以及查询开销. 最后通过仿真实验对 Overlay 路由扩展性能进行分析，证实了 Overlay 路由在时延优化方面的有效性，同时实验结果表明了 ACOHPR 在降低维护开销和运行开销的可行性和有效性.

2　面向时延优化的 Overlay 路由

2.1　问题描述

Overlay 路由是建立在现有网络基础上满足不同通信需求的路径选择过程. Overlay 路由系统的性能取决于两个很重要的因素：用户的应用需求和网络的具体工作状况. 本文从时延性能的角度讨论 Overlay 路由对现有互联网中网络连通性能的影响，首先引入系统工作的几点基本假设.

假设 1.　IP 路由次优性在广域网环境下客观存在.

本文假设 Overlay 路由系统工作在一个足够复杂的广域网环境下，具体表现为：IP 路由提供的单播路径在时延上未必为最优路径，可以通过 Overlay 方式的中间节点转发进行优化(减小通信时延).

假设 2.　Overlay 网络中的节点具有足够的数据处理和数据转发能力.

本文假设 Overlay 网络中节点具有足够能力来对数据进行处理和转发，系统性能的瓶颈主要存在于传输链路上，表现为不合理的路由结构以及路由控制策略. 针对 Overlay 网络中通信性能的时延参数，假定在节点上进行数据处理和数据转发的开销是很小的，相对于链路传输性能造成的影响可以忽略不计. Overlay 路由系统的主要目标就是利用 Overlay 网络中节点的处理能力来选择合理的通信路径，让数据通信沿着时延较小的路径进行，进而优化通信时延.

假设 3. Overlay 网络可以通过一定的检测机制来对网络链路时延状况进行评估.

本文假设链路时延状态可以通过多次实时测量中的最小值反映出来,随着通信路径上路由结构的变化而变化.

在上述假设基础上,考虑这样一个部署在现有互联网上的 Overlay 路由系统工作场景:Overlay 网络中的节点通过一定策略的检测来估量到网络中其他节点之间的通信时延状况,并且通过一定的方式来发送这些路由信息给其他节点.当节点接受其他节点发出的路由通告后,就可以根据这些通告来计算本地的路由信息.节点根据这些信息来动态地更新和维护链路信息库,通信之前通过查询来选择合适的转发节点进行数据传输.

定义 1. 单跳路径路由(One Hop Path Routing, OHPR). Overlay 网络中任意两点之间的通信最多只通过一个节点作为中间节点进行数据转发.

定义 2. 邻居节点集合(Neighbour Set). 每个节点维护的一个相对较小的节点集合,集合中的元素根据自己的网络状况从整个网络节点中选取.

定义 3. 时延优化(Delay Optimization). 对任意两点之间的通信时延进行优化.在本文中,时延优化的策略是采用 Overlay 路由对 IP 路由方式进行优化.

单跳路径路由作为 Overlay 路由方式的一种特例,实现了通信的"绕行",有效地减少传统多条转发路由策略中慢收敛、摆动等问题,但同时也意味着每个节点都需要实时地维护一个与网络规模同样大小的路由信息库支持路由选择,路径选择时需要查询一个与网络规模相当的路由表,这种工作方式使得整个系统在扩展性上受到了很大的限制.近年来的一些研究人员提出了使用 k 随机选择算法[6]来提高系统的扩展性,通过在一个小范围内进行选择转发来提高系统对失效链路的快速反应.但这种方式不能够完全满足对通信质量(时延、带宽等)有要求的情况.

邻居节点集合的方式缩小了每个节点的查询范围,即每个节点只需要根据网络状况维护一个相对较小的节点集合.发送数据前,首先查询邻居接节点集合中是否有节点可以对本次通信进行时延优化,如果有,则使用该节点进行数据转发;否则,使用原有 IP 路由进行通信.这样只需要为每个节点找到最合适的邻居节点集合就可以对整个 Overlay 路由通信性能以及维护开销进行改善.

因此 Overlay 系统中每个节点维护的邻居节点集合的大小可以作为考察系统维护开销的一个指标,系统扩展性问题就可以转变为考虑如何在保证时延优化效果的基础上降低节点维护邻居节点集合的大小.在仿真实验中将对该参数以及 Overlay 路由系统实现的时延优化作进一步的比较.

