

基于 Tabu 搜索的 QoS 多播路由快速优化算法*

高茜⁺, 罗军舟

(东南大学 计算机科学与工程系 网络室, 江苏 南京 210096)

A Tabu-Search-Based Fast QoS Multicast Routing Optimal Algorithm

GAO Qian⁺, LUO Jun-Zhou

(Network Laboratory, Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

+ Corresponding author: Phn: +86-25-83795595, Fax: +86-25-83794838, E-mail: gq@seu.edu.cn, <http://www.seu.edu.cn>

Received 2004-02-13; Accepted 2004-07-15

Gao Q, Luo JZ. A tabu-search-based fast QoS multicast routing optimal algorithm. *Journal of Software*, 2004,15(12):1877~1884.

<http://www.jos.org.cn/1000-9825/15/1877.htm>

Abstract: The key problem of QoS multicast routing optimization algorithm is to build QoS constrained multicast tree. This is the most noted problem in computer networks as constrained minimum Steiner tree problem, which has also been shown as NP-complete. A fast multicast routing optimal algorithm with QoS constraints based on Tabu-search is proposed in this paper. It chooses delay and bandwidth as QoS parameters, can improve the search speed, and make a better solution by using the merits of Tabu-search. Simulation results with random network models show that TSQMA is faster, easier for implementation, and more suitable for large group scale.

Key words: multicast routing; QoS; Tabu search

摘要: QoS 多播路由算法的核心问题是建立满足 QoS 约束的多播树,这就是计算机网络中著名的受约束最小 Steiner 树问题,是一个 NP 完全问题。目前已有的启发式算法的时间复杂度大,不能获得最优解。提出了一个基于 Tabu 搜索的 QoS 多播路由选择快速优化算法,它选择延迟与带宽约束为 QoS 参数,利用 Tabu 搜索的集中性与广泛性并存的优点,在提高搜索速度的同时可以更加接近最优解。仿真结果表明:该算法具有快速、易实施等特点,更加适合在组规模比较大的情况下应用。

关键词: 多播路由; QoS; Tabu 搜索

中图法分类号: TP393 **文献标识码:** A

* Supported by the National Natural Science Foundation of China under Grant No.90204009 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1998030402 (国家重点基础研究发展规划(973)); the China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No.20030286014 (高等学校博士学科点专项科研基金课题); the Key Laboratory of Network and Information Security of Jiangsu Province of China under Grant No.BM2003201 (江苏省“网络与信息安全”重点实验室)

作者简介: 高茜(1975—),女,陕西韩城人,博士生,主要研究领域为高性能网络协议和算法;罗军舟(1960—),男,博士,教授,博士生导师,主要研究领域为高性能网络与协议工程,网络安全,网络管理,网格计算。

随着网络技术与多媒体技术的飞速发展,许多分布式实时应用(如视频会议,VoD 点播,远程教学,计算机支持的协同工作,游戏和仿真等)应运而生,它们对目前尽力而为的 Internet 网络提出了新的要求.从这些实时应用和用户的角度出发,希望能得到一定的服务质量(QoS)的保证,比如要求保证带宽要求、满足端到端的延迟、延迟抖动约束等等.从网络运营者的角度出发,由于这类应用传输数据量大,并且具有一点到多点或多点到多点的传输形式,因而他们希望能采用多播的方式传输数据来最大限度地利用有限的网络资源,从而获得更高的经济利益.而提供满足 QoS 需求的多播服务的关键是如何建立满足多个 QoS 约束的最小代价树,这个问题又可以称为受约束最小 Steiner 树问题.

