第32卷 第13期 计 算 机 工 程 2006年7月
Vol.32 № 13 Computer Engineering July 2006
·网络与通信· 文章编号：1000—3428(2006)13—0103—03 文献标识码：A 中图分类号：TP393

支持 QoS 的覆盖组播分布式动态路由研究

张 丽[1,2]，余镇危[1]，张 扬[3]，李 宁[1]
(1. 中国矿业大学研究生院，北京 100083；2. 河南理工大学应用数学与信息科学系，焦作 454003；3. 空军第一航空学院，信阳 464000)

摘 要：研究了带度和延时约束的覆盖组播动态路由问题，提出了动态适应性覆盖组播路由协议 OMP，给出了一个基于分布式触发重组的组播路由算法——动态覆盖组播路由算法(DDCOMR)，最后对该算法的复杂度进行了推证，对协议和算法的有效性进行了网络模拟。
关键词：覆盖组播；度和延时约束；动态路由；分布式触发重组

QoS-based Distributed Dynamic Routing of Multicast Overlay Network

ZHANG Li[1,2], YU Zhenwei[1], ZHANG Yang[3], LI Ning[1]
(1. Graduate Student College, China University of Mining and Technology, Beijing 100083; 2. Department of Applied Mathematics and Information Sciences, Henan Polytechnic University, Jiaozuo 454003; 3. The First Aeronautic Institute of Air Force, Xinyang 464000)

【Abstract】 This paper studies the degree and delay constrained overlay multicasting dynamic routing problem, and proposes a new dynamic adaptable overlay multicasting protocol called OMP, and an arithmetic DDCOMR according to the distributing triggers the reorganization. In the end, the paper calculates the complexity of the arithmetic, and simulates the OMP.
【Key words】 Overlay multicast; Degree and delay constrained; Dynamic routing; Distributed triggered reorganization

1 概述

组播是一种由源节点可以同时向多个目的节点发送信息的通信方式，组播路由技术也是实时多媒体应用、计算机协同工作等新型分布式计算的关键技术之一。然而，由于技术和经济的双重原因，直到目前，全互联网范围内的 IP 多播服务尚未部署完成[1]。人们转而希望在应用层解决 IP 组播面临的问题，提出了覆盖组播(Overlay Multicast)的概念[2]。覆盖组播的思想是由系统而不是核心路由器实现多播通信的所有功能，其最大的优势在于无需改变现有的 IP 网络设施，可灵活部署，因而成为近年来的所热衷研究的问题。

但是，由于端系统的特点，覆盖组播也带来了新的问题。首先，端系统受 CPU 处理能力和接入带宽的限制，其能同时向下游转发数据流的最低分支数(度)十分有限，所以建立组播树时，必须考虑一个节点能支持的连接限制，而这将显著增加路由延时。其次，端系统具有较大的动态不稳定性，而任一端系统的离开或失效，都有可能引起路由分裂，这会降低路由性能并增加系统维护路由的开销。实时多媒体应用通常对数据传输的延时上限有严格的规定，以满足实时性要求。因此，构造带度和延时约束的覆盖组播路由是支持 QoS 的覆盖组播研究的热点问题之一。Shi 等[4]研究了具有度和延时约束覆盖组播生成问题，提出了一系列启发式路由算法。目标是在构造满足约束条件的组播树时，使端系统的剩余度趋于平衡，以使系统接纳更多的组播会话。但是提出的算法属集中式，需维护全局状态，复杂度高，并且该算法是会话级，并未考虑节点的动态性。分布式算法在可扩展性和动态适应性方面有优势，文献[3]采用了分布式算法来构造具有延时约束的覆盖组播树，目标是能尽可能多地接纳组成员，算法同时还考虑了端系统的动态性。但是，此算法是启发式局部优化算法，易陷入局部搜索使性能下降。研究表明，在端系统受度约束的情况下，这些方案一般都不能有效地解决具有延时约束的覆盖组播路由问题。

