

一类基于源路由的多约束实时组播路由优化控制算法

赵 键¹, 吴介一¹, 罗军舟², 张飒兵²

(1. 东南大学自动控制系, 南京 210096; 2. 东南大学计算机科学与工程系, 南京 210096)

摘 要: 随着 Internet 网络的不断发展, 它将成为未来网络服务的平台, 而未来网络服务的种类又是多种多样的, 既有传统的尽量服务 (Best Effort Service) 应用, 又有对网络服务质量^[1] (QoS: Quality of Service) 有着特殊需求的应用。本文探讨了如何在网络层路由这一层面进行 QoS 组播路由支持的问题。目前, 已有许多组播 QoS 路由算法提出, 在 QoS 路由性能、网络资源使用效率以及路由吞吐量等方面取得了许多好的成果。针对这些 QoS 路由算法中存在的一些问题, 提出了一种组播 QoS 路由算法。实验结果表明: 该算法在适应网络拓扑结构、网络规模、对称/非对称链路负载等方面, 与其它 QoS 路由算法相比, 在算法性能上有了一定的改进。

关键词: 服务品质; 多约束度量; 路径选择; 组播路由

中图分类号: TN393 **文献标识码:** A **文章编号:** 0372-2112 (2001) 04-0490-05

A Source Routing Based Multi-Constrained Real Time Multicast Routing Control Algorithm

ZHAO Jian¹, WU Jie yi¹, LOU Jun zhou², ZHANG Sa bing²

(1. Department of Automatic Control, Southeast University, Nanjing 210096, China;

2. Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: With the rapid growth and development of Internet, it will become the ultimate platform which provides all kinds of network services. Among these network services based on the Internet platform, there are not only best effort services, but also quality guaranteed services^[1]. In this article, we discuss the issues on how to support quality services of multicast communications on the network layer by way of QoS (Quality of Service) routing strategy. Currently, there have been many multicast QoS routing algorithms proposed in literature. All these algorithms achieved lots of exciting results in network resource utilization and throughput, but still have some problems to be improved. Based on these problems, we proposed our QoS routing algorithm, and the experiment shows that compared to the other proposed algorithms, our algorithm has many improvements on adaption to network topology, network size, symmetric and asymmetric link load.

Key words: quality of service; multi constraints metric; path selection; multicast routing

1 引言

随着互联网络规模、链路传输带宽及其可靠性的不断提高, 给一些关键应用 (如实时多媒体视频点播 VOD、远程分布控制等) 在高速、实时网络环境下的实现奠定了基础。由于视频、音频等信息自身的特点以及网络的传输特性, 必须为其提供端到端的实时 QoS 服务品质的保证, 以满足端用户对服务品质的要求。具体地讲, 一方面, 端用户的操作系统应提供实时调度机制, 而在联网的设备 (路由器、交换机) 之间提供调度和路由选择的支持。因此, QoS 品质就是指端主机资源和网络资源等服务参数, 如 CPU 工作周期、网络带宽、传输延迟、抖

动和链路租用代价等。从网络的角度上来看, 提供 QoS 保障的途径有多种, 其中, 基于网络层的路由选择就是一种可行的解决方法。QoS 路由就是在两个端点 (指单播) 或点对多点 (指组播) 之间找出一条满足 QoS 参数要求的路径。本文探讨了组播条件下多个 QoS 参数约束的路由选择问题。

针对组播 QoS 多约束路由问题, 研究人员已经提出了许多路由算法。其中, 以 Zhu 等的 BSMA 算法^[2]、Kompella 等的 KPP DMST 算法^[3, 4]和 Salama 等的 DCMST 算法^[5]为典型代表。BSMA (Bounded Shortest Multicast Algorithm) 算法是最有效率的延迟约束最小 Steiner 树启发式探索算法, 但其计算复杂度偏高;

DMST(Delay constraint Minimum Steiner Tree Heuristics)作为分布延迟约束最小 Steiner 树启发试探算法, 它的路由计算速度最快。但是, 它是针对网络链路负载是对称的情况而提出的, 因此, 若应用在非对称链路负载的网络环境下, 其效率是很低的; DCMST(Delay constraint Minimum Spanning Tree Heuristics)则适合于对称和非对称链路负载的网络环境, 其效率与 BSMA 基本相同, 但是, 随着网络规模的扩大, 其效率略显不足。BSMA 算法、DMST 算法和 DCMST 算法的计算复杂度分别为 $O(kn^3 \log n)$ 、 $O(n^3 \Delta)$ 、 $O(n^3 \Delta)$ 。