不同的 QoS 度量参数具有不同的性质,按照这些性质,可以将 QoS 度量参数分为以下 3 种类型:累加型、累乘型以及凹型.累加型度量如端到端延迟等,累乘型度量如报文丢失率等,凹型度量如瓶颈带宽等.目前已经证明了基于任意两个或两个以上的累加型的或累乘型的多约束条件下的路由选择问题是 NP 完全问题^[1],只能寻求于启发式方法来求解,而受约束最小 Steiner 树问题也是一个 NP 完全问题,近些年来国内外有不少学者都在从事这方面的研究^[2-7].Kompella VP, Pasquale JC 和 Polyzo GC 等人于 1992 年提出了以他们名字命名的延迟约束最小 Steiner 树算法 KPP^[2],该算法首先在原始网络 N 中计算一个延迟约束的完全闭图 N_1 (完全闭图是一个图,它包含网络中的所有结点,在每对结点之间有一条边,代表这对结点之间的满足延迟约束的最小代价路径),然后通过 Prim 算法,基于 N_1 构造一个最小生成树 T ,最后将 T 中的边用原始网络 N 中的路径代替,如果有循环路径则删除.KPP 算法的一个限制是假定链路延时和延时约束 Δ 都是整数,并且只有当 Δ 是一个固定值时,它才能够求在多项式时间内求解,它的时间复杂度为 $O(\Delta|V|^3)$.Zhu Q 等人提出了一种约束最短组播路由算法 BSMA^[3](bounded shortest multicast algorithm)来解决延迟约束的最小 Steiner 树问题,该算法首先生成一棵最小延迟路径树,然后迭代地使用不在延迟最短路径树中的一条代价更低的“超边”来替换该树中相应的路径,替换的前提是不违背给定的延迟约束,而发现超边的方法是使用 Kth-最短路径算法.BSMA 算法是目前最好的延迟约束多播路由优化算法,它的时间复杂度是 $O(K|V|^3 \log(|V|))$.Goel A 等人以动态规划算法为基础,提出了延迟缩放算法 DSA(delay scaling algorithm)^[4],该算法假设延迟和延迟约束条件都为整数,通过降低延迟需求来降低多播树的代价,提高运算速度,它需要考虑延迟松弛所带来的不精确性,最坏情况下,该算法的时间复杂度为 $O\left(\frac{(|E|+|V| \log |V|)|V|}{\varepsilon}\right)$, ε 为缩放因子.Zhao J 等人提出了“破圈回滚”组播路由算法 CRMA^[5](cycle-breaking rollback multicast algorithm),首先寻求生成的组播树代价最小,然后再考虑满足用户通信延迟的要求,该算法比较均衡地考虑了延迟和代价在路由搜索过程中的折衷,它的时间复杂度为 $O(|D||V|^3)$, D 为多播组成员集合.

综上所述,目前存在的启发式算法的最大缺陷是时间复杂度比较大,只考虑了延迟约束,并没有考虑带宽等其他 QoS 约束条件.Tabu 搜索(TS,也称作禁忌搜索)是 Glover 首先提出的用于大型组合优化问题的随机全局优化方法^[8,9].它可以模拟人脑的记忆和推理功能,具有搜索的集中性与广泛性并存的优点,在很大程度上避免搜索陷入局部最优解,从而使最终解更接近全局最优解.因此,我们提出了一种基于 Tabu 搜索的 QoS 约束的多播路由选择优化算法(Tabu search based QoS multicast algorithm,简称 TSQMA),并选择延迟与带宽作为 QoS 参数的代表对该算法进行了详细的描述和分析.对 BSMA 和 KPP 算法的仿真比较结果表明,TSQMA 在提高多播路由的搜索速度的基础上取得了更优的解.

1 QoS 约束的多播路由选择优化问题

在分析 QoS 路由问题时,我们可以把通信网络用一个加权图 $N(V,E)$ 来表示^[10],其中 V 是 N 中节点的集合, E 是链路集合. $|V|$ 和 $|E|$ 分别代表节点和链路的数目.根据不同应用的需要,每条链路 $e \in E$ 会与不同的参数相关联,比如:链路带宽函数 $b(e):e \rightarrow R^+$,链路延迟函数 $d(e):e \rightarrow R^+$,链路代价函数 $c(e):e \rightarrow R^+$,这些参数统称为链路状态;每个节点 $v \in V$ 也可以与一些参数相关联,如节点分组丢失率 $Lr(v):v \rightarrow R^+$,端口的自由缓存 $Fb(v):v \rightarrow N^+$,这些参数统称为节点状态.

设 s 为多播组的源节点, $D \subseteq V - \{s\}$ 为多播组 G 的成员集合,多播数据流由源节点 s 经过多播树 $T=(V_T,E_T)$ 传往所有的目的节点,其中, V_T 为多播树上的节点集合, $V_T \subseteq V$, E_T 为多播树上的链路集合, $E_T \subseteq E$.

定义 1. QoS 约束的最优多播树(optimal multicast with QoS constraints,简称 OMQC).给定 s 和 D ,寻找多播树 $T=(V_T,E_T)$,使得 $Cost(T)=\sum_{e \in E_T} c(e)$ 取最小值,并且满足各项 QoS 参数的约束条件:

- 1) $q_i(a) \geq QoSBound_i, \forall a \in E_T \cup V_T$;
- 2) $\sum_{e \in Path(s,m)} w_i(e) \leq QoSBound_i, \forall m \in V_T$.