对于覆盖组播来说，由于事先无法预测那些端系统将会加入或离开组播组，因此，多点通信所特有的动态路由优化问题，在覆盖组播中将会更加复杂，而其研究成果也将对实际网络更具现实意义。有鉴于此，本文提出的覆盖组播路由协议(Overlay Multicasting Protocol，OMP)力图从动态适应性角度，以分布式触发的思想对上述问题进行研究和解决。

2 网络模型及问题描述

覆盖组播 QOS 路由问题的概念模型可以表示为网络 $G=(V,E)$，其中 v 是节点的集合，表示端系统，$E=V\times V$ 是边的集合，表示逻辑信道，每一逻辑信道对应于一条下层 IP 网络的物理路径。假设 IP 网络采用最短路径单播路由。

对 $\forall v\in V$, 节点有度约束值 $d_{\max}(v)\in N$，$d_{\max}(v)$ 表示节点 v 能同时支持的最大的媒体流转发分支数。例如，节点 v 的分组转发能力为 C，媒体流的平均带宽为 B，则 $d_{\max}(v)=\lfloor C/B\rfloor$，即节点 v 最多只能同时向 $d_{\max}(v)$ 个下游节点转发媒体流数据。对应每条边 $C(u,v)$，定义边权函数 $C:E\to R^{+}$，$C(u,v)$ 表示节点 u 到 v 的最小单播延时。$P_T(s,v)$ 表示组播树 T 中从源节点 s 到目的节点 $v\in V$ 的路径。以网络资源占用量(带宽×延时)作为

基金项目：国家博士点基金资助项目(20030290003)；河南理工大学自然科学(青年)基金资助项目(646138)
作者简介：张 丽(1973—)，女，博生，主研方向：新一代网络体系结构，覆盖网络关键技术及多面体主动式路由技术；余镇危，教授；张 扬，学士、助教；李 宁，硕士生
收稿日期：2005-08-04 **E-mail：**zhanglily86@126.com

组播树的代价，当带宽抽象为“度”时，新加入节点接入树上延时距离最近的节点，可使树的代价(增幅)最小。

定义 1 节点 v 在树 T 上的延时：$delay_T(v)=\sum_{e(u,v)\in P(s,v)} C(u,v)$，即从 s 到 v 沿 T 上路径的所有的有向边的代价之和。

问题 1 有度和延时约束的覆盖组播路由问题就是构造一棵以 s 为根的生成树 T，使得 $\forall v\in V$ 满足：

(1)度约束：$d_{used}(v)\le d_{\max}(v)$，其中 $d_{used}(v)$ 表示 v 已经用的度。

(2)时延约束：$delay_T(v)\le D_T$，D_T 为组播树的最大延时上限。

容易证明，问题 1 是 NP 完全的[4]。

定义 2 在 $G(V,E)$ 中，对于 $\forall(u,v)\in E$，$P(u,v)$ 为从节点 u 到 v 的可行路径，若 $P(u,v)$ 满足：

$d_{used}(v)\le d_{\max}(v)\wedge delay_T(v)\le D_T$。

定义 3 在 $G(V,E)$ 中，对于源节点 v，目的节点 $v\in V$，节点 s 与 v 之间代价最小的可行路径称为最优路径，用 P_v 表示。

3 动态覆盖组播路由算法(DDCOMR)

在其时间复杂度可以忍受的前提下，目前所提出的组播树生成算法大多是在寻求次优解，许多用启发式算法或者遗传算法、蚂蚁算法等求解组播路由问题的文章相继发表。然而应该认识到，启发式算法有其自身的缺陷，即只能找到局部最优，遗传算法和蚂蚁算法虽然都是全局算法，但是其编解码方法都比较复杂，个体编解码复杂度为 $O(n^2)$~ $O(n^3)$。所以，在此本文利用简洁的 Prüfer 编码，通过构造完全图的方式，把覆盖组播树“树核”的求解问题改为全组播路由问题来求解，以得到更快的通用覆盖组播网络组播树的遗传算法。

