

带宽预留的成组多播快速路由算法

胡光岷¹, 李乐民¹, 安红岩²

(1. 电子科技大学宽带光纤传输与通信系统技术国家重点实验室, 四川成都 610054; 2. 成都理工学院应用数学系, 四川成都 610059)

摘 要: 本文在现有成组多播路由算法的基础上, 通过最大限度地使用原有的计算结果、以存储空间换取计算时间两种策略, 获得了计算效率很高的带宽预留的成组多播快速路由算法 FGMRA (Fast Group Multicast Routing Algorithm). 使用 FGMRA 得到的多播生成树总费用与 C. P. Low 和 N. Wang 算法 (一种总代价和路由分配失败概率较低的成组多播路由算法) 基本相同, 计算效率大为提高. 随机网络模型的仿真结果表明: FGMRA 算法快速、稳定, 是一种值得推广使用的高效算法.

关键词: 成组多播; 路由; 带宽; 最小代价

中图分类号: TP393.07

文献标识码: A

文章编号: 0372-2112 (2003) 04-0569-04

A Fast Routing Algorithm for Group Multicast with Bandwidth Reservation

HU Guang-min¹, LI Le-min¹, AN Hong-yan²

(1. National Key Lab of Optical Fiber Communications, UESTC, Chengdu, Sichuan 610054, China;

2. Department of Applied Mathematics, CDUT, Chengdu, Sichuan 610059, China)

Abstract: Based on existing routing Algorithm, we propose a fast algorithm called FGMRA (Fast Group Multicast Routing algorithm) for group multicast by using two strategies. One is to make the best of calculation in existence. Another is to enlarge the storage spaces to get high speed. Compared with algorithm proposed by C. P. Low and N. Wang, the total cost of multicast trees created by two algorithms is almost equal, but our algorithm is faster. The simulation result shows that FGMRA is fast, stable and efficient, and can be used widely.

Key words: group multicast tree; routing algorithm; bandwidth; minimum cost

1 引言

IP 多播有效地解决了单点发送多点接收、多点发送多点接收的问题, 在多媒体会议、数据分发、分布式并行处理和分布式交互仿真等方面获得了广泛的应用. 成组多播是一种特殊的多播技术, 它的每一个多播组成员都可能向多播组的其它成员发送多播信息, 每一个多播组成员既是信息的发送者也是接收者. 成组多播的典型应用是远程多媒体会议, 每一个参加者均能并发地发送或接收信息.

成组多播的一种实现方案是由每一个多播组成员各生成一棵最小代价生成树. 尽管每一棵最小代价生成树生成时, 要考虑带宽限制, 但不考虑整体带宽限制, 使 QoS 得不到保障. 另一种方案是采用 CBT (Core Based Tree) 技术^[1], 对整个多播组生成一棵以多播核为根, 包括所有多播组成员的多播最小生成树. 该方案在多播核附近的链路上需要的带宽过大, 容易引起严重的拥塞, 且网络延迟较大. 随着成组多播应用的扩展, 越来越多的用户要求 IP 多播技术能为成组多播应用提供完善的 QoS 保障, 也就是要求 IP 多播路由能保证将 IP 分组传

送到多播组的每一个成员. 对此一个简单有效的方法是在路由选择时, 为成组多播应用预留足够的带宽. Jia、Wang 基于 Kou、Markowsky 和 Berman (KMB) 的多播最小代价生成树算法^[2], 提出了一种带宽预留的成组多播路由算法 (简称 GKMB 算法)^{[3][6]}, C. P. Low、N. Wang 基于 Takahashi 和 Matsuyama 多播最小代价生成树算法^[4], 对 GKMB 算法进行了改善 (简称 GIM 算法)^[5], 进一步降低了多播树的总代价和路由分配失败概率 (网络不能产生满足带宽要求的成组多播路由称为路由分配失败). 这两种算法除所使用的多播最小代价生成树算法不同外, 其余部分大致相同.

GKMB 算法和 GIM 算法在一定程度上满足了成组多播路由选择的需要, 但由于在实现过程中, 当网络状态变化时 (删除饱和边后), 这两个算法需要使用静态算法完全重新计算最短路由和多播最小代价生成树. 且所使用的多播最小代价生成树算法的计算效率较低, 因而成组多播路由选择所需的计算时间较长. 针对以上情况, 本文提出一种新的带宽预留的成组多播路由算法 FGMRA. 新算法以适当地增加存储开销为代价, 根据成组多播路由计算的特点, 利用了一些适应动态

收稿日期: 2001-11-29; 修回日期: 2002-05-16

基金项目: 国家自然科学基金 (No: 60002004)

变化的策略和快速的静态多播最小生成树算法 FMPH^[7], 使计算复杂性相对于 GKMB 算法和 GIM 算法大为降低. 且新算法产生的多播生成树的总费用与 GIM 算法基本相同.