其中, q_i 为凸型 QoS 参数, w_i 为累加型 QoS 参数, $QoSBound_i$ 为相应的 QoS 约束值, $Cost(T)$ 为多播树 T 的总代价, $Path(s,m)$ 为源节点 s 到成员节点 m 之间的路径.累乘型 QoS 参数可以通过取对数转变成累加型参数来处理.

在本文中,我们以带宽和延迟为 QoS 参数代表来描述 QoS 约束的多播路由选择优化问题.其他凸型参数的描述与处理过程与带宽类似,其他累加型参数的描述和处理过程与延迟类似.

定义 2. 带宽和延迟约束的最优多播树(optimal multicast with bandwidth and delay constraint,简称 OMBDC).给定 s 和 D ,寻找多播树 $T=(V_T,E_T)$,使得 $Cost(T)=\sum_{e \in E_T} c(e)$ 取最小值,并且满足以下几个条件:

- 1) $b(e) \geq BWBound, \forall e \in E_T$;
- 2) $\sum_{e \in Path(s,m)} d(e) \leq DelayBound, \forall m \in V_T$.

其中, $BWBound$ 为带宽约束, $DelayBound$ 为端到端延迟约束.

2 基于 Tabu 搜索的多播路由算法

随着网络规模的增大和组成员的增加,OMBDC 问题的解空间也变得越来越大,找到一个满足条件的最优树通常是不可能的,一般只需找出近似最优解.Tabu 搜索自提出以来得到了大量的应用,在解决包括组合优化问题在内的许多问题中都取得了更好的结果^[11~13].我们将它用于解决 OMBDC 问题,也取得了很好的效果.

2.1 Tabu搜索的原理

Tabu 搜索的核心是尽量模拟人脑的记忆和推理功能,引入禁止搜索表(tabu list)的概念,通过这张表来指导具体的搜索过程.满足禁止条件(tabu condition)的解或者解的某项属性被放入禁止搜索表中,这样便可以保证搜索过程不会陷入死循环,同时,禁止搜索表中满足期望条件(aspiration level condition)的解被忽略其禁止状态,这样便增加了搜索的广泛性,可以更加快速地逼近全局最优解.Tabu 搜索算法的框架如下所示:

步骤 1. 初始化:随机生成一个初始解 i ,令最优解 $i^*=i$,迭代次数 $k=0$,禁止搜索表 $T(i,1)=\phi$.

步骤 2. 生成本次迭代的候选解集合: $k=k+1$,从解 i 的邻域中找出一定数量的解作为候选解集合 $N(i,k)$,并且满足 $N(i,k) \cap T(i,k)=\phi$.

步骤 3. 选择本次迭代的最优解:

- 1) 若 $N(i,k)=\phi$,则转 2);否则,从 $N(i,k)$ 中选出最优解 j_1 .
- 2) 若 $T(i,k)=\phi$,则转 3);否则,从 $T(i,k)$ 中选出满足期望条件的最优解 j_2 .
- 3) 取 j 为 j_1 和 j_2 之间最优的那个解,令 $i=j$;若 i 优于 i^* ,则 $i^*=i$.

步骤 4. 更新禁止搜索表 $T(i,k+1)$.

步骤 5. 判断终止条件:若迭代次数到或者终止条件满足,则输出 i^* ,否则,转向步骤 2.

在具体应用中,算法中邻域的选择、禁止条件以及禁止搜索表长度的设置、期望条件和终止条件的选择并没有一个确定的模式,可以根据实际问题灵活地实现.

2.2 TSQMA的基本思想

一棵多播树主要由源节点(s)、目的节点集(D)和树上起连接作用的中继节点(R)组成,这 3 类节点统称为树上节点.可以看出,只要中继节点确定了,这棵多播树便唯一确定.因此,寻找最优多播树的过程也就是确定该树

中的中继节点集合的过程。TSQMA 的核心思想就是通过 Tabu 搜索寻找最优多播树的中继节点集合。初始解的中继节点集为最小延迟路径树的中继节点集,它的邻域解集为在当前解的中继集合中添加或删除一个中继节点所生成的解集。在 Tabu 搜索的每次迭代过程中都从邻域解集内选择一个最优解作为当前解,并且把本次操作的类型(添加或删除)以及操作的节点记录在禁止搜索表中,为了控制循环操作,在禁止搜索表中的节点被禁止再次进行操作,除非再次操作该节点会引导当前解走向最优解,这就是本算法中期望条件之一。禁止搜索表和期望条件是 Tabu 搜索中控制解的走向的两个主要方法,也是保证搜索能够快速收敛到最优解的主要手段。