3.1 覆盖层组播“树核”生成算法

设网络 $G(V,E)$，V 是节点的集合，E 是边的集合，边权表示节点之间的组播代价。以 *Dijkstra* 算法由图 G 中的最短路构造其完全图 K_n。

完全图 K_n 的不同生成树共有 n^{n-2} 棵，正好可以用 n-2 位的 n-进制整数来表示，每位数字都在[1，n]之间，它对应为 K_n 的节点编号，我们可以方便的利用 Prüfer 编码对其进行调制，由 Prüfer 序列到生成树 T 的解码算法 *seqToTree* 如下：

算法 1 *seqToTree*

[算法开始]

输入：Prüfer 编码序列 P；

确定节点数 $n:=P$ 的长度+2；

统计每个节点在 P 中的出现次数 A[1..n]；

生成具有 n 个孤立节点的图 T，节点依次标记为 1,2,…,n；

令 Q 为 T 的所有不在 P 中的节点编号集合，并非减有序；

当 P 非空，循环做：

从 P 中移出第 1 个元素 v，从 Q 中移出第一个元素 u，A[v]:=A[v]-1；

在 T 中增加边<v,u>；

如果 A[v]=0，则折半插入 v 到 Q 中；

从 Q 中移出最后两个元素 v 和 u，在 T 中增加边<v,u>；

输出 T。

[算法结束]

由于 Prüfer 编码代表的生成树是无序树(不指定根)，而组播树需要指定根，因此我们扩充 Prüfer 编码为 n-1 位，最后一位代表根节点编号。另外，还需要对解码得到的 K_n 的生成树进行剪枝，删除不在组播终点集 D 中的叶子节点才能得到所求的组播树，剪枝算法 *prune* 的设计如下：

算法 2 *prune*

[算法开始]

输入：K_n 的生成树 T，组播终点集合 D;

对 T 做后根次序遍历，一边遍历一边做：

如果当前被遍历节点 v 是叶子节点并且不在 D 中，则：从 T 中删除 v；

输出 T。

[算法结束]

遗传算法设计如下：

染色体用扩充的 Prüfer 编码表示；给定具有 n 个节点的网络 G 和种群规模 P 后，初始种群由随机生成的 P 行 n-1 列的矩阵表示，其中的每个元素为[1，n]之间的整数，代表 G 中的节点编号；计算每个染色体(Prüfer 序列)的适应值时，依次用算法 *seqToTree* 和 *prune* 得到组播树 T，通过计算该个体的优化目标值 g，再对 g 做标定得到个体适应值 f=1/(1+g)；选择算子采用改进的一次旋转赌轮方法；交叉算子采用随机多点交叉；变异算子采用基因位变异和基因片变异(包括迁移和翻转)；保优策略，即每代最优秀个体直接进入下一代而不参与交叉和变异。在求得最优解后，得到组播源点集 S 和组播功能节点集 MV 的 steiner 树。再将此组播树按 G 中的最短路展开，并对相同路作归并操作。就得到了所求覆盖组播生成树的“树核”。

3.2 DDCOMR 节点动态加入算法

考虑到系统的可扩展性和动态适应性，DDCOMR 节点加入优化算法采用分布式动态算法，每个节点 $\forall v\in V$ 只需维护局部状态信息。节点的主要状态集为

$$\{ch\_list(v), s\_path(v), d_{\max}(v), d_{used}(v), delay_T(v), delay_p(v)\}$$

其中 $ch\_list(v)$ 表示 v 的孩子节点列表；$s\_path(v)$ 表示组播树上从根节点 s 到 v 的路径，记为根路径，用于检测回路；$d_{\max}(v)$，$d_{used}(v)$，$delay_T(v)$ 的定义同前；$delay_p(v)$ 表示 v 的父节点到 v 的延时，有关系式：$delay_T(v)=delay_T(v)+delay_p(v)$ (记 p 为 v 的父节点)。