2 定义和术语

首先将一个通信网络表示为一个带权无向图 $G = (V, E, C, B)$, 其中 V 代表网络中节点的集合; E 是一组边 (edge) 的集合; C 是各边对应的费用 (cost) 的集合, 每条边用对应的两个节点 i, j 来表示 (i, j) , 一对节点对应的两条边 (i, j) 和 (j, i) 费用相等, 即这两条边对称; B 是各边对应带宽的集合, 一般说来带宽是不对称的, 即 $b_{ij} \neq b_{ji}$.

给定一个网络图 $G = (V, E, C, B)$, 若 $D = (v_1, v_2, \dots, v_m)$ 是由所有多播组成员构成的集合 (称为多播组), $D \subseteq V$, D 中的每一个节点可称为多播节点或端节点, 其带宽需求 q_i 由用户确定. 成组多播路由问题就是要寻找有向的路由树集合 $\{T_1, T_2, \dots, T_m\}$, 每一个多播组成员对应一棵生成树, 满足如下条件:

$$\begin{aligned} & \min_{i, j \in V} \sum_{l=1}^m c_{i,j} X_{i,j}^l \quad (1) \\ & q_i X_{i,j}^l \leq b_{ij}, X_{i,j}^l = \begin{cases} 1 & (i, j) \in E \\ 0 & (i, j) \notin E \end{cases} \quad (2) \end{aligned}$$

$T_i = (V_i, E_i)$ 表示一棵以 $S_i \in D$ 为根节点的生成树, 且 $V_i \subseteq V, D \subseteq V_i$. 通常 V_i 中包含一些集合 $V-D$ 中的节点 (Steiner 节点), 但生成树中所有的叶节点必定是多播组成员. 式 (1) 是 GMRA 问题的目标函数, 保证了所生成的多播树集合费用最小. 式 (2) 为约束条件, 保证分配的带宽不超过实际可利用的带宽.

定义 1 饱和边: 网络中某一条边实际可利用的带宽与已分配的带宽之差, 小于用户所需要的带宽, 则称这条边为饱和边.

定义 2 节点到树的最短路径: 节点到树 $T_i = (V_i, E_i)$ 的各节点最短路径中费用最小的那条, 称为节点到树的最短路径. 该路径用 $PS(u, v)$ 表示 $(u \in V - V_i, v \in V_i)$, 其中节点 v 被称为节点 u 的最小接应节点. 节点到树的最短路径也称为树到节点的最短路径, 节点到树最短路径的费用称为节点到树的最小距离.

定义 3 路径节点: 节点 $u (u \in V - V_i)$ 到生成树的最短路径上, 节点与最小接应节点之间的 Steiner 节点称为该节点相对于树 $T_i = (V_i, E_i)$ 的路径节点.

3 带宽预留的成组多播路由算法

3.1 FGMRA 算法的基本思路

FGMRA 算法对 GIM 算法的改善主要基于以下两条思路, 一是尽量使用原有的计算结果, 避免重复计算; 二是以较小的存储空间为代价, 换取较高的计算效率. 具体的改善措施包括以下几个方面:

- (1) 用快速的多播最小生成树算法 FMPH 代替 MPH 算法.
- (2) 当某些边被标记为饱和边时, 采用动态算法计算网络

节点间新的最短路径和多播最小生成树.

(3) 对于成组多播路由问题来说, 一次需要计算多棵生成树 (其数量大于或等于多播节点的个数), 后面的计算可以使用前面的计算结果来提高计算效率. 当某些边被标记为饱和边后, 这样的饱和边组合在以后有可能还会遇到, 我们将去掉这些饱和边生成的最短路径称为该饱和边组合对应的最短路径 (简称饱和边最短路径, 相应地将保留所有边得到的最短路径称为原始最短路径) 保留在存储器中, 供将来使用. 为避免这样的最短路径占用过多的存储空间, 可根据存储空间的大小, 确定可保留的饱和边最短路径的最大个数 n_{max} . 当所需存储的饱和边最短路径大于 n_{max} 时, 根据最近最久未使用原则, 从已保留的饱和边最短路径中选择一个, 将它淘汰, 其存储空间被新的饱和边最短路径占用.