2.2.1 初始化

算法假设整个网络的拓扑信息以及状态信息在一定的时间间隔内保持不变,并且可以通过扩展已有的路由协议,如 OSPF(open shortest path first)协议^[14]来获得 QoS 状态信息。算法首先删除网络 $N(V,E)$ 中不满足带宽约束条件的链路,生成新图 $N_1(V_1,E_1)$,其中 $V_1 \subseteq V, E_1 \subseteq E$ 。然后在 N_1 上使用 Dijkstra 最短路径算法,以延迟作为参数计算最小延迟路径树 $T(V_T, E_T)$, $V_T = S \cup D \cup R_T$, R_T 为 T 的中继节点集合。 T 为初始解,令最优解 $T^* = T$ 。

2.2.2 邻域解集的生成和候选节点的选择

通过在 R_T 上进行“添加”或“删除”中继节点的操作来生成邻域解集。同时,为了减少邻域解集的大小以及计算量,我们对邻域解集做了一定的限制,使得邻域解集中的解尽量为有效解,并且使优于当前解的可能性增大。

1) “添加”操作的邻域解集

设当前解 T 上最大代价的链路为 e , 令 $\text{MaxCost} = c(e)$, 那么“添加”操作的邻域解集 Ne_{add} 中的每个节点 $m(m \in (V_1 - V_T))$ 必须满足以下条件:

首先,节点 m 不满足禁止条件;第 2,节点 m 至少与当前解中的两个节点有链路相连,这样便可保证该节点加入后到当前树中不会成为叶子节点;第 3,节点 m 连接到树上的链路必须小于 MaxCost ;第 4,节点 m 添加到当前树中有可能造成环路,增加链路的代价值必须小于解除环路要删除的链路的代价。设 m 连接到 T 的两条链路分别为 (a,m) 和 (m,b) , $\text{CostPath}(a,b)$ 为 T 中从 a 到 b 的路径的代价,则要求 $\text{CostPath}(a,b) > c(a,m) + c(m,b)$;最后,添加节点 m 后, m 子树的延迟必须满足约束条件。这就需要计算添加 m 到 T 之后生成的解 T_1 ,计算方法为:取 $V_2 = V_T + m, E_2 = \{e | e \in E_1, \text{并且 } e \text{ 的两个端点都属于 } V_2\}$, 构造一个图 $N_2(V_2, E_2)$ 。在图 N_2 中使用 Prim 最小生成树算法生成一棵包含 V_2 中所有节点的代价最小的树,删除不是组员的叶子节点,检查源节点到所有组员的延迟,若不满足延迟约束,则以 T 中源节点到该节点的路径代替不满足延迟约束的路径,并消除环路。若 T_1 与当前解 T 不相同,则将 m 和 T_1 添加到“添加”操作的邻域解集 Ne_{add} 中。

2) “删除”操作的邻域解集

“删除”操作的邻域解集 Ne_{del} 中的每个节点 $m(m \in R_T)$ 必须满足以下条件:

首先,节点 m 不满足禁止条件;其次,删除节点 m 后, m 的子树的延迟必须满足约束条件。同样需要计算从当前解 T 中删除 m 之后生成的解 T_1 ,计算方法与添加节点的算法类似。若 T_1 与当前解 T 不相同,则将 m 和 T_1 添加到“删除”操作的邻域解集 Ne_{del} 中。

3) 选择候选节点

当前解 T 的邻域解集 $Ne = Ne_{add} \cup Ne_{del}$, 分别计算当前解与 Ne 中各解的代价差值,取 Ne 中代价差值最大的解为候选解,与它相对应的节点称为候选节点。

2.2.3 禁止搜索表与期望条件

每次操作后,把当前操作(添加或删除)的类型标识和节点记录在禁止搜索表中。禁止搜索表采用队列结构,用来记录最近操作的历史记录。每次迭代所进行的操作都被记录在队首,队列溢出时排在队尾的记录被删除。这样的结构使得队列中永远是最新的操作记录。队列长度用来限制历史记录的长度,合理地设置队列的长度可以避免搜索陷入死循环,同时又可以使一些优异的解尽快地被搜索,通过仿真实验,我们设置队列长度为 10。