3.2.1 节点动态优化加入

假设节点可以通过第三方机制(如汇聚点 RP)获得组播组根节点 s 的信息。当一个新的节点 v 要加入组播组时，首先令 s 为当前探测父节点(记为 p_n)。v 向 p_n 发送查询请求报文(Query)。p_n 收到请求报文后，立即将自己的孩子节点列表信息通过查询响应报文(QueryResp)返回给 v。v 获得响应报文后，对 p_n 及其孩子节点进行探测。令 CP_n 为 p_n 孩子节点集合，则 v 的候选父节点集合 $CP=\{p_n\}\cup CP_n$。$\forall u\in CP$，v 通过探测获得如下临时信息

$$\{C(u,v), d_{\max}(u), d_{used}(u), delay_T(u)\}$$

其中 $C(u,v)$ 为 u 和 v 之间的延时，可通过 ping 等延时测量工具得到，其他参数可利用捎带技术在测量延时同时得到。令

$$Valid\_CP=\{U\mid delay_T(u)+c(U,V)\le D_T, U\in CP\}-\{U\mid d_{\max}(u)=d_{used}(u), u=p_n\}$$

其中 $Valid\_CP$ 为有效候选节点集合。若 $Valid\_CP$ 为空，说明没有符合条件的候选节点；否则，从 $Valid\_CP$ 中选取一个最优(我们选用 $C(u,v)$ 最小)的候选节点(记为 p_0)。若 p_0 为当前的 p_n，则 v 直接向 p_0 发送加入请求(Join)；否则，v 令 p_0 为 p_n，发起新一轮查询。p_n 收到 Join 报文时，若其孩子数 $d_{used}(p_n)$

小于度约束 $d_{\max}(v)$，就接纳 v 成为自己的孩子节点，并返回捎带有 s_path 信息的加入确认报文(JoinAck)；否则，返回加入拒绝报文(JoinNAck)。节点 v 若收到 JoinAck 报文，则加入成功，并更新自己的 $s\_path(v)$，$delay_T(v)$，$delay_p(v)$ 等信息；若收到 JoinNAck 报文，则加入失败。

完整的节点加入算法如下：

[算法开始]

(1)通过汇聚点 RP 获得根节点 s，令 p_n=s；

(2)发送 Query 报文到 p_n；

(3)if 收到 QueryResp 报文 then

对 CP 中的节点进行探测，选择 *Valid_CP* 中局部最优的候选父节点 p_0。若 *Valid_CP* 为空，说明没有符合条件的候选节点，则加入失败，算法结束。

(4)if p_0=$p_n$then 发送 Join 报文到 p_0，等待响应；

else 令 p_n=p_0，转到步骤(2)；

(5)if 收到 JoinAck 报文 then

加入成功，更新 s_path，算法结束。

else if 收到 JoinNAck 报文 then

加入失败，算法结束。

[算法结束]

3.2.2 节点状态维护

受文献[5]的启发，OMP 协议采用交换报文来维护组成员之间的关系，即孩子和父亲节点之间周期性地交换刷新(Refresh)报文和路径(Path)报文，以交换和更新信息。孩子节点向父亲节点发送 Refresh 报文。父亲节点收到报文后，则向孩子节点发送带有 s_path 和 $delay_T$ 信息的 Path 报文以应答。孩子节点根据 Path 报文中的 s_path 信息更新自己的根路径信息。孩子节点还可根据报文交互的往返程时间(RTT)估算其与父节点之间的延时($delay_p$)，并利用 Path 报文中的 $delay_T$ 信息更新自己在树上的延时信息。这种邻接节点之间的分布式的信息交换能逐跳传播到整个组播组。

