文章编号:1001 -9081(2010)05 -1176 -03

时延受限组播路由的最短路径加速算法求解

李元臣, 刘维群
(洛阳师范学院 信息技术学院,河南 洛阳 471022)
(yuanchen_li@126.com)

摘　要:分析了时延受限的 Steiner 树问题,总结了在构建组播树过程中的代价和计算复杂度变化规律,并根据实际网络环境,从优化最短路径出发,提出了一种基于优化最短路径的时延受限组播路由算法 AOSPMPH。该算法以 MPH 算法为基础,利用 Floyd 最短路径优化算法求出节点对之间的最短路径,选择满足时延要求的最小代价路径加入组播树,进而产生一棵满足时延约束的最小代价组播树。仿真结果表明,AOSPMPH 不但能正确地构造时延约束组播树,而且其代价和计算复杂度与其他同类算法相比得到了优化。

关键词:Steiner 树;MPH 算法;Floyd 最短路径优化;启发式算法;组播通信

中图分类号: TP393.02　　**文献标志码**:A

Delay-constrained multicast routing algorithm based on optimized Floyd shortest path algorithm

LI Yuan-chen, LIU Wei-qun
(*College of Information Technology, Luoyang Normal University, Luoyang Henan* 471022, *China*)

Abstract: An important issue in multicast communication is how to create an efficient and robust Steiner tree through using multicast routing algorithm. Based on the analysis of delay-constrained Steiner tree, the cost and computational complexity when constructing a multicast tree, starting from the practical requirements and optimizing shortest paths, a new algorithm named Algorithm of Optimizing Shortest Path based on MPH (AOSPMPH) was proposed. On the basis of Minimum cost Paths Heuristic (MPH), the proposed algorithm found the shortest path between nodes using optimized Floyd shortest path algorithm and selected the minimum cost path to meet the requirements of delay constraint to add into the multicast tree. By this way, a low cost multicast tree could be constructed. The simulation results show that AOSPMPH not only can construct delay-constrained multicast tree correctly, but also is of less cost and lower computational complexity than those of many other multicast algorithms.

Key words: Steiner tree; Minimum cost Paths Heuristic (MPH) algorithm; optimum of shortest path algorithm devised by Floyd; heuristic algorithm; multicast communication

0　引言

对于组播路由问题,一般要建立代价最小的组播树,而最优 Steiner 树被认为是实现组播通信的最好办法之一,Steiner 树理论及其算法也成为求解组播树的基础。在构造组播树时,一般用树的代价来衡量组播树的好坏。在网络中寻找覆盖给定节点集合的代价最小的生成树问题在数学上被称为 Steiner 树问题,这是一个 NPC 问题[1]。求解 NPC 问题的最优算法无法在多项式时间内完成,因此,需要构造有效的启发式算法,降低复杂度,节省网络带宽资源,提高组播通信的资源利用率,减少拥塞,降低网络负载,在性能上逼近理论算法。Steiner 树问题的启发式算法本质上都是借鉴最小生成树和最短路径算法的思想,其优点在于算法简单、高效,而这正是组播路由算法必需的。目前 Steiner 树问题启发式经典算法很多,如 MPH 算法[2]、KMB 算法[3]、GRD 算法[4]、ADH 算法[5]等,它们都可用来求解最小代价组播树。但是经典算法存在复杂度较高、对组播成员的动态变化不敏感等缺陷,使得它们难以在实际网络环境中应用。因此,本文基于 MPH 算法[2]提出了一种利用 Floyd 最短路径动态优化的方法求解时延受限组播路由算法,仿真表明,该算法复杂度低、综合性能好,能应用于实际网络。