如果禁止搜索表中所记录的某个操作会搜索到一个从未访问过的更优的解,那么这个操作就不应该被禁止。设置期望条件可以使禁止搜索表中的某些操作的禁止状态被忽略。在这里,我们设置的期望条件是:第 1,位于队首的节点不能被再次添加或删除,这个条件保证了刚刚添加的节点不能立即被删除,或者刚刚删除的节点不能在下一迭代中被添加;第 2,操作后所得的解优于当前最优解。

2.2.4 结束条件

在本算法中,我们设置了最大迭代次数 $\max_iteration$ 和最大未改进次数 $\max_unchanged$ 来控制 Tabu 搜索的结束.如果算法已经进行了 $\max_iteration$ 次迭代,则结束搜索;如果连续迭代搜索 $\max_unchanged$ 次所得到的多播树的代价都没有减少,则结束搜索.

2.3 算法伪代码

定义数据结构 $Neighbor$ 为三元组 $(m, type, Tree)$, 其中 $Neighbor.m \in V$ 为添加或删除操作的对象, $Neighbor.type$ 为操作类型, $Neighbor.Tree$ 中记录邻域解的生成树信息, 它也是一个三元组 $(T, R, Cost)$. $Tree.T$ 记录树的结构, $Tree.R$ 为树的中继节点集合. $Tree.Cost$ 记录树的总代价. 函数 $ComputeTree$ 用来计算添加或删除操作后的满足延迟约束条件的生成树. 近期操作过的节点被记录在禁止搜索表 $TabuTable$ 中. T 和 T^* 都是 $Tree$ 类型的变量, 前者记录当前解, 后者记录最优解. 下面为 TSQMA 的伪代码.

TSQMA Algorithm($N, s, D, BWBound, DelayBound$)

1. Delete the links of N which is unsatisfied with the $BWBound$, and get a new Graph N_1 ;
2. Compute the shortest delay path tree T_1 use Dijkstra Algorithm;
3. $TabuTable = \emptyset$; $Count_Not_Changed = 0$; $Count_Iteration = 0$; $T = T_1$; $T^* = T_1$;
4. While $Count_Iteration < Max_Iteration$ and $Count_Not_Changed < Max_Not_Changed$ Do
 5. For all $m \notin TabuTable$ and $m \in V - (D \cup T.R \cup \{s\})$ Do //对不属于 TabuTable 中的非树上节点进行搜索
 6. //检查 m 是否满足添加操作条件
 7. If $c(a, m) < MaxCost$ and $c(m, b) < MaxCost$ and $c(a, m) + c(m, b) - CostPath(a, b) < 0$ Then
 8. $T_1 = ComputeTree(m, N_1, T)$; //计算添加节点 m 后的满足约束条件的最小生成树
 9. If $T_1 < T$ Then $Ne_{add} = Ne_{add} \cup \{m, "add", T_1\}$; End If //添加到“添加”操作邻域解集 Ne_{add} 中
 10. End If
 11. End For
 12. For all $m \in TabuTable$ and $m \in V - (D \cup T.R \cup \{s\})$ Do //搜索期望条件的 TabuTable 中的非树上节点
 13. $T_1 = ComputeTree(m, N_1, T)$; //计算添加节点 m 后的满足约束条件的最小生成树
 14. If $T_1.Cost < T^*.Cost$ Then $Ne_{add} = Ne_{add} \cup \{m, "add", T_1\}$; End If //添加到“删除”操作邻域解集 Ne_{add} 中
 15. End For
 16. For all $m \in T.R$ Do //搜索当前树中的中继节点
 17. $T_1 = ComputeTree(m, N_1, T)$; //计算删除节点 m 后的满足约束条件的最小生成树
 18. If $(T_1 < T$ and $m \notin TabuTable)$ or $(T_1.Cost < T^*.Cost$ and $m \in TabuTable)$ Then
 19. $Ne_{del} = Ne_{del} \cup \{m, "del", T_1\}$; //将满足删除操作条件的解加入邻域解集 Ne_{del} 中
 20. End If
 21. End For
 22. $\max = -\infty$;
 23. For all $neighbor \in Ne_{add} \cup Ne_{del}$ Do //在邻域集中寻找最佳邻域解 $\max_neighbor$
 24. If $neighbor.Tree.cost - T^*.cost > \max$ Then
 25. $\max = neighbor.Tree.cost - T^*.cost$
 26. $\max_neighbor = neighbor$;
 27. End If
 28. End For
 29. $TabuTable = TabuTable \cup (\max_neighbor.m, \max_neighbor.type)$; //更新 TabuTable
 30. If $\max_neighbor.Tree.cost < T^*.cost$ Then //如果当前解 T 优于最优解 T^* , 更新最优解
 31. $T^* = \max_neighbor.Tree$;

```

31.      Count_Not_Changed = 0;           //复位计数器 Count_Not_Changed
32.      Else
33.      Count_Not_Changed ++;           //更新计数器 Count_Not_Changed
34.      End If
35.      Count_Iternation ++; T = max_neighbor.Tree; //更新计数器 Count_Iternation 和当前解 T
36.      End While