3.3 节点动态退出算法

针对端系统固有的随意性和不稳定性，我们采用软状态的方法来处理节点离开(如退出)和失效(如系统突然崩溃)。

当成员节点离开组播树时，需要同时分别向其父亲和孩子节点发送 leave 报文进行通告。其父亲节点就把这个节点从自己的孩子列表中删除，其孩子节点将重新运行 3.2.1 节中的加入算法。没有声明离开的成员节点离开就等效于节点失效。其父亲和孩子节点可通过 Refresh 或 Path 报文的丢失事件来发现。

3.4 环路检测与消除

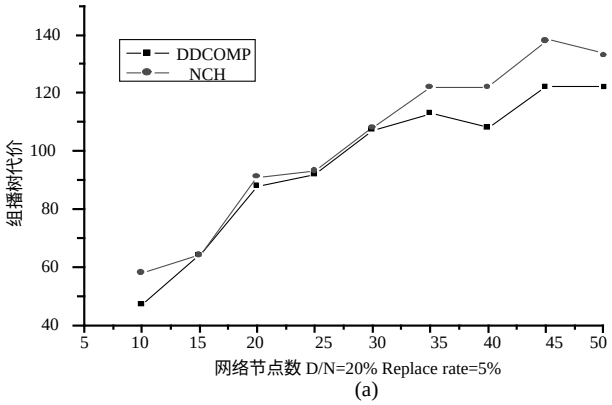
由于端系统的动态特性，可能造成节点状态不一致，从而产生路由环路(loop)。为解决环路问题，节点在更新 s_path 信息时要检测根路径上是否存在自己的标识。若存在，就说明已经产生了 loop，则立即向父亲节点发出 leave 报文，并开始重新加入组。

4 算法复杂度及网络模拟

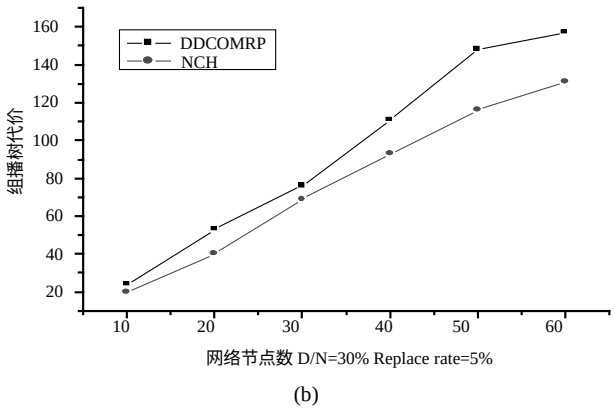
覆盖组播树核生成时，采用 Dijkstra 算法，各组播节点及源节点 s 间彼此最短路寻径，设端系统节点数为 m，网络中总的节点数为 n，则连通网络生成在最坏情况下时间复杂度为 $O(m^2n)$。而树核生成的遗传算法和 *seqToTree* 算法的时间复杂度为 O(P×maxG×O(mlogm)(其中 p 为群规模，maxG 为最大进化代数))。节点动态优化加入算法中，每次加入过程的控制报文开销为 $O(d\cdot\log_d\cdot t_T)$，其中 t_T 为当前树 T 上的节点总数，$d\in[1,\Delta]$，$\Delta=\max_{u\in T} d_{\max}(u)$，除了 d＝1 的最坏情况，加入算法的复杂度与 t_T 的对数成正比。

因为目前对于支持 QOS 的覆盖网络分布式动态组播树的生成算法还没有进行过系统的研究，所以也没有基准算法来用于网络模拟和比较。本文的网络模拟采用课题组覆盖网络静态组播树生成算法 NCH 来作为模拟基准。