1　网络模型与研究内容定义

组播问题的网络模型可表示为赋权连通无向图 $G = (V, E)$,其中:V 是网络中节点的集合,每个节点代表主机或路由器,$n = |V|$ 表示节点数;E 为网络中通信链路即边的集合。对于 E 中的每一条边 $e = (x,y)(e \in E, x, y \in V)$,定义代价函数 $C(e): E \to \mathbf{R}^+$ 和时延函数 $D(e): E \to \mathbf{R}^+$,$\mathbf{R}^+$ 为正实数集。设 s 为组播源节点,$D \subseteq V - s$ 为组播目的节点集,$m = |D|$ 为组播节点个数。

定义 1　路径。图 $G = (V,E)$ 中,如果存在节点序列:

$$p(v_1, v_l) = (v_1, v_2, \cdots v_i, \cdots, v_l);$$
$$1 < l \leqslant n, 1 \leqslant i \leqslant l, v_i \in V \quad (1)$$

则称 $p(v_1, v_l)$ 为从节点 v_1 到节点 v_l 的一条路径。

定义 2　时延和代价。路径 $p(u,v)$ 的时延为链路时延的和,代价为链路代价的和。分别表示为:

收稿日期:2009 -11 -17;**修回日期**:2010 -01 -03。
基金项目:河南省自然科学基金资助项目(2008B520027);河南省高等学校青年骨干教师资助计划项目(2006104)。
作者简介:李元臣(1966 -),男,河南新蔡人,副教授,硕士,主要研究方向:网络性能分析、网络管理、嵌入式系统;　刘维群(1971 -),女,河南内乡人,副教授,硕士,主要研究方向:网络管理、嵌入式系统、神经网络。

$$D(p(u,v)) = \sum_{e \in p(u,v)} D(e) \quad (2)$$

$$C(p(u,v)) = \sum_{e \in p(u,v)} C(e) \quad (3)$$

定义 3　Steiner 树。给定图 $G = (V,E)$ 和源节点 s、目的节点集 $D \subseteq V - s$，若从图 G 中找出覆盖 D 中所有节点的最小生成树 T，即使得：

$$\min\left\{ \sum_{v \in D, e \in T} C(e) \right\} \quad (4)$$

成立，该最小生成树 T 称为 Steiner 树。

定义 4　时延约束 Steiner 树。Steiner 树 T 中，给定时延阈值 Δ（正实数），若 T 满足：

$$\sum_{e \in p(s,v), v \in D} D(e) \leqslant \Delta \quad (5)$$

则称树 T 为时延约束的 Steiner 树，记作 T_s。

时延受限的组播路由问题主要研究时延约束 Steiner 树的构建问题，即设计高效的组播路由算法，使生成的时延约束 Steiner 树代价最优。

2　MPH 加速优化算法 AOSPMPH

MPH 算法代价较低，是公认的求解 Steiner 树问题的优秀启发式算法。其基本思想是每次从目的节点集 D 尚未添加到生成树的节点中选取到树 T 具有最小代价路径节点 v，将其及其最小代价路径加入树 T 中。

基本算法如下所示。

步骤 1　初始化。$i \leftarrow 1, T_i \leftarrow \{v_1\}, V_i \leftarrow \{v_1\} (v_1 \in D)$；

步骤 2　对于节点 $v_i (v_i \in D - \{v_1\}, i = 2, \cdots, m)$，选择到 T_{i-1} 的最小代价路径 $p = (v_i, \cdots, V_{i-1})$，将其加入到树 T_i 中，即 $T_i \leftarrow T_{i-1} \cup p$；