```

3 算法分析

3.1 正确性分析

算法的初始化部分首先删除网络中不满足带宽约束的链路,并在精简后的网络中以延迟作为参数计算最小延迟路径树,若该树不能满足给定的延迟约束条件,显然不存在能满足带宽和延迟约束条件的树.否则,算法将通过 Tabu 搜索最优的中继节点集合来缩小多播树的代价.因为 Tabu 搜索的基础是 Prim 最小生成树算法,每一轮搜索都要进行延迟约束检查,对不满足延迟约束的路径进行路径替换,并且消除了路径替换引起的环路.因此,最后生成的树一定是满足带宽和延迟约束条件的次优多播树.

3.2 时间复杂性

从算法描述看,TSQMA 的核心是计算邻域解集,而计算邻域解集的关键是 Prim 算法,在每次迭代中,需要运行 n 次 Prim 算法, $n \leq (|V|-|D|-1)$. 当组成员比较少的时候,根据“添加”操作的第 4 条限制条件, $n \ll |V|$. 随着组成员的增多, $|V|-|D|-1 \ll |V|$. Prim 算法的运行时间为 $O(|E|+|V|\log|V|)$, 因此,在该算法中一次迭代运算的时间复杂度为 $O(|E|+|V|\log|V|)$, 则整个算法的时间复杂度为 $O(k(|E|+|V|\log|V|))$, 其中 k 为迭代次数.

4 仿真结果

我们在 Intel PIV 2G 微机 Windows XP 操作系统环境下,使用 C++ Builder 编程来设计仿真软件.采用由于 Waxman B^[15] 随机图模型生成仿真实验中的网络拓扑.该模型的数学表达式为

$$p(i, j) = \alpha \times \exp \left[-\frac{\text{distance}(i, j)}{L\beta} \right],$$

$p(i, j)$ 表示节点 i, j 间链路存在概率; $\text{distance}(i, j)$ 为节点 i, j 间的笛卡尔平面距离; L 为 $\text{distance}(i, j)$ 最大值, 参数 α 用来控制链路的数量, 大的 α 将增加链路的数量; 参数 β 用来控制短链路的数量, 小的 β 将增大短链路的数量. 在本文中, 取网络大小为 100 个节点, $\alpha=0.2, \beta=0.4$, 节点的平均度数为 4. 链路可用带宽分布在 [0.56Mbps, 10Mbps] 区间, 链路延迟由 $\text{distance}(i, j)$ 决定. 源节点及目的节点是随机选择的, 多播业务所需要的带宽在 1Mbps~8Mbps 之间变化, 多播业务所需的延迟上限在 10ms~200ms 之间变化.

在上述环境下,我们对本文提出的 TSQMA 和 BSMA, KPP 算法进行了比较. BSMA 是目前提出的延时约束多播路由算法中性能最好的算法, 由于 BSMA 和 KPP 都没有考虑带宽约束, 有可能导致搜索出的多播树中某些链路不满足带宽约束, 从而使寻路的成功率降低, 图 1 描述了多播组成员为 20, 延迟约束为 100ms 时带宽约束与寻路成功率之间的关系, 图中的每个数据都是从 100 次随机实验中求平均值得到的. 可以看到, TSQMA 具有最高的寻路成功率, 这是因为它在选路时同时考虑了带宽和延迟约束. 而随着多播业务所需带宽的增加, BSMA 和 KPP 算法的寻路成功率下降趋势很快. 考虑到对比的公平性, 在下面的实验中, 在运行 BSMA 和 KPP 算法之前, 我们对网络也进行了类似 TSQMA 的带宽预处理.

图 2 比较了延迟约束变化时各种算法生成的多播树代价的情况, 多播组大小固定为 20 个成员, 延迟约束由 10ms~200ms 变化. 从图中我们可以看到, 在延迟约束比较紧的情况下, TSQMA 所得的多播树代价与 BSMA 的代价相差无几, 随着延迟约束的逐渐放松, TSQMA 要略优于 BSMA. 而 TSQMA 和 BSMA 产生的多播树的代价明显小于 KPP 算法.