(4) 某些边被标记为饱和边时, 先在存储器中寻找该饱和边组合对应的饱和边最短路径. 若找到, 则直接使用. 若未找到, 则在所有饱和边最短路径和原始最短路径中, 寻找与其饱和边组合最为相似的饱和边最短路径 (一个饱和边组合增加或减少最少的饱和边就与另一个饱和边组合相同, 则称这两个饱和边组合最为相似), 在其最短路径基础上采用动态算法计算网络节点间新的最短路径.

假定每个多播组成员要正确传送信息所需的带宽相同 (不失一般性, 可设所需带宽为 1). 根据以上讨论, 对已知无向图 G , FGMRA 算法的基本思路如下:

- (1) G 是非连通的, 或存在一个多播组成员节点其输入带宽小于 $m - 1$ (m 为成员节点个数), 算法失败.
- (2) 计算每一个多播节点到 G 中每一个节点的距离.
- (3) 从 D 中选择一个节点 i , 将 i 从 D 中删除, 使用 FMPH 算法, 建立以节点 i 为根的多播最小生成树 T_i . 若 T_i 中不包含饱和边则转 (9).
- (4) 从图 G 中删除饱和边, 得到图 G . 在存储器中寻找该饱和边组合对应的饱和边最短路径. 若找到, 则转 (5). 否则在所有饱和边最短路径和原始最短路径中, 寻找饱和边组合与现有饱和边组合最为相似的饱和边最短路径, 得其最短路径. 使用最短路径计算的动态算法, 重新计算每一个多播组成员节点到 G 中每一个节点的距离.
- (5) 以 T_i 为基础, 使用边减少时的多播最小生成树计算的动态算法, 得到新的以 i 为根的多播最小生成树 T_i . 计算 $O_i = C(T_i) - C(T_i)$. 设饱和边的集合为 E_s , 在最近形成的多播最小生成树中, 寻找一个多播最小生成树集合 W (设其中树的棵数为 k), W 包含 E_s 中所有边.
- (6) 令 $j=1$, 从 W 中选出 T_j . 从图 G 中删除饱和边, 得到图 G . 在存储器中寻找该饱和边组合对应的饱和边最短路径. 若找到, 转 (7). 若找不到则在所有饱和边最短路径和原始最短路径中, 寻找饱和边组合与现有饱和边组合最为相似的饱和边最短路径, 得到其最短路径. 使用最短路径计算的动态算法, 重新计算每一个多播组成员节点到 G 中每一个节点的距离.
- (7) 以 T_j 为基础, 使用边减少时的多播最小生成树计算的动态算法, 得到新的 j 为根的多播最小生成树 T_j . 计算 O_j

= C(T_j) - C(T_j), j = j + 1, 若 j = k, 转(6).

(8) 假如 O_i = O_j, T_i 不变, 对 W 中的每一棵树 T_j 用 T_i 代替. 否则, T_j 用 T_i 代替.

(9) 若 D 非空, 转(3). 否则程序结束.

上述算法运行中还需要使用另外三个算法: (1) 网络节点间最短路径计算的动态算法^[9]; (2) EMPH 算法^[7]; (3) 边减少时, 多播最小生成树计算的动态算法. 前两个算法可参看相关文献, 下面介绍第三个算法.

3.2 边减少时的多播最小生成树动态算法

在已知多播最小生成树 T 的情况下, 若从网络中删除一条边 (u, v), 重新构成多播最小生成树的动态算法是: 如果 (u, v) 不是多播最小生成树中的边, 则多播最小生成树不受影响. 如果 (u, v) 是多播最小生成树中的边, 则删除 (u, v). 不失一般性, 假设此时节点 u 与 T 中的根节点相连接, 将 v 及其的子孙节点中的端节点组成一个集合 M_d, 将删除了集合 M_d 的子树记为 T_i, 删除 T_i 中出度为零的 Steiner 节点. 采用如下算法将 M_d 中的节点加入生成树:

(1) 初始化: 计算 M_d 中每一个节点到子树 T_i 的最短距离, 记为 T_i. DIS. 并令集合 Q 为空.

