

一种非精确状态下满足多 QoS 约束的 动态组播路由算法

王 立, 李增智, 宋承谦, 闫 焱

(西安交通大学计算机系统结构与网络研究所, 陕西西安 710049)

摘 要: 提出了一种非精确状态下满足多 QoS 约束的动态组播路由算法)) DMIQ, 该算法采用改进的 BF 算法作为路径搜索算法, WFQ 作为分组调度机制, 链路延迟不确定且服从均匀分布, 能在非精确状态且满足带宽、延迟抖动和丢包率约束的前提下确定具有最小跳数和开销的动态组播路由. 仿真实验表明, DMIQ 能在非精确状态且满足多 QoS 约束的前提下建立动态组播路由, 在路由请求平均成功率、平均延迟和平均跳数等衡量指标下表现出良好的性能.

关键词: 非精确; 动态组播路由; QoS; DMIQ; BF 算法

中图分类号: TP393 **文献标识码:** A **文章编号:** 03722112 (2004) 08124204

A Dynamic Multicast Routing Algorithm with Inaccurate Information Satisfying Multiple QoS Constraints

WANG Li, LI Zengzhi, SONG Chengqian, YAN Yan

(Institute of Computer System Architecture & Network, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China)

Abstract: A dynamic multicast routing algorithm with inaccurate information satisfying multiple QoS constraints named DMIQ has been proposed. DMIQ uses modified BF algorithm as path search algorithm, WFQ as scheduling mechanism, and it can build a multicast tree with the least hops and costs after satisfying bandwidth, delay, delay jitter and loss constraints while link delays are inaccurate and uniformly distributed. The simulation results show that DMIQ has better performances than the multicast routing algorithms that does not satisfy any QoS.

Key words: inaccurate; dynamic multicast routing; QoS; DMIQ; BF algorithm

1 引言

常用的路由算法, 如 Dijkstra^[1,2] 和 BF^[2-4], 只考虑单 QoS 约束. 多 QoS 约束的组播路由问题是一个 NP 完全问题^[5-7]. 对动态组播路由^[8] 问题, 大多数研究仅考虑单 QoS 约束情况^[9-12]. 实际组播路由还具有不精确性的特征^[13-15]. 研究非精确状态下满足多 QoS 约束的动态组播路由问题在诸多应用领域有着广泛而迫切的需求. 本文从端到端延迟绑定的角度, 基于速率服务模型、BF 算法和 WFQ 策略^[16-19], 提出了一种在非精确状态下满足多 QoS 约束的动态组播路由算法) DMIQ. 理论分析证明该算法具有合适的复杂度, 能在非精确且满足多 QoS 的条件下确定最小跳数和开销的动态组播路由, 仿真实验也证明 DMIQ 算法在不精确状态下表现出良好的路由请求平均成功率、平均延迟和平均跳数性能.

2 网络模型与问题描述

网络用加权有向图 $G(V, E)$ 表示, V 是节点集, E 是节点

间双向链路集. $M < V$ 为组播组, T 为组播树, $P_T(v_s, v_d)$ 表示树 T 中一条从源点 v_s 到目标节点 $v_d \in M - \{v_s\}$ 的路径. 非精确状态下端到端多 QoS 约束动态组播路由问题可描述为, 在满足 QoS 约束条件

$$P \leq D \leq M, D(P_T(s, d)) = \sum_{(u, v) \in P_T(s, d)} d(u, v) [D_{max} \quad (1)$$

$$P \leq P_T(s, d) \leq T, P(P(s, d) | T, F_D(P_T(s, d)) \setminus F_D(P(s, d)) \quad (2)$$

$$J(P_T(s, d)) = \sum_{(u, v) \in P_T(s, d)} j(u, v) [J_{max} \quad (3)$$

$$B(P_T(s, d)) = \min_{(u, v) \in P_T(s, d)} b(u, v) \setminus B_{min} \quad (4)$$

$$P_N(P_T(s, d)) = \prod_{(u, v) \in P_T(s, d)} (1 - l(u, v)) \setminus P_{min} \quad (5)$$