2.2 数学模型

将上述描述抽象成一个具体问题:在一个基于 OHPR 工作方式的 Overlay 路由系统中,如何在保证网络通信时延优化最大化的情况下,使得每个节点维护最小代价的邻居节点集合.由于整个网络的维护开销可以看作每个节点上维护开销的代数累加,首先考虑某一个节点 s 上的工作情况:假设 Overlay 系统中的节点集合为 $N=\{1,2,\cdots,i,\cdots,n\}$ 中,用一个 $n\times n$ 的矩阵 $\boldsymbol{A}=(a_{ij})$ 表示 Overlay 系统工作时选取节点 i 作为中间转发节点对节点 s 和节点 j 之间通信的时延优化程度,$\boldsymbol{A}$ 中所有元素非负;n 维非负向量 $\boldsymbol{cost}=(c_i)$ 表示将节点 i 放入本地邻居节点集合中的代价.那么根据工作场景描述,希望在节点 s 上找到这样一个子集 $NeighborSet_s$,使下面两个目标可以达到最优:

(1) 通过邻居节点实现对网络通信时延优化最大,即满足目标函数定义:

$$\max Z_1=\sum_{j\in N}\{\max_{i\in NeighborSet_s}(a_{ij})\}.$$

(2) 邻居节点的维护代价最小,用目标函数形式表示为

$$\min Z_2=\sum_{i\in NeighborSet_s}c_i.$$

采用限定法来对这个多目标优化问题的求解复杂度是满足 NP-hard 的.

证明. 设通信优化矩阵 $\boldsymbol{B}=(b_{ij})_{n\times n}$, $b_{ij}\in\{0,1\}$. b_{ij} 表示节点 i 是否可以对通信路径 s-j 进行优化,其中 $b_{ij}=1(i\neq j)$ 对应原问题中 $a_{ij}>0$ 情况.在这里假定对于所有节点 i 有 $b_{ii}=1$,表示节点可以对自己进行优化.

设代价矩阵 $\boldsymbol{C}=(c_i)_{1\times n}$, $c_i>0$. c_i 表示将节点 i 放入本地邻居节点集合中的代价,则原问题的第一个目标要求 $Z_1=n$,原问题就可以相应地转换成一个在 0-1 矩阵 $\boldsymbol{B}$ 上集合覆盖 SCP[14] 的问题,即在 0-1 矩阵 $\boldsymbol{B}$ 上找到一个列的集合使得矩阵的每一行都能被"覆盖"且选取列的代价总和最小.其中第 i 行被第 j 列覆盖当且仅当 $b_{ij}=1$.

转换后的问题可以用数学规划的模型来表示:

$$Zscp=\min\sum_{j\in N}c_jx_j,$$

$$\text{s.t.}\ \sum_{j\in N}b_{ij}x_j\geq 1,\ \forall i\in N,$$

$$x_j\in\{0,1\},\ \forall j\in N,$$

其中 $x_j=1$ 表示解中包含第 j 列，对应原问题在邻居节点集合中，否则 $x_j=0$.

由于 SCP 问题的求解复杂度是已被证明 NP 难的，而 SCP 问题对应了原问题的一个特例，由限制法就可以证明原问题为 NP 难问题，无法在多项式时间里求解.　　证毕.

3　基于蚁群算法的 Overlay 路由策略

3.1　蚁群算法简介及特性分析

蚂蚁系统(Ant System，AS)[15]最早是由意大利人 Doringo 等人提出的，该算法是从蚁群觅食过程中得到启发而构造出的一种模拟进化算法. 蚁群算法最早被应用在图论中求解旅行商问题(Traveling Saleman Problem，TSP)[16]. 随着人们对蚂蚁系统的深入研究和讨论，蚁群算法被广泛应用于优化问题的求解中：组合优化问题，数据挖掘中的规则提取、聚类分析以及解决网络中的 QoS 路由问题等，这些工作为许多 NP 难问题的近似求解提供了可行的工程解决方案. 从蚁群算法的原理分析可以看到蚁群算法在处理组合优化问题中体现出来的几点优势.

(1) 正反馈特性

蚁群算法与其他进化算法的一个很大区别就在于算法自身是通过正反馈的方式来实现对问题的逐步优化和求解. 前面的蚂蚁通过在经过路径上留下信息素来指导后面蚂蚁的工作，后面的蚂蚁依据前面蚂蚁留下的信息素作为自己选择路径的参考. 这种工作方式一方面可以通过信息素的不断叠加加快结果的收敛，另一方面需要防止蚁群算法中的由于信息素过多产生的凝滞问题.