以课题组 EAD 仿真模型所生成的模拟网络来做模拟平台，采用以下假设：加入组播组的端节点以一种稳定的速度进行更新，即某一时刻有占网络节点总数 5%的组播组成员退出组播组，而另有占网络节点总数 5%的非组播组成员加入组播组。边的代价值在[2，10]范围上均匀分布，在每个数据点上均运行 100 次随机实验，然后统计所有实验的平均值。而初始状态选择组播组成员占总节点数的 20%和 30%（分别如图 1 中的图(a)和图(b)所示），图 1 给出了网络节点从 5 变化到 50 和 10 变化到 60 的过程中，DDCOMR 算法和 NCH 基准算法所得组播代价值的比较。由模拟数据可知，DDCOMR 算法因为可以在节点加入组播组及节点退出组播组时都能触发树结构的部分重组。所以其费用能够被限定在基准以下的的区域内，在组播组成员比例进一步增加时，算法表现出很好的收敛性，这一结果还是令人鼓舞的。



(a)



(b)

图 1 网络大小变化时组播树的代价

5 结束语

OMP 协议采用了分布式触发的思想，无需覆盖层对整体的费用作出监控及评价。动态组播树生成算法(DDCOMR)减轻了路由优化的复杂性和额外开销，并能最大限度地减小算法的散列跃度，算法收敛性好。因为是基于“树核”的，所以算法可以在一定路由协议的支持下进行多面体分级路由模式的改良，也可以方便地向多点对多点组播路由算法改进，使算法具有较好的可扩展性。同时，DDCOMR 避免了启发式

（下转第 113 页）

而有效地提高副本定位的查询性能。此外，位于 LRI 的局部副本目录缓存和位于 GRI 的全局副本目录缓存也能够显著提高副本定位的性能。特别的，由于每个 VO 内部节点的相关性，使得它们访问数据文件的时间局部性非常强，因此设置在 LRI 的局部副本目录缓存可以发挥显著的作用。

(2)可扩展性

当数据网格规模扩大时，SN 和 LRI 的数目相应增多，此时需要增加 GRI 的数目来达到提高性能的目的。由于 GRI 之间采用 Chord 算法组成对等网，每个 GRI 节点仅需维护 O(logN)个临近节点的信息(N 为 GRI 总数)，并且每次查询仅需经过 O(logN)跳即可找到目标 GRI，因此系统具备良好的可扩展性。

(3)灵活性

本文提出的副本定位机制具有较强的灵活性。每个 VO 可以设置一个或多个 LRI，每个 LRI 也可以自由选择与之相连的 GRI，这样可以根据具体的需求调整 LRI 的数量及其与 GRI 的连接关系以满足不同的副本定位需求。此外，位于 GRI 的全局副本目录缓存通常情况下只保存(LDN, LRI)映射，但在存储资源允许的情况下也可以直接缓存(LDN, SN, timeout, QoS_Params)信息，从而进一步提高查询性能。虽然目前我们采用 Chord 算法将 GRI 组成 P2P 覆盖网络，但该网络内部的搜索算法跟 LRI 及 SN 是独立的，因此可以在不影响 LRI 和 SN 连接方式和调用服务方式的情况下，对算法进行升级或调整，进一步提高系统的性能、可扩展性和可靠性。

(4)可靠性

Chord 算法会随着新节点加入或退出(失效)自动调整其结构，即使用 Chord 算法组成的系统处于不断变化的状态，算法也会确保总有节点对查询请求进行响应[5]，因此 Chord 算法本身具有较强的可靠性。同时还可以采取为 VO 增加 LRI 的方法提高 SN 到 LRI 的可靠性。对于由 GRI 组成的 P2P 覆盖网络，可以对关键节点的副本目录进行复制，提高系统的可靠性。

3.2 模拟结果

我们在一台 Pentium 4 计算机(CPU 2.4GHz，内存 512MB)上对 GRI 组成的 P2P 覆盖网络的性能进行了模拟。用运行在 Java 虚拟机上的独立进程来模拟 GRI 节点和 LRI 节点，它们之间采用 socket 通信，模拟中没有包括全局和局部副本目录缓存。我们分别对 2～16 个 GRI 节点、每个 GRI 上分别存储 100 000 条和 1 000 000 条全局副本目录记录共计 16 种情况进行了模拟。对上述每种情况各进行了 1 000 次查询操作，得到 GRI 查询操作的平均响应时间和上下界，如图 3 所示。

