求解有度约束多播路由问题的分布式算法*

刘 莹[1,2], 吴建平[1], 刘三阳[2], 唐厚俭[2]

[1](清华大学 计算机科学与技术系,北京 100084);
[2](西安电子科技大学 数学系,陕西 西安 710071)
E-mail: liuying@csnet1.cs.tsinghua.edu.cn
http://netlab.cs.tsinghua.edu.cn

摘要: 在应用多播(multicast)时,有效的多播路由是关键.现有的多播路由算法一般假定每个节点都支持multicast,但在实际网络中,某些节点并不支持多播,而为了保证网络速度,需限制进行多播所要复制信息的数量.为此,采用度约束来表示每个节点的多播能力,提出了一种有度约束的分布式多播路由算法.算法的复杂度和所需传递信息的数量都低于已有的同类算法.

关 键 词: 多播;度约束;多播路由算法;分布式算法
中图法分类号: TP393 文献标识码: A

多播技术是网络支持多媒体业务的关键技术之一,是指网络中点到多点的通信形式.这是网络中大量业务存在着点到多点通信需求的必然结果.一般要应用多播技术,首先要确定多播路径,这就是多播路由算法所要解决的问题.确定多播路由的一般做法是,构造一棵包括源节点和目的节点,且叶子节点为目的节点的多播树,信息沿这棵树确定的路径进行发送时,信息只需在树的分枝处进行复制,这样能够节省网络资源.大多数多播路由算法假设每个节点都支持多播通信,但在实际网络中,有些节点并不支持多播,而为了保证网络速度,要限制进行多播所要复制信息的数量.为此,本文通过给每个节点指定度约束来表示各个节点不同的多播能力.我们在Jia 的分布式带延迟约束多播路由算法[1]的基础上,提出了解决有度约束多播路由问题的分布式算法.与已有的同类算法相比,其性能有了较大的提高.

1 相关工作

目前已有的带度约束多播路由算法主要由F.Bauer 提出.他首先在文献[2]中研究了有度约束的集中式多播路由算法,其中包括带度约束最短路启发式算法 SPH(shortest path heuristic).SPH 算法的基本思路是,初始时多播树中只包括一个源节点,然后依次将与多播树距离最近且满足度约束的目的节点加入到树中,直到所有目的节点都加入到多播树中为止.带度约束 Kruskal 最短路启发式算法 K-SPH(Kruskal SPH)的做法是,每次连接距离最近的两棵满足度约束的子树,直到所有的目的节点都加入到多播树中为止.以上算法都是无约束多播路由算法的改进.其后,F.Bauer 又提出了分布式带度约束多播路由算法[3],它们是带度约束分布式 SPH 算法、分布式 K-SPH 算法和 Fixup 算法.前面两个算法是对应集中式算法的分布实现.Fixup 算法的基本思想是,首先使用任意一个无约束分布式多播路由算法求得一棵多播树,然后寻找树中违反度约束的点,替换边使其满足度约束.实验结果表明,K-SPH 算法得到的多播树的代价最小,其次是 SPH 算法,最后是 Fixup 算法.这 3 个算法在最坏情

* 收稿日期: 2000-06-21; 修改日期: 2000-09-18
基金项目: 国家自然科学基金资助项目(69972036;90104002)
作者简介: 刘莹(1973－),女,甘肃天水人,博士,主要研究领域为多播路由算法,多播协议;吴建平(1953－),男,山西太原人,博士,教授,博士生导师,主要研究领域为计算机网络体系结构,计算接机系统性能评价;刘三阳(1959－),男,陕西临潼人,博士,教授,博士生导师,主要研究领域为优化理论;唐厚俭(1975－),男,湖南衡阳人,助教,主要研究领域为计算机网络路由算法.

况下的收敛性、所需传递信息数量以及所能解决的网络的个数占实验网络总数的百分比的比较见表 1.其中,m 表示目的节点的个数,n 表示网络节点总数,R 表示图的直径.这里,Fixup 算法的复杂度并没有考虑所采用的无约束算法的复杂度.