(2) 较高的并行度

由于蚁群算法很好地模拟了现实生活中蚁群寻食的过程，因此算法本身的并行度较高，有利于系统在分布式的环境下的实现和应用. 本文使用蚁群算法在每个独立节点上可以实现互不干扰的计算. 针对本文中所需处理的问题，蚁群算法的这一特性可以很好地适应单跳路由方式，同时并行地对网络路由信息进行计算.

(3) 较好的健壮性

相对于其它算法，蚁群算法对初始路线的要求不高，初始条件的设置对将来结果的取得不会有太大的影响，蚁群算法的搜索结果不依赖于初始线路的选择. 在不需要对蚂蚁的爬行进行人工的干预情况下可以完成从初始化到得到搜索结果的整个计算过程，对于不同的网络环境只需要调整几个控制变量就可以很容易地实现蚁群算法对环境的快速适应.

3.2　算法伪代码描述

针对文中 2.1 节的分析，本文提出了一种基于蚁群算法的单跳路由路径算法(Ant Colony Optimization based one-Hop Path Routing，ACOHPR)，在详细介绍蚁群算法工作流程之前，首先介绍算法中几个重要的数据结构和参数.

(1) 启发因子矩阵 $\boldsymbol{\eta}$

规模为 $N\times N$ 的启发因子矩阵是蚂蚁选择邻居节点时的启发因子依据. $\eta_{ij}(t)$ 表示 t 时刻邻居节点中已经有节点 i，考虑加入节点 j 的权重. 这个值主要用于节点 i 和节点 j 共同出现在邻居节点集合中时的收益程度，为人工蚂蚁选择邻居节点作出参考. 在传统的 TSP 问题求解过程中启发因子参数通常都是不变参数，而本文考虑的网络时延状态会随着时间的变化而产生扰动和跳跃，并把网络的这一影响体现在启发因子的改变中.

(2) 信息素矩阵 $\boldsymbol{\tau}$

规模为 $N\times N$ 的信息素矩阵. 和启发因子一样，信息因子也是用于指导蚂蚁寻找路径，并且由蚂蚁根据自己的路径信息来对信息素因子进行更新. $\tau_{ij}(t)$ 表示系统在 t 时刻蚂蚁经过节点 i 和节点 j 后留下的与质量相关信息，用于启发后面蚂蚁在组合路径时选择下一跳节点.

(3) 优化度矩阵 ***improvement***

规模为 $N\times N$ 的优化度矩阵用于辅助系统工作时对优化性能和优化质量的计算. $improvement_{ij}(t)$ 表示通过节点 i 进行数据转发对通信对端 (s,j) 之间通信的优化程度，该参数是可以随着 Overlay 网络路由信息的更新而更新的.

(4) 覆盖度矩阵 ***improvable***

规模为 $N\times N$ 的覆盖度矩阵为一个 0-1 矩阵，反映节点转发对路径优化的可行性. $improvable_{ij}(t)$ 表示 t 时刻中间转发节点 i 对通信对端 (s,j) 进行通信优化的可行性，可行则值为 1. 根据矩阵的数学含义，若对矩阵的第 i 行求和表示节点 i 作为中间节点对源节点 s 优化的覆盖度. 该矩阵内容与优化度矩阵相对应，因此，可以直接根据 ***improvement*** 矩阵计算出来.

(5) 禁忌搜索节点集合 *allowset*

用于记录系统当前状态下蚂蚁尚未访问的节点

集合,在系统工作时主要用于实现对节点选择范围的控制.

每个节点上详细的算法流程如下:

1. 根据系统测量的网络时延信息来初始化计算辅助信息(启发因子矩阵 $\boldsymbol{\eta}$ 和信息素矩阵 $\boldsymbol{\tau}$). 同时设置两个全局变量 *NeighborSetRec* 和 *QualityRec* 分别用于记录最优邻居节点集合和最优邻居节点集合评估质量,初始两者皆为空.

2. 每个节点上生成一定数量的人工蚂蚁,即将 m 只蚂蚁置于节点 n 上,禁忌搜索节点集合 *allowset* 设置为整个节点集合,表明初始情况下尚未访问任何节点.