步骤 3　重复步骤 2 直到所有的目的节点都加入到 T_i 中。

MPH 算法中，将一个节点加入到生成树 T 之前，该节点必须是余下的节点中距离树 T 最近的，即将距离树 T 代价最低的 D 中尚未添加到生成树的节点通过代价最短路径加入到生成树 T 中，要执行 m 次最短路径算法，搜索范围大，用时较长。文献[6] 证明 MPH 算法的时间复杂度为 $O(mn^2)$，文献[7] 通过引进共享边来改进 MPH 算法的代价，使其应用到时延约束的路由组播树的构建中，取得了明显的效果。本文根据 MPH 算法中构建 Steiner 树的特性，改进组播树构造方法，将其推广到时延约束的求解组播路由问题，提出一种优化最短路径的时延受限组播路由算法（Algorithm of Optimizing Shortest Path based on MPH, AOSPMPH）。利用 Floyd 最短路径优化算法[8] 求出节点对之间的最短路径，满足时延要求时，通过选择与当前组播树 T 有最小代价路径的节点构成的子树加入组播树；否则，选择最小时延路径加入组播树 T，从而生成满足时延约束的 Steiner 树。基本过程如下所示。

步骤 1　任选一源节点 $v_i (i = 1, 2, \cdots, n)$，初始化 Steiner 树 $T_s \leftarrow \{v\}$。

步骤 2　运用 Optimized_Floyd 算法（Floyd 最短路径优化算法）求出图 G 各节点之间最短路径树 $T_{\min}$ 的时延及其代价。

步骤 3　每个 $T_{\min}$ 的时延和时延阈值 Δ 比较，寻找满足时延约束条件的 $T_{\min}$。

步骤 4　对于 $u \in D - \{v\}$，若时延 $T_{\min} \leqslant \Delta$，则 $T_s \leftarrow T_s \cup p(u,v) (p(u,v) \in T_s)$；否则 $T_s \leftarrow T_s \cup p(u,v) (p(u,v) \in T_{\min})$。

步骤 5　执行步骤 4，直到所有目的节点全部加入 T_s。

根据文献[8]，Optimized_Floyd 算法的优化思路是按照路径长度递增的次序产生源节点到各节点的最短路径，每次经过的中间节点必须是入度大于 0 的点，即优化选择中间节点以裁剪路径中不可能经过的中间节点所消耗的无关循环。Optimized_Floyd 的时间复杂度为 $O(n^2 + |AV| \times e^2/n^2)$（$e$ 为边数，n 为节点数，AV 为入度大于 0 的节点集合，$|AV| \leqslant n$）。上述算法利用 Optimized_Floyd 算法求节点对之间的最短路径树，避免了 MPH 算法中无效的循环比较，在计算最短路径时使用序号矩阵动态记录使路径变短的节点集，这样在寻找最短路径时迭代速度快、计算量小、效率高。AOSPMPH 的伪代码表示如下：

```
AOSPMPH (G,D,Δ)
T_s ← v_1; Dest_vertex ← V − {v_1};
For each u,v ∈ V do
  {T_min ←Optimized_Floyd (G,D,u,v);
  If D(T_min) > Δ then halt;
    C(p(u,v)) ←Optimized_Floyd (G,D,u,v);}
Repeat
  Select vertex x ∈ Dest_vertex, y ∈ T_s and min(C(p(x,y)));
                         /* p(x,y) 路径树的代价最优 */
  If D(p(x,y)) − D(T_s) ≤ Δ then          /* 满足时延 */
    {T_s ← T_s ∪ p(x,y);
    Dest_vertex ← Dest_vertex − {x}};
  Else
    {T_s ← T_s ∪ min(p(x,v_1));
                    /* p(x,v_1) 为节点 x 到 v_1 的路径 */
    Dest_vertex ← Dest_vertex − {x}};
Until Dest_vertex is empty;
Output T_s;
```

3　算法分析

定理 1　AOSPMPH 不会构成环路。

证明　Optimized_Floyd 算法是对 Floyd 算法的优化，不会产生环路。对于 $\forall v \in D - T_s$，若 v 到 T_s 的最小时延路径 $p(v,u) (u \in T_s)$ 存在环，则该环和树 T_s 相连的节点数不小于 2，这和定义 1 及定义 4 是矛盾的。由于 AOSPMPH 是利用 Optimized_Floyd 算法求 D 中尚未连接的节点到树 T_s 的最短路径，因此不会形成环路。