Table 1 Comparison of the existing algorithms
表 1 现有算法的比较

Algorithms①	Converge time②	Number of messages③	Percentage of solved networks④ (%)
K-SPH	mnR	mn^2	95.6
SPH	mnR	mn^2	96.4
Fixup	mR	m	99

①算法,②收敛时间,③传递信息的数量,④解决的网络个数的百分比.

2 本文的算法

2.1 带度约束多播路由问题的数学模型及几个定义

通信网络用无向图 $G=(V,E)$表示,V 表示通信节点集;E 表示两节点间的通信链路的全体,在 E 上定义一个代价函数 $c: E \to R^+$,表示通信时链路的代价.有度约束的多播路由算法要构造一棵包含源节点 S 和目的节点集合 $D=\{d_1,d_2,\ldots,d_m\}$的最小代价多播树 T(multicast tree)(这里,树的代价是树的每条边的代价的总和),同时,每个节点要满足各自的度约束:

$$\deg_i \le k_i,\ i=1,2,...,n\ .$$

其中 $\deg_i$ 表示节点 i 在多播树中的度,k_i 为节点 i 的度约束,它表示节点 i 可以向与其邻接的 k_i–1 个节点传送信息,因为节点还要接收信息,所以每个非叶子节点的度约束应大于或等于 2.

下面给出几个相关定义.

定义 **1.** 树 T 到树 T 外的一个节点 v 的最短路是指从节点 v 到树中某一节点 u 的最短路,记为 $SP(u,v)$,其中节点 u 满足下面条件:对树中任意节点 k 有不等式

$$\sum_{l\in SP(u,v)} c(l) \le \sum_{l\in SP(k,v)} c(l)$$

成立.节点 u 称为树 T 中离节点 v 最近的节点.

定义 **2.** 树 T 到树 T 外的一个节点 v 的代价定义为

$$C(T,v)=\sum_{e\in SP(u,v)} c(e)\ .$$

这里,u 是树 T 中离节点 v 最近的节点.

定义 **3.** 称树 T 外的一个节点 w 为距离树 T 最近的节点,如果它满足以下条件:对树 T 外的任一节点 v,有 $C(T,w)\le C(T,v)$ 成立.

本算法假设网络是连通的,每个节点存储了它到网络中所有节点的当前最短路信息,并假设在算法执行期间网络的拓扑结构不发生变化,并且所有的非叶子节点的度约束大于或等于 2.

2.2 算法的关键步骤

本算法的基本思想如下:多播树 T 初始时只包括源节点 S,然后将目的节点集合中与多播树 T 距离最近且满足度约束的那个目的节点加入到树中,同时,此目的节点到树 T 的最短路上的点也加入到树中.如此这样做直到所有的目的节点都在树中为止.

算法用数组 TD 记录已形成的多播树到每个目的节点的最短路信息,其中 $TD[d_i].c$ 记录树 T 到目的节点 d_i 的代价,$TD[d_i].treenode$ 记录树 T 中离目的节点 d_i 最近的节点,$TD[d_i].tag$=no 表示目的节点 d_i 不在树 T 中;$TD[d_i].tag$=yes 表示目的节点 d_i 已在树 T 中.此外,每个节点 u 都保存有一个局部路由表 R_u,其中 $R_u[d_i].c$ 记录此节点 u 到目的节点 d_i 的最短路的值,$R_u[d_i].next$ 记录 u 沿最短路到目的节点 d_i 的下一个邻接节点.degree[•]记录节点当前的度,每个节点的度约束用 deg$cons$[•]记录.算法有以下几个关键步骤:

(1) 初始化

算法初始时,每个节点都处于睡眠状态,接收到建立多播树消息(initial)的节点成为源节点 S.此消息包括 degree[•]和 deg$cons$[•].源节点初始化操作如下:对所有目的节点设置 $TD[d_i].c=R_s[d_i].c$,$TD[d_i].treenode=S$,$TD[d_i].tag$=no.然后,源节点选择 TD[•].c 值最小的那个目的节点为加入到树的下一个节点,记此节点为 a,同时设置 $TD[a].tag$=yes;deg$ree[s]$++.最后,源节点发送一个建立消息(setup)给 $R_s[a].next$ 记录的节点记为 u,消息包含 TD.