的前提下实现如下优化目标:

$$H(P_T(s, d)) = \sum_{(u, v) \in P_T(s, d)} h(u, v) \rightarrow \min \quad (6)$$

$$C(P_T(s, d)) = \sum_{(u, v) \in P_T(s, d)} c(u, v) \rightarrow \min \quad (7)$$

收稿日期: 20030919; 修回日期: 20040420

基金项目: 国家自然科学基金(No190304006); 高等学校博士学科点专项科研基金(No120020698018)

式中, D_{\max} 为延迟上限, J_{\max} 为抖动上限, B_{\min} 为带宽下限, P_{\min} 为包无损传输概率下限, $F_D(P)$ 为保证路径 P 端到端延迟为 D 的概率, $H(P)$ 为路径 P 的跳数, $C(P)$ 为路径 P 的开销. 若式(6)和(7)发生冲突, 优先考虑(7), 即新节点路径尽量利用已有组播树. 这是因为从源节点到新节点可能有多条满足式(1)~(5)的路径, 优先满足(7)将构建开销最小且满足最大跳数限制的组播树. 相同开销情况下, 选择跳数较小的路径.

3 算法描述与性能分析

3.1 问题简化

DMIQ 基于速率服务模型, 采用单一漏桶约束机制和 WFQ 策略, 对流 q 有

端到端延迟上限为:

$$D_{\max} = \frac{b^q}{\min_{i=1}^K w_i} + \sum_{i=1}^K \frac{L_{\max}^q}{w_i} + \sum_{i=1}^K \frac{L_{\max}^q}{c_i} + \sum_{i=1}^H p_i \quad (8)$$

端到端延迟抖动上限:

$$J_{\max} = (b^q + HL_{\max}^q) / w_q \quad (9)$$

在第 n 次分组转发需要的缓冲空间: $Buf_n^q = b^q + nL_{\max}^q$ (10)

式中, b^q 为流 q 的突发分组数量, w_i 为节点 i 输出链路可用带宽, c_i 为节点 i 输出链路容量, p_i 为节点 i 输出链路传播延迟, H 为路径跳数, K 为路径节点数. 式(8)中排队延迟主要由瓶颈节点决定^[18]. L_{\max}^q 为流 q 中最大分组长度, Buf_n^q 为流 q 在第 n 跳时需要的缓冲大小.

(1) 定义带宽转换函数为

$$P_l = (u, v) \in E, b(l) = \begin{cases} B_{\min} & b(l) < B_{\min} \\ 1/c(l), b(l) \setminus B_{\min} & \end{cases}$$

(2) 由式(8)及假设 $w_1 = w_2 = \dots = w_k = W, c_1 = c_2 = \dots = c_k = C, p_1 = p_2 = \dots = p_k = P$, 可得满足延迟约束的最大路由跳数为 $H_{\max}^{\text{delay}} = \frac{C(WD_{\max} - b^q) - L_{\max}^q(W + C)}{L_{\max}^q(W + C) + PWC} \cdot 8$.

(3) 由式(9)知, 满足延迟抖动约束的最大路由跳数为 $H_{\max}^{\text{ jitter}} = \frac{J_{\max} W q - b^q}{L_{\max}^q} \cdot 8$, 因此, 基于多 QoS 的组播路由最大跳数 $H_{\max} = \min(H_{\max}^{\text{delay}}, H_{\max}^{\text{ jitter}})$.

(4) 由式(10)知, 路由扩展到链路 l 时允许的最大跳数为 $h(l)_{\max} = \frac{Buf(l)^q - b^q}{L_{\max}^q} \cdot 8 - 1$.