定理 2　AOSPMPH 能构造出满足时延约束的且包含所有组播节点的组播树。

证明　设 v_1 为初始节点，$\forall v \in D - \{v_1\}$。如果 $D(p(v,v_1)) \leqslant \Delta$，根据算法，则 T_s 的每个子树都是由边 $p(v,v_1)$ 构成的，因此满足时延约束的组播树存在。算法中树的构造是从组播集 D 中选取节点加入到树 T_s 中，直到 D 为空，因此所构成的树包含了所有的组播节点。

定理 3　AOSPMPH 的时间复杂度为 $O(n^3)$。

证明　文献[8]已经证明 Optimized_Floyd 算法的时间复杂度为 $O(n^2 + |AV| \times e^2/n^2)$。循环建树时，以 v_1 作为初始组播树 T_s，从组播集 D 中逐个选取节点加入到树 T_s 中。根据 MPH 算法，此过程的时间复杂度为 $O(m^2 n)$。因此，AOSPMPH 的时间复杂度为 $O(n^2 + |AV| \times e^2/n^2) + O(m^2 n)$。

因为 $|AV| \leqslant n, e \leqslant n(n-1)/2, m < n$，所以 AOSPMPH 的最坏时间复杂度为：

$$O(n^2 + |AV| \times e^2/n^2) + O(m^2 n) =$$

$$O(n^2 + n(n(n-1)/2)^2/n^2) + O(m^2 n) =$$

$$O(n^2+n(n-1)^2/4)+O(m^2n)=$$
$$O(n^2)+O(n^3)+O(m^2n)=$$
$$O(n^3)$$

事实上，$O(n^3)$ 为图 G 为完全图时的计算复杂度，当图为稀疏图时，算法的执行速度明显加快，仿真实验可以看出 AOSPMPH 的复杂度要小于 $O(mn^2)$。

4 仿真实验与性能分析

实验采用改进的 Waxman 方法产生随机网络模型[4,7]。网络节点随机分布在 200×200 笛卡尔坐标系内(相当于 2000 km×2000 km的矩形区域)，边存在的概率为：

$$p(u,v)=\beta\exp\left(\frac{-d(u,v)}{\alpha L}\right) \tag{6}$$

其中：$d(u,v)$ 表示节点 u 到 v 的距离；L 是任意两个节点间的最大距离。参数 α、β 在 0 和 1 间取值，其值的选取可以控制生成的图的特性：较小的 α 将增大短链路的密度，较大的 β 将导致较高的链路密度。链路的代价定义为两个节点的距离，链路的时延定义为代价乘以一个 0 到 1 之间的随机数。实验中，$\alpha=0.3,\beta=0.2$，实验次数为 100，然后取其平均值作为测量值。

求解满足时延约束的 Steiner 树的启发式算法中，经典的有 KPP 算法[1]、BSMA 算法[9]等，实验把 AOSPMPH 和上述两种算法作了比较，得出了如图 1～4 所示的统计图。