(2) 接收和传送 setup 消息

当节点 u 接收到 setup 消息以后,主要进行的操作是:首先改变自身的当前度,并判断当前度是否等于度约束,如果等于度约束,那么节点 u 只向 $R_u[a].next$ 记录的节点传递 setup 消息;否则,节点首先更新 TD 的值,然后进行 setup 消息的向下传递.

节点 u 改变自身度操作如下:如果 $u=a$,即消息已到达所要加入的目的节点,那么 deg$ree[u]$=1;否则,deg$ree[u]$=2.

如果 deg$ree[u]$<deg$cons[u]$,那么节点 u 进行更新 TD 的操作:① 节点 u 首先更新其局部路由表 R_u,如果它在求到某一目的节点的最短路时发现其中存在不满足度约束的节点,那么节点 u 将其局部路由表中这个目的节点对应的 R_u[•].c 设为无穷大;② 然后,节点 u 对 TD[•].tag=no 的目的节点 d_i 按它们的 R_u[•].c 值从小到大排序;依次对这些目的节点进行检验,如果 $R_u[d_i].c<TD[d_i].c$,那么 $TD[d_i].c=R_u[d_i].c$,$TD[d_i].treenode=u$,这样的操作最多进行 deg$cons[u]$−deg$ree[u]$次.

当 setup 消息到达要连接的目的节点之后,下一个要连接的目的节点就是 TD[•].c 值最小的那个节点,仍记为 a.这时,刚加入树中的目的节点向 $TD[a].treenode$ 记录的节点,记为 t,发送一个分枝消息(fork),至此,一次 setup 消息传递结束.

(3) 传送分枝消息和撤消消息

节点 t 接到 fork 消息之后,首先检查 $R_t[a].next$ 记录的节点是否已在树中,如果其不在树中,那么节点 t 向此节点发送 setup 消息;否则,节点 t 向发送 fork 信息的节点发送撤消消息(delete),并设置 $TD[a].tag$=no;$TD[a].c=\infty$.当节点接收到撤消消息之后,重新选择一个目的节点,再发送分枝消息.

当所有的目的节点都已经包括到多播树中时,最后加入到树中的目的节点向源节点发送完成信息(complete),算法完成.这样,多播路由树的寻找和建立同时完成.

2.3 算法的讨论

定理 **1.** 本算法在执行过程中不会产生回路.

证明:用反证法,假设算法在执行过程中遇到回路(如图 1 所示).

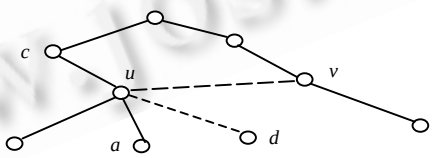

Fig.1 An example figure for Theorem 1

图 1 定理 1 例图

假设将目的节点 d 连接到树中时,回路产生,见图 1 的虚线;这里分两种情况加以证明.

情况 1. 节点 u 比节点 v 先加入到树中.从图中可看出,目的节点 d 距离节点 u 更近,但没有从节点 u 加入.这只能是因为节点 u 的度已满足,节点 v 在计算到节点 d 的最短路时发现节点 u 的度已满足,它会返回一个无穷大值,此时,算法不会选择这条路.因此,这种情况不会发生.

情况 2. 节点 u 比节点 v 后加入到树中.考虑一般情况,节点 u 不是目的节点,而是加入目的节点 a 时将其加入的.从图中可看出,目的节点 d 距离节点 u 更近,但没有从节点 u 加入,这只能是因为节点 u 的度已满足.算法中目的节点的加入都是通过最短路,因此节点 v 到目的节点 d 也是通过最短路,从而可以得,知边(u,v)是 u 到 v 的

最短路.如果边(u,v)的代价小于边(c,u)的代价,那么目的节点 a 将通过 v 加入到树中.而实际上,目的节点 a 是通过节点 c 加入到树中的,这说明边(u,v)的代价大于边(c,u)的代价,因此,路径$\{c,u,d\}$的代价小于路径$\{v,u,d\}$的代价.所以,节点 d 不会通过节点 v 加入,回路也不可能产生.同理可证,当 u 是目的节点时,回路依然不可能产生.