(5) 不妨假设链路额定延迟为 $[a, b]$, 链路 l 的延迟 D_l 在 $(d_l, d_l + \$ d_l)$ 上均匀分布, 简单起见, 取 D_{\max}, d_l 和 $\$ d_l$ 为整数值, 且 $\$ d_l = \$ d$. 则 D_{\max} 可分解为本地延迟^[13]:

$$D_l = (D_{\max} - \sum_{i \in P} d_i) \frac{\$ d}{n \$ d} + d_l \quad (11)$$

$$g_l(D_l) = \frac{D_{\max} - \sum_{i \in P} d_i}{n \$ d} \quad (12)$$

$$F_D(P, s, d) = \prod_{i \in P} P_l(s, d) \setminus \left(\frac{D_{\max} - \sum_{i \in P} d_i}{n \$ d} \right)^n \quad (13)$$

式中, $g_l(D_l)$ 表示链路 l 上保证延迟绑定为 D_l 的概率. 继续简化如下:

(6) 假设链路额定延迟为 $[a, b]$, $P_l = (u, v) \in E$, 链路 l 的延迟 D_l 在 $(d_l, d_l + \$ d_l)$ 上均匀分布, 以 $1 - P(D_l \in [a, b])$ 作为链路的权值 $(P(D_l \in [a, b])$ 为 D_l 属于 $[a, b]$ 的概率). 在确定路径后, 再以式(13)验证, 满足则路径寻找成功, 否则失败.

3.2 算法描述

DMIQ 由网络初始化、节点加入和退出三部分组成, 节点加入和退出部分算法如下.

(1) DMIQ 算法节点加入部分

```
Void Join(Mgraph G, ShortPath Table D, int end, Path Matrix P) {
/* 该算法是在 BellmanFord 算法的基础上且有最大跳数 maxhop 的限制下求有向图 G 中从 V_0 顶点到新加入的顶点 end 的最短路径及其带权长度. 若 P[v][w] 为非零值, 则 w 表示到 v 顶点的路径要经过的顶点, 大小表示顺序. */
InitializeMatrix(G, P, D) // 通过拓扑图 G 来初始化 P 和 D
for(i = 1; i < nodenum; ++ i) {
ShortNode(D) // 找到当前距离 V_0 更近的结点
if(v = end) { // 判断 v 结点是否就是要求的 end 结点
for(m = 0; m < nodenum, m + +) { // 加入组播源前判断是否满足条件
if(n < Gainhl() && JJud()) { // n 为路径总跳数, Gainhl() 为已有最大跳数, JJud() 为用式(13)验证确定路径
n - - ;
if(n = 0) { Jack(s, end, P); break; } // 组播源 s 确认 end 申请加入组播树
}
else { JRef(s, end, P); break; } // 组播源 s 拒绝节点 end 申请加入组播树
}
break; // 找到路径跳出循环
}
Refresh(P, D) // 更新当前最短路径及距离
}
}
```

(2) DMIQ 算法节点退出部分

```
Void Remove(Mgraph G, Path Matrix P, int i) {
/* 删除已有的树中的叶子结点, 若 P[v][w] 为非零值, 则 w 表示到 v 顶点的路径要经过的顶点, 大小表示顺序. i 节点表示要求被删除的节点. */
for(v = 0; v < nodenum; ++ v) { // 判断是否该节点是叶子节点
if(p[v][i] != 0 && v != i) { // 不是叶子节点
Suspend(s, f, end) // 组播源 s 通知节点 v 的父节点 f 挂起链路(f, v)
break;
} // if
} // for
if(i != 0) { // 删除该节点
Delete(i, P)
RAck(s, f, P) // 节点 f 通知 s, 组播树已动态修改
} // if
} // delete
```

313 算法性能分析

定理 1 如果存在, DMIQ 性能在非精确状态下构建一棵满足带宽、延迟、延迟抖动和包丢失率约束且具有最小跳数和开销的动态组播树.