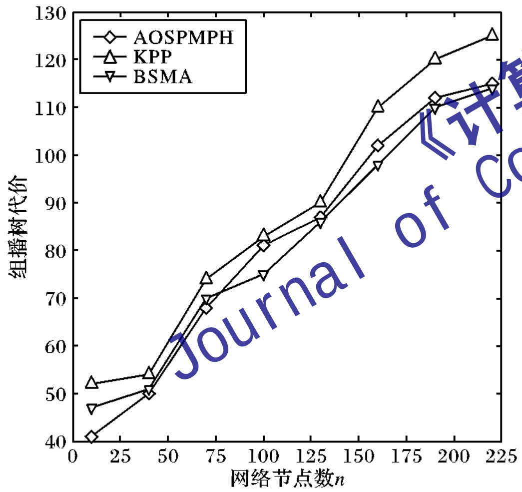

图 1 网络节点数与组播树代价

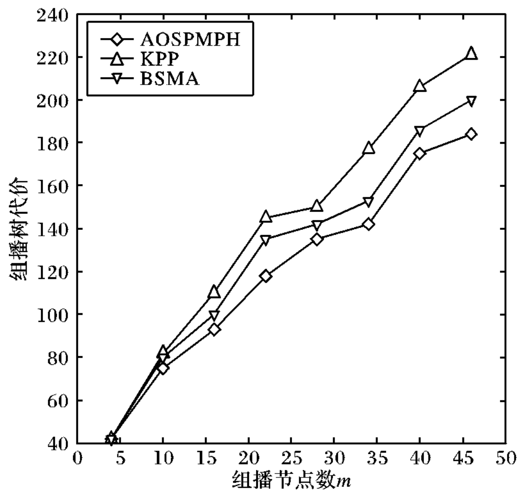

图 2 组播节点数与组播树代价

从图 1 所示的曲线可以看出，网络节点数较少时，AOSPMPH 的代价相比 KPP 算法和 BSMA 算法要少。随着节点的增加，AOSPMPH 的代价小于 KPP 算法略高于算法 BSMA。当保持网络节点不变，增加组播节点数时，AOSPMPH 的性能要优于 KPP 算法和 BSMA 算法，如图 2 所示。图 3 给出了时延约束阈值与组播树代价关系，AOSPMPH 的性能是 KPP 算法和 BSMA 算法的折中。在计算时间上，AOSPMPH 比 KPP 算法和 BSMA 算法用时要少，这主要因为 KPP 算法和 BSMA 算法没有考虑网络中节点和边的变化情况，其时间复杂度分别与网络节点数成立方关系。而 AOSPMPH 的时间复杂度虽然也与网络节点数成立方关系，但它充分考虑了实际网络拓扑图多为稀疏图，大大减少了计算量，图 4 所示的实验结果也验证了这种优化效果。

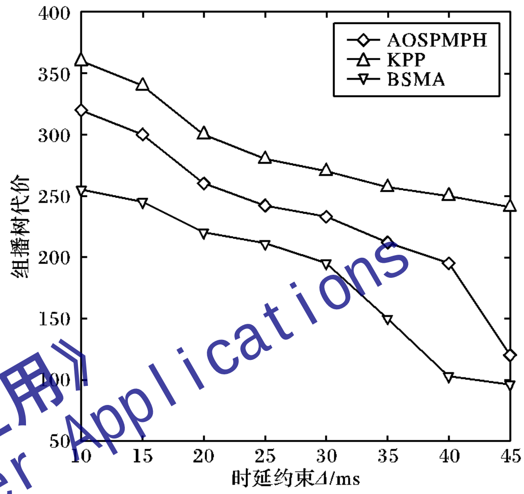

图 3 时延约束与组播树代价

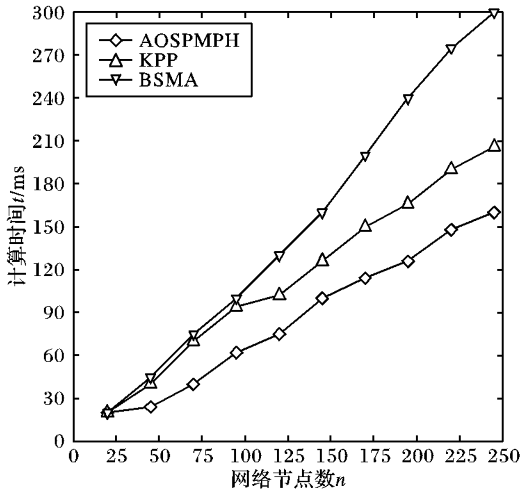

图 4 计算时间与网络节点数

国内成熟的对 MPH 的改进算法有 DCMPH 算法[6]、TPMPH 算法[10]，其时间复杂度分别为 $O(mn^2)$，相比较 MPH 算法，前者有较好的代价性能，后者有较低的计算性能。本文提出的 AOSPMPH 的时间复杂度为 $O(n^3)$，在节点个数较少时计算时间和代价明显优于 DCMPH 算法和 TPMPH 算法，随着节点增加则略低于 DCMPH 算法和 TPMPH 算法。但就现实网络而言，AOSPMPH 综合性能和可行性较高。