综合上述两种情况,算法在执行过程中不会产生回路. □

从上面的证明可看到,算法会有重复链路产生,在情况 2 中,节点 u 的度约束已达到,但节点 C 仍可能通过节点 u 加入节点 d,这样就有重复链路产生了.为此,算法规定,每个节点在接收到 fork 信息时,要检查下一个节点是否已在树中,如果在树中就撤消此次操作,从而避免了这一情况的发生.

定理 **2.** 在最坏情况下,本算法建立一棵包含有 m 个目的节点的多播树算法需 $O(m^2)$次信息传递.

证明: 这里每个节点表示实际的一个路由器,算法在每个路由器上进行,算法将每个目的节点加入到树中所需的控制信息算为一次信息传递,而不考虑中间节点的信息传递.本算法是串行的,如果不产生回路,那么需要用到如下信息:

(1) 连接 m 个目的节点需要从源节点或 fork 节点发送 m 个 setup 信息到目的节点;

(2) 至多 $m-1$ 次 fork 信息从 $m-1$ 个目的节点发出;

(3) 如果出现 k 次重复链路,则需要 k 个 delete 信息和 k 个 fork 信息;

(4) 最后一个完成信息.

本算法在最坏情况下需要 $2(m+k)$次信息传递.这里,最坏情况为每次在加入一个目的节点时,其余未加入的目的节点都出现重复链路的情况,此时,

$$k=(m-2)+(m-3)+...+1=\frac{(m-2)(m-3)}{2},$$

因此,算法共需 $O(m^2)$次信息传递.假设每次信息传递需一个单位时间,那么本算法的收敛时间为 $O(m^2)$. □

3 相关工作比较

由于在一般多播路由中,目的节点个数是网络节点个数的 30%,因此,很容易看出本算法所需传递信息的次数和算法的收敛性都优于 F. Bauer 的算法.并且,本算法同分布式最小生成树算法(MST)[4]相比也表明本算法的性能更好.表 2 给出了这几个算法的比较.其中,m 表示目的节点的个数,n 表示网络节点总数,R 表示图的直径.

Table 2 Comparison of the existing algorithms and our algorithm

表 **2** 本算法与已有算法的比较

Algorithm①	Number of messages②	Converge time③
Our algorithm④	m^2	m^2
MST⑤	$5n\log_2 n+2\|E\|+m5n$	$5n\log_2 n$
F. Bauer's distributed SPH⑥	mn^2	mnR

①算法,②传递信息的数量,③收敛时间,④本算法,⑤MST 算法,⑥F. Bauer 的分布式 SPH 算法.

本算法的另一个优点是,路由的选择和确立是同时完成的.在本算法中,依次将每个目的节点连接后,路由就建立了.在集中式算法中,路由确立后还需要 $O(m)$时间去建立路由.因此,本算法节约了建立路由的时间.

本算法的基本思想与 SPH 算法的思想相似,而在所需传递信息的数量和收敛性方面比 F.Bauer 的分布式 SPH 算法好.但是,F.Bauer 的分布式 SPH 算法不会出现重复链路的情况,因为他的算法在实现过程中每添加一个目的节点,树中所有的节点都会重新计算最短路,寻找下一个目的节点,这样就避免了重复链路和回路的产生.但是,这样会导致计算次数增多和收敛时间较长.