证明 DMIQ 先排除不满足带宽约束的链路, 故所确定的组播树满足带宽约束; 将延迟和抖动约束转换为最大路由约束, 且将端到端延迟约束转化为链路开销约束, 在源节点和目的节点之间选取一条满足约束概率最大的路径, 并以式 (13) 验证, 故所确定的组播树满足延迟与延迟抖动约束. 包丢失率约束转换为链路 l 允许的最大跳数, 第 i 次迭代时, 将从第 i-1 跳最短路径中寻找第 i 跳最短路径, 而所有 h(1)max < i 的链路将被剔除, 因此所确定的组播树满足包丢失率约束. 由式 (6) (7)、BF 算法及 3.1 知, DMIQ 构建的组播满足多 QoS 且跳数最小, 在存在多条相同跳数的路径时, 选择开销较小的路径, 故所确定的组播树是最小跳数和开销的. DMIQ 支持节点动态加入与退出, 故 DMIQ 确定的组播树是动态的.

综上, 如果存在, DMIQ 必能在非精确状态下构建一棵满足带宽、延迟、延迟抖动和包丢失率约束且具有最小跳数和开销的动态组播树.

定理 2 DMIQ 算法初始化的计算复杂度为 O(V^2), 计算复杂度为 O(|E|Hmax).

证明 DMIQ 算法主要由初始化、节点加入和退出三部分组成. 其中, 初始化的节点随机生成部分与 DMIQ 无关; 链路权值初始化的计算复杂度为 O(V^2), 故 DMIQ 初始化计算复杂度为 O(V^2). 节点退出部分的计算复杂度为 O(1). DMIQ 路径搜索采用 BF 算法, 故节点加入部分的计算复杂度与 BF 在同一级别, 为 O(|E|Hmax), 则总体计算复杂度为 O(|E|Hmax).

4 仿真实验

实验网络拓扑采用 Waxman 模型[10], 生成公式为:

P({U, V}) = Bexp(-d(u, v)/LA) (14)

式中, d(u, v) 为节点 u 到 v 之间的笛卡尔距离, L 为两节点间最大距离, 本文引入阈值概率 Pthreshold, 若 P({u, v}) \ Pthreshold, 则 (u, v) 为构成仿真网络的一条链路, 令:

d1 = d(u, v) * b / lmax (15)

式中, lmax 为两节点间实际距离上限, b 为链路额定延迟上限. 本文引入非精确参数 Ia, 且:

Ia = S d1 / d1 (16)

节点分布在 (1000, 1000) 的网格中, 节点度为 4 左右, 类似实际情况. 选择路由请求平均成功率、平均延迟和平均跳数作为衡量指标, 其中路由请求平均成功率指从源节点到目的节点建立满足多 QoS 约束路径的平均概率, 平均延迟指确定路

径的平均延迟值, 平均跳数指确定路径平均经历的跳数. 针对音频和普通数据流, 在节点数从 10 递增到 100 且 Ia = 5% 的情况下, 对 DMIQ、Greedy[10] 和简单最小跳数法各进行 100 实验, 结果取平均值. 本文还仿真了节点数为 100 且 Ia 从 5% 递增到 100% 的情况下, 三种算法的路由请求平均成功率的 100 次实验, 结果取平均值. 实验参数[20] 如表 1 所示, 实验结果如

表 1 实验参数

Table with 2 main columns: 网络拓扑参数 and 网络属性参数. Sub-columns include A, B, L, Pthreshold, C, W, P, B, Dmax, Lmax, b^q, Jmax, Buf(1).

图 1- 8 所示.