3. 每只蚂蚁按照一定的策略构建一个备份邻居节点集合 *NeighborSet*,主要通过计算 t 时刻蚂蚁所选的备份邻居节点集合中加入节点 i 后再加入节点 j 的概率 $P_{ij}(t)$来进行构建. 邻居节点加入过程的控制是通过禁忌列表 *allowset* 来实现的,*allowset* 可以防止向邻居节点集合中加入重复的节点并且保证加入过程的结束.

4. 在备份邻居节点集合 *NeighborSet* 形成之后,系统按照一定的质量控制要求对该集合进行修剪,从而优化邻居节点集合的结构并减少邻居节点集合中的重复覆盖情况.

5. 对所形成的邻居节点集合进行质量评估 *Quality*,以此更新对应节点的信息素矩阵 $\boldsymbol{\tau}$,用于指导后继蚂蚁的邻居节点集构建工作.

6. 重复步 3 直至步 5,直到该节点上的所有蚂蚁完成遍历.

7. 系统记录质量最高的邻居节点集合作为近似最优解 *NeighborSetRec*,算法结束.

图 1 简单描述了在一个节点 s 上蚂蚁系统工作的主要流程.

```
算法 1.  ACOOHPR.
Input: Delay                //时延关系矩阵
Output: NeighborSetRec //邻居节点集合
{
   NeighborSetRec=∅; //记录最优邻居节点集合
   QualityRec=0;       //记录最优邻居节点集合评估质量
   Initialization;
   /*初始化启发因子矩阵、初始化信息素矩阵*/
   While(i<Ant_Generation){        //每只蚂蚁的工作
        NeighborSet=∅;
        NeighborSet_s=GetNeighborSet();      //通过蚂蚁爬行建立一个邻居节点集合
        NeighborSet=Pruning(NeighborSet_s); //对建立起的邻居节点集合根据质量公式进行修剪
        UpdatePheromone(NeighborSet);       //更新路径上的信息素;
        if(Quality(NeighborSet)>Quality(NeighborSetRec)){ //更新最优邻居节点记录
             NeighborSetRec=NeighborSet;
             QualityRec=Quality(NeighborSet);
        }
        i=i+1;
   }
}
```

图 1 蚁群算法伪代码描述

3.3 启发因子及信息素的初始化

在人工蚂蚁选择爬行路径、构建邻居节点集合的过程中,启发因子和信息素是影响计算结果的重要因素. 通常启发因子的设定主要考虑的是环境因素对路径选择过程所产生的直接影响,如 TSP 问题求解中采用与路径长短相关的参数来作为启发因子参数. 而针对本文所考虑问题的特殊性,启发因子的设定主要是从两个方面来作为权衡.

定义 4. t 时刻节点 j 对节点 i 的相对优化率(relative improvement rate),为系统引入节点 i 作为邻居节点后再引入节点 j 所获得的相对优化性增加率. 记作 $RIR_{ij}(t)$:

$$RIR_{ij}(t)=\Big(\sum_{k\in N}(\max(improvement_{ik}(t),improvement_{jk}(t))-improvement_{ik}(t))\Big)\Big/\sum_{k\in N}improvement_{ik}(t) \quad (1)$$

定义 5. 节点 j 对节点 i 的相对覆盖率(relative coverage rate),为系统引入节点 i 作为邻居节点后再引入节点 j 所获得的相对优化覆盖范围增加率. 记作 $RCR_{ij}(t)$.

$$RCR_{ij}(t)=|(improvable_i(t)\bigcup improvable_j(t))-improvable_j(t)|/|N| \quad (2)$$

式(1)和(2)分别从优化程度和覆盖程度两个方面来对两个节点在同一个邻居节点集合中出现的性能进行评估. 采用两者的乘积来表示 t 时刻从节点 i 到节点 j 的启发因子$\eta_{ij}(t)$的取值大小:

$$\eta_{ij}(t)=RIR_{ij}(t)\times RCR_{ij}(t) \quad (3)$$

式(3)中主要考虑了节点 i、j 同时出现在邻居节点集合中时的收益期望. 这个参数的求解通过归一化方法来减小取值范围对最后结果的影响. 初始时刻,系统中所有信息素参数都设置为 $\tau_{ij}(0)=1$.