我们采用 F.Bauer 的实验模型对本算法与集中式 SPH 算法和分布式 SPH 算法进行了比较.F.Bauer 采用文献[5]提出的方法产生网络,即网络的 n 个节点从一个坐标平面中随机选取,每对节点 u,v 之间存在边的概率为

$$\Pr(u,v)=\beta\exp\frac{-d(u,v)}{2\alpha L},$$

其中 $d(u,v)$为这两点间的距离.参数α,β从区间(0,1)中选取.α越大,距离远的节点之间边存在的概率就越大;β越大,网络就越稠密.每条边的代价指定为两点间的距离.网络中每个非叶子节点都给出一个度约束,其值是 2 到其实际度值之间的一个随机数.实验主要有两个方面,一方面是比较每种算法能找到解的网络的百分比;另一方面

是对算法所求多播树代价进行比较.本文采用上面的方法生成网络,对每种网络试验 200 次.首先,我们对节点数分别为 50,70,90 和 100 的网络进行了实验,实验结果表明,本算法对所实验网络的 96%能找到解.而分布式 SPH 算法也可对 96%的网络找到解.其次,本算法对节点数分别为 20~100 的网络求得多播树,比较几个算法的代价性能,实验显示,本算法求得的多播树的代价与分布式 SPH 算法得到的多播树的代价基本相同,相比之下,集中式 SPH 算法的多播树代价最小.然后,我们固定网络节点个数,变换目的节点个数进行求解,实验结果表明,在这种情况下,本算法的多播树代价与分布式 SPH 算法的多播树代价依然接近,而集中式 SPH 算法的性能仍然相对最好.综合实验结果可知,本算法与分布式 SPH 算法在解的代价和所能解决的网络个数两方面的性能相似.集中式算法解的性能较好是因为它利用了全局信息.但是,本算法的运行时间少于分布式 SPH 算法所用的时间,这是因为本算法的时间复杂度和所需传递信息的数量都小于分布式 SPH 算法.

4 结 论

本文提出了一个解决有度约束的多播路由问题的分布式算法,它所需传递的信息数量少、时间复杂度较低,与已有的分布式算法相比,本算法的性能较好.实验结果也说明了这些优势.

References:

[1] Jia Xiao-hua. A distributed algorithm of delay-bounded multicast routing for multimedia application in wide area networks. IEEE/ACM Transactions on Networking, 1998,6(6):828~837.

[2] Bauer, F., Varma, A. Degree-Constrained multicasting in point-to-point networks. In: Proceedings of the Conference on Computer Communications of IEEE INFOCOM'95. Los Alamitos, CA: IEEE Computer Society Press, 1995. 369~376. http://ieeexplore.ieee.org/lpdocs/epic03/.

[3] Bauer, F. Distributed degree-constrained multicasting in point-to-point networks. Technical Report, Department of Computer Science, University of California at Santa Cruz, 1995. ftp://ftp.cse.ucsc.edu/pub/tr/ucsc-crl-95-09.

[4] Gallager, R., Humblet, P., Spira, P. A distributed algorithm for minimum-weight spanning trees. ACM Transactions on Programming Language and Systems, 1983,5(1):66~77.

[5] Waxman, B.M. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 1988,6(9):1617~1622.

A Distributed Algorithm for Degree-Constrained Multicast Routing*

LIU Ying[1,2], WU Jian-ping[1], LIU San-yang[2], TANG Hou-jian[2]

[1](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*);

[2](*Department of Mathematics, Xidian University, Xi'an* 710071, *China*)

E-mail: liuying@csnet1.cs.tsinghua.edu.cn

http://netlab.cs.tsinghua.edu.cn

Abstract: In order to support multicast, efficient multicast routing is crucial. Many present multicast routing algorithms assume that every node in the network support multicast. But in real networks, some nodes may not support multicast, others may limit the number of multicast copies in order to ensure network speed. Thus, the multicast capability of each node is represented in this paper by a degree-constraint. A distributed degree-constrained multicast routing algorithm is proposed which has smaller time complexity and needs smaller number of messages than other existing algorithms.

Key words: multicast; degree-constraint; multicast routing algorithm; distributed algorithm

* Received June 21, 2000; accepted September 18, 2000

Supported by the National Natural Science Foundation of China under Grant Nos.69972036, 90104002