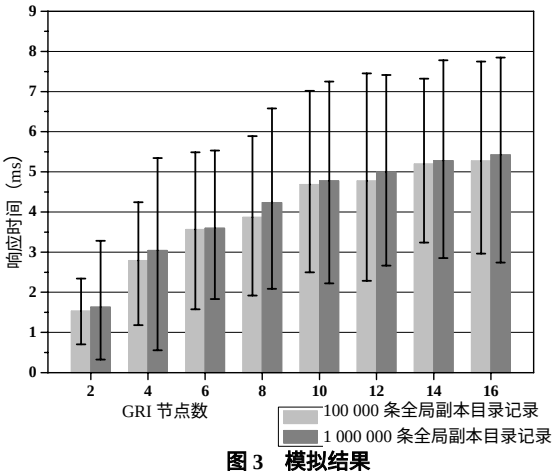


图 3　模拟结果

从上述结果中，可以得出两个结论：(1)我们提出的副本定位机制具有较好的性能；(2)随着 GRI 节点数量的增加，平均响应时间仅成对数关系增加，系统具有良好的可扩展性。

4 结论

针对现存的数据网格副本定位机制的不足，本文提出了一种基于 P2P 覆盖网络的数据网格副本定位机制。本机制采用 Chord 算法将全局副本目录节点组成一个 P2P 覆盖网络，从而起到均衡负载、提高性能和可靠性的目的。本机制具有良好的性能、可扩展性、灵活性和可靠性，能兼顾数据网格的逻辑结构、并能为其他数据网格复制服务提供副本的 QoS 信息，因此具有较好的实用性。下一步的工作，我们将在实际的网格环境中对该机制进行进一步的实验和分析，并寻找提高 P2P 覆盖网络性能和可靠性的途径。

参考文献

1 Baru C, Moore R, Rajasekar A, et al. The SDSC Storage Resource Broker[C]. CASCON'98, Toronto, Canada, 1998.

2 Ripeanu M, Foster I. A Decentralized, Adaptive, Replica Location Mechanism[C]. Proc. of the 11th IEEE International Symposium on High Performance Distributed Computing, Edinburgh, Scotland, 2002.

3 Li Dongsheng, Xiao Nong, Lu Xicheng, et al. Dynamic Self-adaptive Replica Location Method in Data Grids[C]. Proceedings of the IEEE International Conference on Cluster Computing, 2003.

4 Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications[C]. ACM SIGCOMM, 2001.

（上接第 105 页）

算法中局部搜索造成的性能下降，也可以在链路负载平衡方面较原有的 CBT 算法有一个质的提高。但时间复杂度整体来说还较大。如何改进该算法，使其能够适合更一般的情形，将是今后对于 QOS 覆盖组播路由问题研究的方向。

参考文献

1 Diot C, Levine B N, Lyles B, et al. Deployment Issues for the IP Multicast Service and Architecture[J]. IEEE Network, 2000,14(1).

2 EI-Sayed A, Roca V. A Survey of Proposals for an Alternative Group Communication Service[J]. IEEE Network, 2003, 17(1).

3 吴家皋, 杨音颖, 陈益新等. 支持延时约束的覆盖多播路由协议的研究[J]. 通信学报, 2005, 25(18): 13-20.

4 Shi S, Turner J. Multicast Routing and Bandwidth Dimensioning in Overlay Networks[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(8).

5 Zhang B, Jamin S, Zhang L. Host Multicast: A Framework for Delivering Multicast to End Users[C]. Proceedings of the IEEE INFOCOM, New York, 2002.