3.4　邻居节点集合的创建

人工蚂蚁创建邻居节点集合的过程可以看作是一个选择优化组合的过程，采用类似 TSP 的方法来描述蚂蚁的工作：蚂蚁从源点 s 出发，不断地选择新节点加入到自己的邻居节点集合. 当蚂蚁向邻居节点集合中加入节点 i 后，蚂蚁的当前状态就转变为 S_i. t 时刻蚂蚁选择下一个邻居节点 j 进入状态 S_j 的转移概率由 $P_{ij}(t)$ 来决定，直到蚂蚁邻居节点集合加入过程结束.

采用如下的公式来指导人工蚂蚁进行路径选择，当蚂蚁所选的邻居节点集合中加入节点 i 后再加入节点 j 的概率为

$$P_{ij}(t)=\begin{cases}\dfrac{\eta_{ij}(t)^{\alpha}\times\tau_{ij}(t)^{\beta}}{\sum\limits_{j\in allowset}\eta_{ij}(t)^{\alpha}\times\tau_{ij}(t)^{\beta}}, & j\in allowset\\ 0, & j\notin allowset\end{cases}\tag{4}$$

式(4)通过两个方面来帮助蚂蚁寻找好的邻居节点：(1)局部性质的启发因子，查看当前状态下局部比较好的选择；(2)信息素因子，为前面蚂蚁全局计算后留下的参考信息. 其中权重因子 $\alpha,\beta\geq 0$，用于调节启发因子和信息素在节点选择过程中的相对重要性. 这两个参数的选择主要是根据网络变化情况来调整的，在网络状况变化剧烈的情况可以适当地调高 α 值来让网络快速适应网络时延变化，β 主要作用是突出信息素的作用，促进计算的快速收敛.

邻居节点加入过程的控制是通过禁忌列表 *allowset* 来实现的. *allowset* 可以防止向邻居节点集合中加入重复的节点并且保证加入过程的结束. 此外，为了防止系统产生过大的邻居节点集合，采用限定邻居节点集合大小方式缩短蚂蚁的爬行过程.

3.5　邻居节点集合的修剪

当系统中的一只蚂蚁完成爬行后，需要对所得到的邻居节点集合进行适当修剪. 修剪的作用主要体现在两个方面：

(1)优化邻居节点集合的结构. 由于所求目标为一个无序集合，而在蚂蚁系统中却是按照一定的次序加入邻居节点，需要通过对所得邻居节点集合的修剪产生更多种的组合情况，并根据其质量影响系统信息素矩阵，最终指导后继蚂蚁的爬行过程.

(2)减少邻居节点集合中的重复覆盖情况. 修剪工作另外的一个重要作用就是降低邻居节点中的重复覆盖情况，通过在覆盖质量(覆盖率以及优化效果)和节点集合大小之间的权衡来进一步优化计算结果.

采用如图 2 中所示的算法对所得到的邻居节点集合进行修剪.

算法 2.　Pruning.

```
Input: NeighborSet_s
Output: NeighborSet
{
    QualityImproveable=true; //质量改进标记
    QualityRec=Quality(NeighborSet_s);
    indexRec=-1;
    while (QualityImproveable==ture){
        for i in range(NeighborSet_s){
            Temp=NeighborSet_s.removeItem(i); //每次尝试去掉一个引起质量增加最大的元素
            if(Quality(Temp)>QualityRec){      //记录产生最大质量的元素以及下标
                QualityRec=Quality(Temp);
                indexRec=i;
                QualityImproveable=true;
            }
            else{
                QualityImproveable=false;
            }
        }
        if( QualityImproveable)
            NeighborSet_s=NeighborSet_s.removeItem(indexRec);
    }
    NeighborSet =NeighborSet_s
}
```

图 2　邻居节点修剪算法伪代码描述

这一修剪过程反映了系统在邻居节点集合大小与整个邻居节点集合优化质量之间的权衡考虑：系统每轮删除一个使邻居节点集合质量下降最快的节点，直到节点集合不能再被修剪时返回.

3.6　邻居节点集合质量评估

从算法描述中可以看到，邻居节点质量函数的选择会对结果产生很大的影响. 为了使邻居节点质量函数可以有效地表达问题求解的目标，设计质量

评估函数时重点考虑以下几点因素给质量带来的影响:邻居节点可覆盖的节点数、邻居节点集合提供时延优化的程度和邻居节点的规模.采用式(5)来对邻居节点集合的质量进行评估:

$$Quality(S)=CoverageEva(S)[K_1\times improvementEva(S)+K_2\times SizeEva(S)] \quad (5)$$

其中 $CoverageEva$ 表示对邻居节点集合覆盖程度的评估,$improvementEva$ 为邻居节点集合时延改进状况的评估,$SizeEva$ 为节点集合大小的评估,K_1,K_2为权重因子.