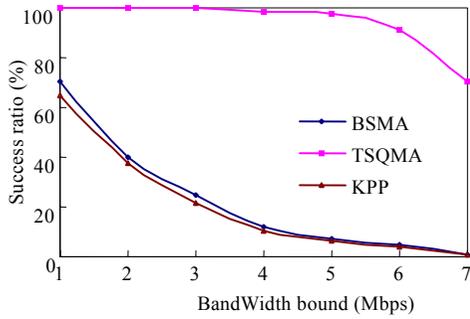


Fig.1 Routing success ratio under different bandwidth bound

图 1 带宽约束变化时寻路的成功率

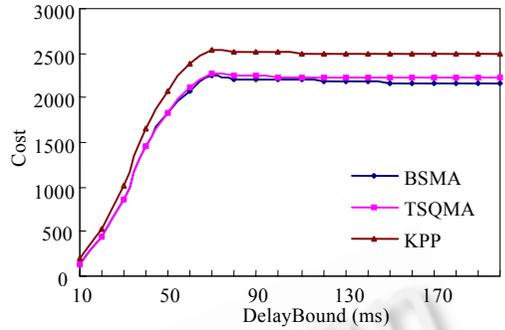


Fig.2 Multicast tree cost under different delay bound

图 2 延迟约束变化时多播树的代价

图 3 描述了多播组大小从 10 个成员~90 个成员变化时各算法的多播树代价的情况,延迟约束为 100ms.同样可以看到,TSQMA 所得的多播树的代价要略优于 BSMA,这是因为 BSMA 在进行超边替换时有着局部性问题,而 TSQMA 所采取的 Tabu 搜索在每一次迭代中不一定取优于上次迭代的解,从而跳出了局部最优解,因而更有可能寻找到全局最优解.

图 4 比较了 TSQMA 和 BSMA 在网络大小固定、多播组大小变化时生成多播树的运行时间.从图中可以看出,TSQMA 的运行时间远低于 BSMA,并且,随着多播组大小的增加,BSMA 的运行时间的增长趋势也远远快于 TSQMA.

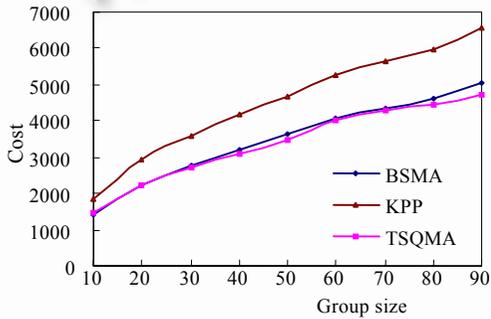


Fig.3 Multicast tree cost under different group size

图 3 多播组大小变化时树的代价

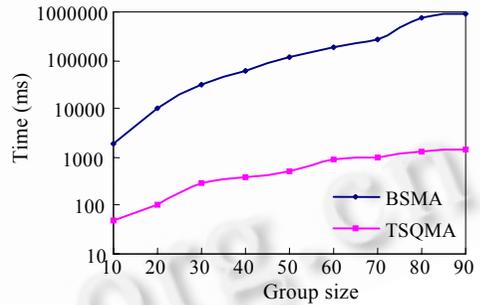


Fig.4 Algorithm run time under different group size

图 4 多播组大小变化时算法的运行时间

图 5 对多播组大小从 10 个成员变化到 90 个成员时 TSQMA 和 BSMA 的搜索的迭代次数进行了比较.从图中可以明显地看出,随着组成员大小的增加,TSQMA 的搜索迭代次数的增长速度远小于 BSMA,而且当组大小增加到一定程度时,TSQMA 的搜索迭代次数反而会下降,这是因为 TSQMA 的搜索空间会随着组成员的增加而减小.图 6 给出了网络大小为 100 个节点,多播组大小为 50 个成员时,不同延迟约束条件下 TSQMA 的收敛情况.从图中可以看出,算法收敛很快.

5 结 论

本文提出了一种基于 Tabu 搜索的 QoS 约束的多播路由优化算法 TSQMA,并以带宽和延迟约束的多播应用为例对算法进行了详细地描述和分析.该算法利用 Tabu 搜索的特点,有效地引导搜索快速逼近最优解.仿真实验结果表明:TSQMA 与同类算法相比更接近最优解,同时它克服了过于复杂而难以实施的缺点,更加适合在组规模比较大的情况下应用.在下一步的工作中,我们将针对多播组成员的动态性和异构性问题对 TSQMA 进行进一步的改善和实验分析,使之具有更强的实用性.