(2) 在 M_d 不为空的条件下, 选择 M_d 中到 T_i 距离最短的端节点 x, 将 x, x 到生成树最小路径上所有的 Steiner 节点和 x 在原生成树中所有的子孙节点一起加入生成树, 从 M_d 中删除所有已加入生成树的端点. 若 M_d 不为空转下一步, 否则程序结束.

(3) 在生成树中每加入一个节点 u, 考察新加入节点到 M_d 中每一节点 v 的距离, 若该距离小于节点 v 到子树 T_i 的距离, 则将该距离作为节点 v 到子树 T_i 的距离. 转(2).

采用上述动态算法产生的生成树总费用不大于 EMPH 算法产生的生成树, 计算速度快于静态的 EMPH 算法, 且所需存储空间小于 EMPH 算法.

3.3 FGMRA 算法的存储开销

若采用邻接矩阵表示图 G, GIM 算法所需的存储容量大致由三部分组成: (1) MPH 算法所需的存储开销 O(mn + w), 其中 w 为生成树中所包含的节点总数; (2) 计算并存放多播节点到 G 中每一个节点的距离 (即最短距离生成树) 所需的存储开销 O(m²n); (3) 存放 m 棵最小代价生成树所需的存储开销 O(m²n). 采用本文提出的 FGMRA 算法后, 存储开销的增加包括两个方面: (1) EMPH 算法的存储开销由 MPH 算法的 O(mn + w) 增加为 O(mn + w + m); (2) 保留的饱和边最短路径所需的存储开销.

一般情况下, 保留的饱和边最短路径所需的存储开销为 O(km²n), 其中 k 为可保留的饱和边最短路径的最大个数 (即 k = nmax). 为减少存储开销, 可只存放饱和边最短路径相对于原始最短路径改变的部分, 而不保存相同的部分, 使实际存储开销大为减少. 在我们的仿真试验中发现: 当 n = 200, m = 100 时, 存放 21 个饱和边最短路径改变的部分所需的存储开销大致相当于存放一个完整的饱和边最短路径的存储开销.

4 仿真试验及分析

为验证文中提出方法的正确性和有效性, 采用了随机网络模型进行试验, 比较 FGMRA 和 GIM 算法生成最小代价多播生成树的总费用、分配失败的概率和计算时间. 结果发现在所有的随机网络模型中, FGMRA 算法计算时间远小于 GIM 算法; 生成最小代价多播生成树的总费用与 GIM 算法大致相当, 分配失败概率与 GIM 算法完全相同.

为产生有实际网络特征的随机网络图, 采用文献[8]中的方法生成随机网络模型. n 个节点随机地分布在直角坐标系内 (节点的坐标均为整数), 节点之间的欧几里德距离作为其费用. 节点之间的连通性由连通概率 P(u, v) 决定, P(u, v) 的值由 u, v 之间的距离决定.

$$P(u, v) = e^{-d(u, v)/L} \tag{3}$$

这里 d(u, v) 是 u 和 v 之间的距离, L 是任意两节点间的最大距离, 和 是调节网络图特征的参数. 调整和 可以产生不同类型的随机网络图, 使之更接近实际网络. 为讨论方便起见, 采用与文献[5]中相同的参数: (= 0.2, = 0.25), 将两点间的距离定义为边的费用. 每条边的带宽能力由如下式(4)计算得到:

$$B(u, v) = B_m + (-1)^p \cdot q \text{ mod } B_m \tag{4}$$

其中: B_m 为给定的最大带宽的一半; p 和 q 是随机正整数. 式(4)保证了所有边的带宽大于零. 本文中所有的仿真试验均在 1.3GHz 的 Pentium 4 计算机 (内存 256M) 上完成, 可保留的饱和边最短路径的最大个数为 30.

图 1 是网络总节点数固定为 200 时, 使用 FGMRA 和 GIM 两种算法, 计算所有最小代价多播生成树的计算时间随端节点数变化曲线. 其中横坐标为端节点数, 纵坐标为计算时间 (单位毫秒). 从图中可看出, FGMRA 和 GIM 两种算法计算时间随端节点数的增加而增加, 但 FGMRA 算法的增长幅度较小, 且所需的计算时间明显少于 GIM 算法. 若用 C_g 表示用 GIM 算法生成的所有最小代价多播生成树总费用, 用 C_p 表示用 FGMRA 算法生成的所有最小代价多播生成树总费用, 定义总费用的变化率 = (C_p - C_g) / C_g. 图 2 是与图 1 同样情况下, 计算得到的总费用变化量随端节点 (多播节点) 数变化曲线. 其中横坐标为端节点数, 纵坐标为总费用变化率. 从图中可看出, 当节点数固定时, FGMRA 和 GIM 两种算法得到的最小代价多播生成树总费用大致相当 (相对差不超过 2%).