3.7 信息素更新策略

按照修剪规则对蚂蚁获得的邻居节点集合进行修剪后,根据集合质量对蚂蚁走过"路径"上的信息素进行局部的更新.在计算过程中采用了简化的更新方式:把"路径"理解为邻居节点集合中节点的顺序排列,并依照这条路径上节点的次序,采用式(6)按照邻居节点集合的顺序依次对信息素进行更新.

$$\tau_{ij}(t)=\rho\times\tau_{ij}(t-1)+\frac{Q}{1+Q}\tau_{ij}(t-1) \quad (6)$$

其中 ρ 为信息素的挥发系数,Q 为邻居节点集合的评估质量.

4 仿真实验及结果分析

Meridian[①] 数据集是由 Cornell 大学计算机科学系 Meridian 项目组采集并发布的,主要用于通过网络时延测量实现轻量级网络定位系统的研究.数据记录了随机选取的 2500 个节点之间从 2004 年 5 月 5 日到 13 日的链路时延值,采集过程重复 10 次,数据采用上三角矩阵方式组织,数据单位为 μs.该数据集测量了大规模节点之间的相互时延,能代表足够复杂的网络连接情况.实验数据分析如表 1 所示.

表 1 实验分析数据集合相关参数

	总链路数/条	平均时延/ms	失效链路数	测量时间	测量方案
Meridian	2500×2500	167.3068	892	04/5/5～04/5/13	King Method

随机选择 Meridian 数据中 100 个节点,共计 10000 条链路进行仿真实验,比较根据 OHPR 构建路由后整体网络链路时延(统计区间增量为 10ms)的分布情况.横轴表示链路时延,纵轴表示链路的计数.

从图 3 中可以看出在 Meridian 数据表示的网络环境中,和原有网络链路时延分布状况(图中实面积标出部分)相比,Overlay 路由系统(图中线条标出部分)可以有效地增加通信时延在 80ms 左右的链路数目.

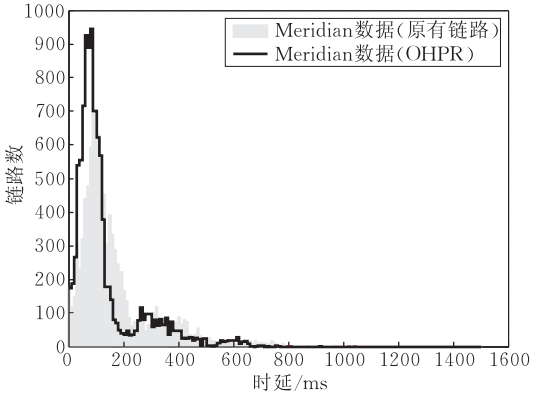

图 3 时延分布比较

为了验证本文所提出算法的有效性与合理性,在 Meridian 数据集合上选择 500 个节点构成 Overlay 网络进行仿真实验.假定 Meridian 数据提供的时延数据为某一时刻网络检测到各条链路的时延状态,在此基础上各个节点开始构建邻居节点集合.

实验中蚁群算法参数选择如下:

(1) 蚂蚁数量:$m=40$;

(2) 信息素挥发系数:$\rho=0.1$;

(3) 权重因子 $\alpha=1$,$\beta=2$;

(4) 权重因子 K_1,K_2 分别为 0.3,0.8.

实验过程使用蚁群算法计算了 Overlay 网络中每个节点的邻居节点集合大小以及邻居节点集合对整个网络通信时延优化情况,并将 OHPR 的计算结果在同一张图中进行比较,结果如图 4、图 5 所示.

图 4 比较了 Overlay 网络中每个节点在两种路由策略下对应维护的节点集合规模大小,横轴为节点编号,纵轴为该节点上维护的邻居节点集合大小,其中方块点代表了在单跳路径路由方式 OHPR 中每个节点需要维护的节点集合大小,叉状点代表了本文提出的基于蚁群算法的单跳路由路径策略中每个节点需要维护的节点集合大小.从中可以看出使用蚁群算法的系统优化可以有效地降低 OHPR 策略的邻居节点大小,即对于规模为 500 个节点的 Overlay 路由系统,大多数节点只需要维护小于 30 个的邻居节点就可以支持在整个 Overlay 网络的通信优化查询.