5 结语

本文对时延受限的组播路由问题进行了研究，在分析研究Steiner树理论和MPH算法的基础上提出了基于Floyd最

(下转第 1182 页)

BF-MIMO 系统的信道容量大于 MIMO 的信道容量，且增加的程度随着收发天线对的增加而变大。如在信噪比 $SNR = P_t/\sigma^2 = 10$ dB 条件下，采用单发单收的 BF-MIMO 系统较 MIMO 系统的信道容量增加值约为 2 bps/Hz；而在相同信噪比条件下，采用双发双收的 BF-MIMO 系统相对于 MIMO 系统所获得的信道容量增加值达到了约 4 bps/Hz；当采用四发四收时，容量增加值更是达到了约 7.6 bps/Hz。

图 3 给出了采用两发两收、发送子天线阵元数取不同值时的信道容量随信噪比变化曲线。从图 3 中可以看出，当系统收发天线对不变时，随着发送子天线阵中阵元数的增加，信道容量在信噪比较高时类似线性增大。如在本仿真条件下，当信噪比大于 10 dB 时，随着阵元数每增加一倍，BF-MIMO 系统信道容量的增加值约为 2 bps/Hz。

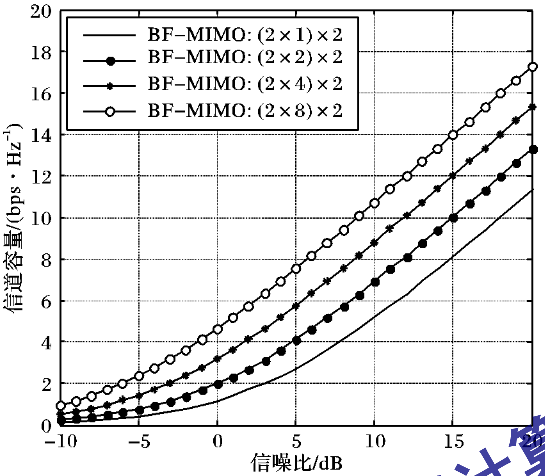

图 3 发射天线阵采用不同阵元数时的信道容量

4 结语

本文将智能天线中的波束赋形技术和多输出多输出技术结合，建立了一种通用的 BF-MIMO 下行链路系统模型，得到了 BF-MIMO 系统在平坦衰落环境中的组合信道矩阵形式。分析讨论了赋形加权矢量的选取问题，得出了赋形加权后的系统信道容量闭环表达式。理论分析和仿真结果表明：BF-MIMO 兼具波速赋形和多入多出的优点，能够获得较 MIMO 系统更高的信道容量。需指出的是，本文采用的基于导向矢量的赋形机制并不是最优的，且信道容量的分析也是在该赋形机制以及发端不知CSI的假设条件下获得的。因此，在 BF-MIMO 系统中如何设计性能更优的波束赋形机制，在该赋形机制下以及发端已知 CSI 时的系统信道容量提升问题如何，都有待于在今后的工作中加以研究。

参考文献：

[1] PAULRAJ A J, GORE D A, NABAR R U, *et al.* An overview of MIMO communications—a key to gigabit wireless [J]. Proceedings of the IEEE, 2004, 92(2): 198 – 218.

[2] PAULRAJ A, NABAR R, GORE D. Introduction to space-time wireless communications [M/OL]. New York: Cambridge University Press, 2003 [2009 – 06 – 12]. http://assets. cambridge. org/97805218/26150/sample/9780521826150ws. pdf.

[3] WINTERS J H. Smart antennas for wireless systems [J]. IEEE Personal Communications, 1998, 5(1): 23 – 27.