4.1 路由请求平均成功率

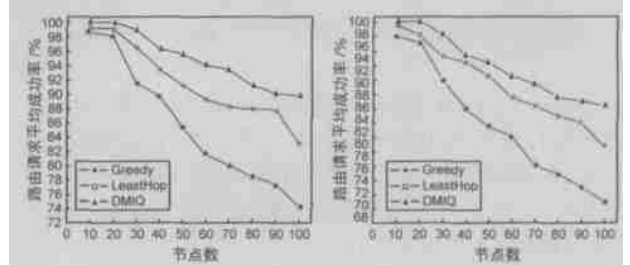


图 1 音频数据流的路由请求成功率

图 2 普通数据流的路由请求成功率

4.2 路由平均延迟

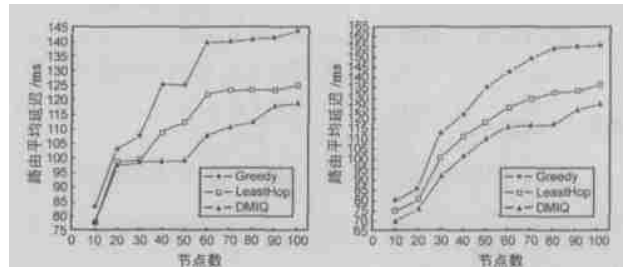


图 3 音频数据流的路由平均延迟

图 4 普通数据流的路由平均延迟

4.3 路由平均跳数

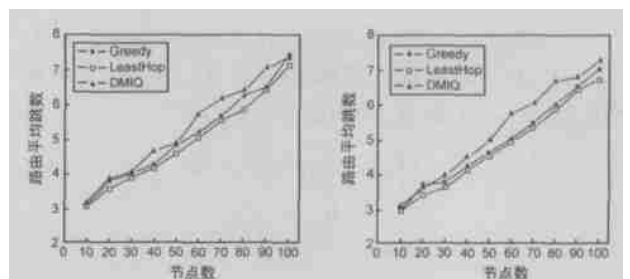


图 5 音频数据流的路由平均跳数

图 6 普通数据流的路由平均跳数

41.4 不同非精确率下的路由请求平均成功率

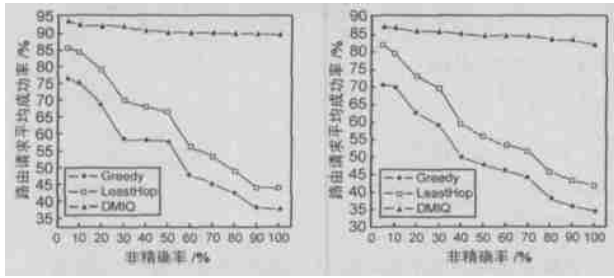


图7 不同精确率下音频数据流的平均成功率

图8 不同精确率下普通数据流的平均成功率

41.5 结论

由实验结果可见, DMIQ 和没有考虑 QoS 约束与非精确率影响的两种动态组播路由算法相比, 当非精确率为 5% 且节点数从 10 递增到 100 时, 路由平均跳数性能相差无几, 但表现出更好的路由搜索和延迟性能. 当节点数为 100 且非精确率从 5% 递增到 100% 时, DMIQ 的路由请求平均成功率表现出明显的优势. 另外, 随着节点数的增加, 分布趋于广泛, 落在矩形网格边角的节点将影响三种算法的衡量指标, 使其随节点数变化而变化. 仿真实验还证明, 在非精确率为 0% 时(精确状态), DMIQ 同样具有以上的性能优势, 因此也适合于小规模精确状态的动态组播应用.

5 小结

本文建立了网络模型, 描述并简化了所讨论的问题, 提出了非精确状态下满足多 QoS 约束的动态组播路由算法) DMIQ, 给出了算法描述与复杂度分析, 理论推导和仿真实验结果表明, 在非精确状态条件下, 对音频和普通数据流, DMIQ 算法具有更好的动态组播路由性能, 同时, 作为非精确状态的特例, DMIQ 算法也适合于确定信息条件下的动态组播应用.