① Cornell University. Meridian data [OL]. http://www.cs.cornell.edu/People/egs/meridian, 2004

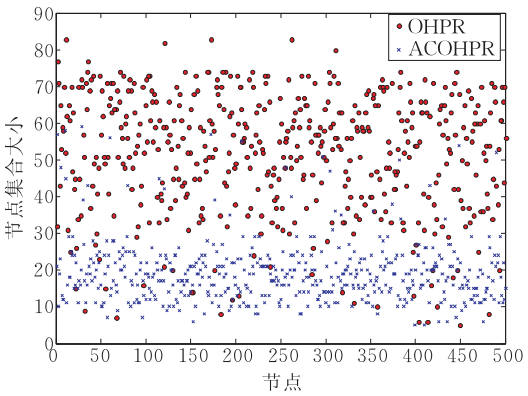

图 4　邻居节点集合大小比较

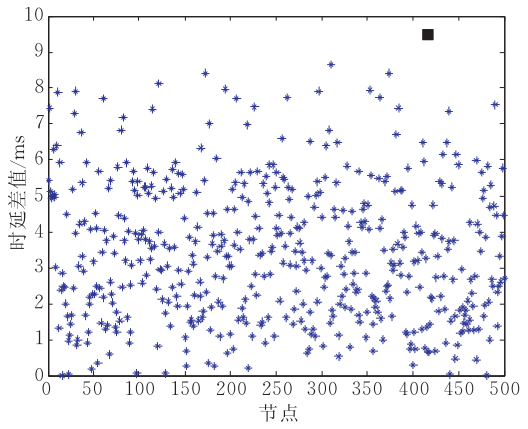

图 5　平均时延优化效果比较

在图 5 中描绘了每个节点在两种路由策略下获得网络通信时延的平均值的差值情况，横轴为节点编号，纵轴为两种路由方式下获得平均通信时延改进的差值. 由于 OHPR 为单跳转发情况下的时延优化最优解，因此时延差值取值非负. 从中可以看到 OHPR 与本文提出的基于蚁群算法的单跳路由路径策略所获得的网络平均通信时延差别很小，最大为 9.483ms，即图中黑色点坐标(416，9.483)处，85%的节点在两种路由策略下的通信时延介于[2ms，6ms]区间中，平均值只有 3.4ms 左右.

从实验结果比较来看，本文提出的基于蚁群算法的单跳路由路径策略基本实现了预期目标，即在在保证时延优化效果的基础上，本算法维护更小的邻居节点集合，降低系统整体的运行开销.

5　结束语

面向时延优化的扩展性问题一直是 Overlay 路由系统设计的关键问题之一. 本文首先对 Overlay 路由过程进行了描述，分析了其中存在的扩展性问题，随后对该问题进行数学建模. 在此基础上本文提出了一种基于蚁群算法的单跳路由路径策略来处理邻居节点集合大小的问题，通过控制每个节点上维护的邻居节点集合大小来降低整个 Overlay 网络路由的管理开销以及查询开销. 仿真实验首先验证了 Overlay 路由在时延优化方面的有效性，同时结果表明在 500 个节点的 Overlay 系统中，在与 OHPR 算法获得的网络通信时延平均值差别很小的情况下，本文提出的 ACOHPR 算法能够大幅降低每个节点维护的邻居节点集合大小，大约在 22 个左右.

未来的工作包括以下两个方面：一方面是要考虑 ACOHPR 在真实互联网环境中的部署，并对其性能及优化效果进行评估，Planetlab 为下一步工作提供了很好的实验平台. 另一方面，下一步工作需要综合考虑时延、带宽、节点处理能力等因素，对 Overlay 路由扩展性问题进行多方位权衡，使得在改进系统性能的同时获得良好的可扩展性.