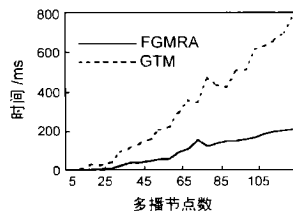


图 1 计算时间随端节点数变化曲线

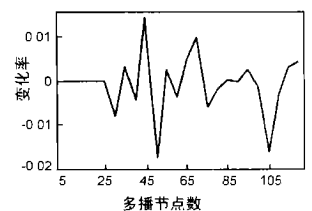


图 2 总费用变化率随端节点数变化曲线

图 3 是端节点数固定为 60 时, 使用 FGMRA 和 GIM 两种算法, 计算最小代价多播生成树的计算时间随网络总节点数

变化曲线. 其中横坐标为网络总节点数, 纵坐标为计算时间(单位毫秒). 从图中可看出, 当端节点数固定时, FGMRA 和 GIM 两种算法计算所有最小代价多播生成树的计算时间随节点数的增加而增加, 但 FGMRA 算法的增长幅度更小, 所需的计算时间明显少于 GIM 算法. 与图 2 类似, 当端节点数固定时, FGMRA 和 GIM 两种算法得到的最小代价多播生成树总费用大致相当.

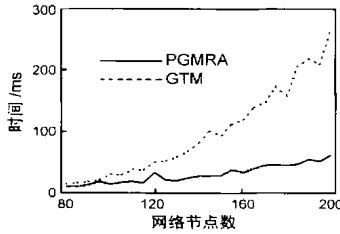


图 3 计算时间随总节点数变化曲线

5 结论

本文通过最大限度地避免重复计算、以存储空间换取计算时间两种策略, 获得了计算效率很高的带宽预留的成组多播快速路由算法 FGMRA. 随机网络模型的仿真结果表明: 使用 FGMRA 得到的多播生成树总费用与 GIM 算法基本相同, 分配失败的概率与 GIM 算法完全相同, 计算效率得到了较大提高. 该算法快速、稳定, 是一种值得推广使用的高效算法.

参考文献:

[1] T Ballardie, P Francis, J Crowcroft. Core based trees (CBT) [A]. Proceedings of the ACM SIGCOMM '93 Conference on Communications Architectures, Protocols and Applications [C]. San Francisco 1993: 85 - 95.

[2] L Kou, G Markowsky, L Berman. A fast algorithm for Steiner trees [J]. Acta Informatica 1981, 15(1): 141 - 145.

[3] X Jia, L Wang. Group multicast routing algorithm by using multiple minimum Steiner trees [J]. Computer Communications, 1997, 20(4): 750 - 758.

[4] H Takahashi, A Matsuyama. An approximate solution for the Steiner problem in graphs [J]. Mathematica Japonica, 1980, 6(3): 573 - 577.

[5] C P Low, N Wang. An efficient algorithm for group multicast routing with bandwidth reservation [J]. Computer Communication, 2000, 23(8): 1740 - 1746.

[6] Bin Wang, Jennifer C Hsu. Multicast routing and its QoS extension: problem, algorithms and protocols [J]. IEEE Network, January/February, 2000, 14(1): 22 - 35.

[7] 胡光岷, 李乐民, 安红岩. 最小代价多播生成树的快速算法 [J]. 电子学报, 2002, 30(5): .

[8] Bernard M Waxman. Routing of multipoint connections [J]. IEEE journal on selected area in communication, 1988, 6(9): 1617 - 1622.

[9] Paolo Narvaez, Kai Yeung Siu, Hong Yi Tzeng. New dynamic algorithms for shortest path tree computation [J]. IEEE/ACM Transactions on Networking, 2000, 8(6): 734 - 746.

作者简介:

胡光岷 男, 副教授, 1966 年 9 月出生于四川省眉山县, 1986 年本科毕业于南京大学计算机软件专业, 1992 年和 2000 年在成都理工大学分别获得硕士、博士学位, 现为电子科技大学宽带光纤传输与通信系统技术国家重点实验室博士后, 主要研究方向为宽带通信网络技术.

李乐民 (见本期第 568 页)