[4] LIBERTI J C, RAPPAPORT T S. Smart antennas for wireless communications: IS-95 and third generation CDMA applications [M]. Upper Saddle River, NJ, USA: Prentice Hall, 1999.

[5] GESBERT D, SHAFI M, SHIU D-S, *et al.* From theory to practice: An overview of MIMO space-time coded wireless systems [J]. IEEE Journal on Selected Areas in Communications, 2003, 21(3): 281 – 302.

[6] LEI ZHONGDING, CHIN F P S, LIANG YING-CHANG. Combined beamforming with space-time block coding for wireless downlink transmission [C]// VTC 2002-Fall: Proceedings of IEEE 56th Vehicular Technology Conference. Washington, DC: IEEE Press, 2002, 4: 2145 – 2148.

[7] ALAMOUTI S M. A simple transmit diversity technique for wireless communications [J]. IEEE Journal on Selected Areas in Communications, 1998, 16(8): 1451 – 1458.

[8] ZHU F, LIM M S. Combined beamforming with space-time block coding using double antenna array group [J]. Electronics Letters, 2004, 40(13): 811 – 813.

[9] LIN MIN, LI MIN, YANG LUXI, *et al.* Combined adaptive beamforming with space-time block coding for multi-antenna communications [J]. Science in China Series F: Information Sciences, 2008, 51(12): 2062 – 2073.

[10] KIM I H, LEE K, CHUN J. A MIMO antenna structure that combines transmit beamforming and spatial multiplexing [J]. IEEE Transactions on Wireless Communications, 2007, 6(3): 775 – 779.

[11] 方舒，李立华，张平. 一种基于发送赋形的空间复用 MIMO 方案 [J]. 系统仿真学报, 2008, 20(16): 4216 – 4220.

（上接第 1178 页）

短路径优化的时延受限组播路由算法。该算法基于 MPH 算法低代价优势，利用 Floyd 最短路径优化思想，能快速生成满足时延约束的组播树。仿真实验表明，与 KPP 算法和 BSMA 算法相比，该算法时间复杂度小、代价低，综合性能较好，能够应用于实际网络环境。

参考文献：

[1] KOMPELLA V P, PASQUAL J C, POLYZOS G C. Multicast routing for multimedia communication [J]. IEEE/ACM Transactions on Networking, 1993, 1(3): 286 – 292.

[2] WINTER P. Steiner problem in networks: A survey [J]. Network, 1987, 17(2): 129 – 167.

[3] KOU L, MARKOWSKY G, BERMAN L. A fast algorithm for Steiner trees in graphs [J]. Acta Informatica, 1981, 15(2): 141 – 145.

[4] WAXMAN B M. Routing of multipoint connection [J]. IEEE Journal on Selected Areas in Communication, 1988, 6(9): 1617 – 1622.

[5] RAYWARD-SMITH V J, CLARE A. On finding Steiner vertices [J]. Networks, 1986, 16(3): 283 – 294.

[6] 周灵，孙亚民. 基于 MPH 的时延约束 Steiner 树算法[J]. 计算机研究与发展, 2008, 45(5): 810 – 816.

[7] 李元臣，刘维群. 基于共享边的时延约束组播路由算法[J]. 计算机应用, 2009, 29(11): 1213 – 1215.

[8] 李洪波，王茂波. Floyd 最短路径算法的动态优化[J]. 计算机工程与应用, 2006, 42(34): 60 – 63.

[9] PARSA M, ZHU QING, GARCIA-LUNA-ACEVES J J. An iterative algorithm for delay-constrained minimum-cost multicasting [J]. IEEE/ACM Transactions on Networking, 1998, 6(4): 461 – 474.

[10] JIANG TING-YAO, LI QING-HUA. An improved multicast routing algorithm [J]. Journal of Shanghai University: English Edition, 2004, 8(3): 317 – 321.