参　考　文　献

[1] Loguinov D, Casas Juan, Wang Xiaoming. Graph-theoretic analysis of structured peer-to-peer system: Routing distances and fault resilience. IEEE/ACM Transactions on Networking, 2005, 13(5): 1107-1120

[2] Chun B, Fonseca R, Stoica I et al. Characterizing selfishly constructed Overlay routing networks//Proceedings of the IEEE INFOCOM. Hong Kong, China, 2004: 1329-1339

[3] Gummadi K, Gmmadi R, Gribble S et al. The impact of DHT routing geometry on resilience and proximity//Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocol for Computer Communication. Karlsruhe, Germany, 2003: 381-394

[4] Christin N, Chuang J. A cost-based analysis of Overlay routing geometries//Proceedings of the IEEE INFOCOM. Miami, USA, 2005: 2566-2577

[5] Zhu Yong, Dovrolis Constaintinos, Ammar Mostafa. Dynamic Overlay routing based on available bandwidth estimation: A simulation study. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2006, 50(6): 742-762

[6] Akella Aditya, Pang Jeffrey, Maggs Bruce et al. A comparison of Overlay routing and multihoming route control. ACM SIGCOMM Computer Communication Review, 2004, 34(4): 93-106

[7] Vleeschauwer B D, Turck F D, Dhoedt B et al. On the construction of QoS enabled Overlay networks//Proceedings of the 5th International Workshop on Quality of Future Internet Services. Barcelona, Catalonia, Spain, 2004: 164-173

[8] Savage Stefan, Anderson Thomas, Aggarwal Amit et al. Detour: Informed Internet routing and transport. IEEE Micro, 1999, 19(1): 50-59

[9] Zhang H, Tang L, Li J. Impact of Overlay routing on end-to-end delay//Proceedings of the 15th International Conference on Computer Communications and Networks. Boston, Massachusetts, USA, 2006: 435-440

[10] Nakao Akihiro, Peterson Larry, Bavier Andy. Scalable routing Overlay networks. ACM SIGOPS Operating Systems Review, 2006, 40(1): 49-61

[11] Lua Keong, Croweroft Jon, Pias Marelo et al. A survey and comparison of peer-to-peer Overlay network schemes. IEEE Communication Surveys & Tutorials, 2005, 7(2): 72-93

[12] Zhao Ben Y, Kubiatowicz John, Joseph Anthony D. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. University of California Berkeley: Technical Report: CSD-01-1141, 2001

[13] Gummadi Krishna P, Madhyastha Harsha V, Gribble Steven D et al. Improving the reliability of internet paths with one-hop source routing//Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation. San Francisco, CA, USA, 2004: 13-19

[14] Lessing Lucas, Dumitrescu Irina, Stutzle Thomas. A comparison between ACO algorithms for the set covering problem//Proceedings of the ANTS, 2004: 1-12

[15] Merkle Daniel, Middendorf Martin. Modeling the dynamics of ant colony optimization. Evolutionary Computation, 2002, 10(3): 235-262

[16] Engebretsen Lars, Karpinski Marek. TSP with bounded metrics. Journal of Computer and System Sciences, 2006, 72(4): 509-546

YE Ping, born in 1984, Ph. D. candidate. His research interests include overlay applications and P2P security.

LI Yi-Zhong, born in 1984, M. S.. His research interests focus on overlay applications.

XIA Qin, born in 1957, senior engineer. His research interests include computer network, wireless protocol and IPv6 technique.

Background

Overlay network plays an important part in the progress of the computer network, while routing scalability is one of the most important problems in Overlay network. The network delay is one important branch of the scalability field as a key parameter. Nodes in Overlay network can forward the messages in the progress of Overlay routing in order to optimize the communication delay. Generally the routing strategy is classified into two main classes: Overlay based Shortest Path Routing (OSPR) and One-hop Path Routing (OHPR). Compared with OSPR, OHPR can efficiently resolve the problem of slow convergence and swing. However, in OHPR strategy, every node maintains a routing table with the same size of the network scale, which means a large neighbor set should be queried in the progress of message forwarding. This paper is focused on reducing the size of the neighbor set as well as ensuring the maxim degree of delay optimization.

This paper gives a description of Overlay routing and its scalability problem, on the basis of which the mathematical modeling are established. This paper proposes an improved one-hop path routing algorithm based on ant colony optimization (ACOHPR). Simulations include: Analyzing the performance of Overlay routing and validating its feasibility in delay optimization. It also shows that the proposed algorithm ACOHOR can effectively reduce the size of neighborhood set with little influence in average delay.

This research is supported by the National Natural Science Foundation of China under grand Nos. 90604003 and 60603067.