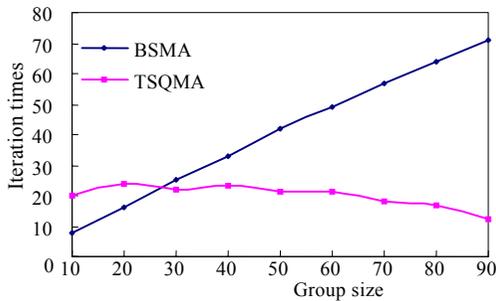


Fig.5 Searching iteration times under different group size

图5 多播组大小变化时搜索的迭代次数

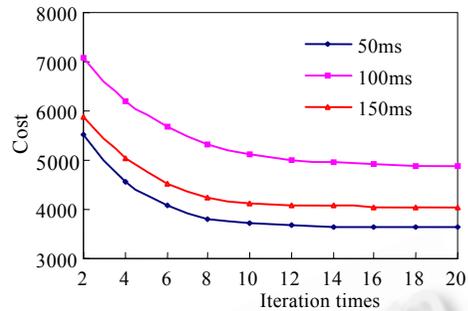


Fig.6 Multicast tree cost under different delay bound and iteration times

图6 不同迭代次数和延迟约束下的多播树代价

References:

- [1] Wang Z, Crowcroft J. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 1996, 14(7): 1228~1234.
- [2] Kompella VP, Pasquale JC, Polyzos GC. Multicasting for multimedia applications. In: *IEEE Computer Society, ed. Proc. of the IEEE INFOCOM'92. Vol 3, Florence: IEEE Press, 1992. 2078~2085.*
- [3] Zhu Q, Parsa M, Garcia-Luna-Aceves J. A source-based algorithm for delay-constrained minimum-cost multicasting. In: *IEEE Computer Society, ed. Proc. of the IEEE INFOCOM'95. Vol 1, IEEE Press, 1995. 377~385.*
- [4] Goel A, Ramakrishnan KG, Kataria D, Logothetis D. Efficient computation of delay-sensitive routes from one source to all destinations. In: *IEEE Computer Society, ed. Proc. of the IEEE INFOCOM 2001. Vol 2, AK: IEEE Press, 2001. 854~858.*
- [5] Zhao J, Hassanein H, Wu J, Luo J. CRMA: A cycle-breaking multicast routing algorithm for supporting QoS over the internet. In: *IEEE Computer Society, ed. Proc. of the 36th Annual Hawaii Int'l Conf. on System Sciences. IEEE Press, 2003. 285~293.*
- [6] Kuipers F, Mieghem PV. MAMCRA: A constrained-based multicast routing algorithm. *Computer Communications*, 2002, 25(8): 802~811.
- [7] Melody MW, Nguyen B. QoS-Guaranteed one-to-many and many-to-many multicast routing. *Computer Communications*, 2003, 26(7): 652~669.
- [8] Glover F. Tabu search-Part I. *ORSA Journal on Computing*, 1989, 1(3): 190~206.
- [9] Glover F. Tabu search-Part II. *ORSA Journal on Computing*, 1990, 2(1): 4~32.
- [10] Feng J, Ma X, Gu G. Research of network model adapt to QoS routing mechanism. *Chinese Journal of Computers*, 2000, 23(8): 799~805 (in Chinese with English abstract).
- [11] Yan S, Ali M, Jitender D. Route optimization of multicast sessions in sparse light-splitting optical networks. In: *IEEE Computer Society, ed. GLOBECOM 2001. Vol 4, IEEE Press, 2001. 2134~2138.*
- [12] Bastos MP, Ribeiro CC. Reactive tabu search with path-relinking for the Steiner problem in graphs. In: *Ribeiro CC, Hansen P, eds. Essays and Surveys in Metaheuristics. Boston: Kluwer Academic Publishers, 2002. 39~58.*
- [13] Yang W. A tabu-search based algorithm for the multicast-streams distribution problem. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 2002, 39(6): 729~747.
- [14] Apostolopoulos G, Williams D, Kamat S, Guerin R, Orda A, Przygienda T. QoS routing mechanisms and OSPF extensions. RFC No. 2676, Internet Engineering Task Force, 1999.
- [15] Waxman BM. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 1988, 6(9): 1617~1622.

附中文参考文献:

- [10] 冯径, 马小骏, 顾冠群. 适应 QoS 路由机制的网络模型研究. *计算机学报*, 2000, 23(8): 799~805.