参考文献:

- [1] Dijkstra E. Two problems in connection with graphs[J]. *Numerische Mathematik*, 1959, 1(5): 269- 271.
- [2] Cormen T H, Leiserson C E, Rivest R L. *Introduction to Algorithms* [M]. 2nd ed. Cambridge, USA: MIT Press, 2001.
- [3] Bellman R. *Dynamic Programming* [M]. Princeton, USA: Princeton University Press, 1957.
- [4] Ford L R, Fulkerson D R. *Flows in Networks* [M]. Princeton, USA: Princeton University Press, 1962.
- [5] Wang Z, Crowcroft J. Quality of service routing for supporting multimedia applications[J]. *IEEE J Selected Areas in Communication*, 1996, 14(7): 1228- 1234.
- [6] Xiang F, Junzhou L, Jieyi W, et al. QoS routing based on genetic algorithm[J]. *Computer Communications*, 1999, 22(15- 16): 1392- 1399.
- [7] Israel Cidon, Raphael Rom, Yuval Shavitt. Multipath routing combined with resource reservation[A]. *IEEE INFOCOM 97* [C]. Kobe, Japan, 1997. 92- 100.
- [8] Cristian Huitema. *Routing in the Internet* [M]. Englewood Cliffs, USA:

Prentice Hall, 1995.

- [9] M Doar, I Leslie. How bad is naive multicast routing[A]. *IEEE INFOCOM. 93* [C]. San Francisco, USA: IEEE INFOCOM. 93, 1993. 82- 89.
- [10] B M Waxman. Routing of multiple connections[J]. *IEEE Journal on Selected Areas in Communications*, 1988, 6(9): 1617- 1622.
- [11] Chen S G, Klara N. An overview of quality of service routing for next generation high speed networks: Problems and solutions[J]. *IEEE Net works*, 1998, 12(6): 64- 79.
- [12] Cho J, Breen J. Analysis of the performance of dynamic multicast routing algorithms[J]. *Computer Communications*, 1999, 22(7): 667- 674.
- [13] R Guerin, A Orda. QoS based routing in networks with inaccurate information: Theory and algorithms[A]. *IEEE INFOCOM. 97* [C]. Kobe, Japan, 1997. 75- 83.
- [14] A Orda. Routing with end-to-end QoS guarantees in broadband networks[J]. *IEEE/ACM Transactions on Networking*, 1999, 7(3): 365- 374.
- [15] S Chen, K Nahrstedt. Distributed QoS routing with imprecise state information[A]. *Proceedings of 7th IEEE International Conference on Computer, Communications and Networks (ICCCN 98)* [C]. La Fayette, LA, 1998. 614- 621.
- [16] Parekh A K, Gallager R G. A generalized processor sharing approach to flow control in integrated services networks: The single node case[J]. *IEEE/ACM Transactions on Networking*, 1993, 1(3): 344- 357.
- [17] Keshav S. *An Engineering Approach to Computer Networking* [M]. Reading, USA: Addison Wesley, 1998.
- [18] Stallings W. *High Speed Networks TCP/IP and ATM Design Principles* [M]. Englewood Cliffs, USA: Prentice Hall, 1998.
- [19] Kleinrock L. *Queueing Systems, Volume 1: Computer Applications* [M]. New York, USA: John Wiley & Sons Inc, 1976. 27- 34.
- [20] 赵健, 吴介一, 顾冠群. 一类基于网络服务品质要求的单播路由算法[J]. *通信学报*, 2001, 22(11): 30- 41.

作者简介:



王立男, 1975 年生于湖南湘潭, 1998 年毕业于西安交通大学电子工程系, 获工学学士学位, 2002 年毕业于西安交通大学计算机科学与技术系, 获工学硕士学位, 现为西安交通大学计算机系统结构与网络研究所博士生, 主要研究方向为新一代网络管理、网络行为学.



李增智男, 1938 年生, 教授, 博士生导师, 西安交通大学计算机系统结构与网络研究所所长, 国家 863 计划 CIMS 网络系统主任设计师, 中国计算机学会网络与数据通信专业委员会委员, 在国内外期刊发表论文 200 余篇, 出版专著 6 本, 主要研究方向为计算机网络及应用.