本文基于以上三种有代表性的组播路由算法存在的不足, 提出了“破圈回滚”路由算法, 文章组织结构如下: 第 2 部分对组播路由进行了分类, 指出了现有组播路由实施过程所存在的一些问题; 第 3 部分对网络进行了形式化定义, 指出了满足多约束条件的组播多约束 QoS 路由问题是一个 NP 完全问题; 第 4 部分详细描述了我们所提出的“破圈回滚试探” CRMA 算法, 并给出了相应的例子; 第 5 部分给出了实验仿真结果; 第 6 部分对本文进行了总结。

2 集中式与分布式组播路由的比较

本文采用了基于源路由的路由选择策略, 目前路由研究的方法^[6]基本是围绕着源路由和分布路由来进行的, 它们各有优缺点。

2.1 集中式源路由

源路由采用了集中处理的方式, 因此在发出路由请求时, 可以立即获取预先(proactive)存在的路由信息, 无需在各个网络节点之间进行信息交换。但是, 它需要在网络中的每个节点上都维持一张有关整个网络全局链路状态的“图”。另外, 经过源路由计算的路径是不含回路的。总之, 源路由是一种简单、明了、易于实施、评估、检查和升级维护的路由计算方法。与设计解决属于 NP 完全问题的分布路由启发试探算法相比, 其实施是很容易的。但基于源路由的路由发现算法也存在一些问题, 即: (1) 为维持整个网络通信链路的状态信息必须进行高频率的状态刷新, 给网络造成了负载; (2) 由于不可避免地存在着刷新延迟, 因此存在着状态信息滞后的问题。

2.2 分布式路由启发试探算法

在分布路由中, 路径计算是激励式的, 即当有路由请求时, 它不会马上获取到路由信息, 需要一个路径搜索延迟。但是, 该类路由选择和优化都是基于局部状态信息的, 不需要象源路由那样为网络所有节点都维持一张链路状态表。因此分布路由算法能够更好地适应网络规模, 但计算复杂度比较高。同时, 在多个方向上并行搜索多个可行路径, 增加了搜索的成功率。此外, 当链路状态信息由于广播延迟而造成不同节点的有关链路状态不一致时, 回路就会发生, 从而增加了网络带宽的消耗。

3 网络组播树模型

网络组播问题可以用树的结构来描述, 假定网络中的某个源节点 $S \in V$ 将信息传送给一组节点 $D = \{d_1, d_2, \dots, d_n\} \subseteq V$ 。在源节点 S 发出数据之前, 必须首先确定出多条从 S 到

D 的路径, 即寻找这样一棵树 $T = \langle V_T, E_T \rangle$ ($V_T \subseteq V, E_T \subseteq E$): 该树的根在源节点 S , 而叶节点则是组播对象 D , T 中允许存在充当中继作用的非组播节点, 但是这些中继节点不能是 T 中的叶节点。对于 T 中的一条路径 $P(S, d_i) \subseteq E_T$, 定义该条路径的总延迟以及组播树 T 的总代价分别为 $\sum_{\ell \in P(S, d_i)} D(\ell)$, $\sum_{\ell \in E_T} C(\ell)$ 。在构造上述组播树 T 的过程中, 通常需要考虑两方面的因素: 第一是用户提出 QoS 的服务品质要求, 如带宽、延迟、链路代价等; 第二是网络资源的总体使用效率。

基于上述分析, 给出问题的形式化描述为: 对于给定的网络图 $G = \langle V, E \rangle$, 其中 V 是网络节点的集合, E 是网络通信链路的集合, 组播源节点 $S \in V$, 组播目的节点集合 $D = \{d_1, d_2, \dots, d_n\} \subseteq V$, 带宽函数 $B: E \rightarrow R^+$, 延迟函数 $D: E \rightarrow R^+$, 链路代价函数 $C: E \rightarrow R^+$, QoS 约束条件 $\Delta_{delay}, \Delta_{bandwidth}$, 构造满足以下条件的组播树 $T = \langle V_T, E_T \rangle$ ($V_T \subseteq V, E_T \subseteq E$):

$$\begin{cases} B(\ell) \geq \Delta_{bandwidth}, \forall \ell \in P(S^+ D_i) \cap \forall D_i \in D, i = 1, 2, \dots, n; \\ \sum_{\ell \in P(S^+ D_i)} D(\ell) \leq \Delta_{delay}, \forall D_i \in D, i = 1, 2, \dots, n; \\ \text{minimize } \sum_{\ell \in E_T} C(\ell); \end{cases}$$

4 组播 QoS 路由算法

已提出的组播路由算法(见第 1 部分)基本上是围绕着延迟约束、最小代价两个问题展开的, 而未对网络带宽进行相应的讨论。但是, 在实际的网络运作过程中, 带宽也是一个不可忽视的因素, 它对其它两个 QoS 约束参数(延迟、代价)有着非常直接的影响。一般情况下, 带宽越高, 延迟越小, 但是代价可能会越高。因此, 针对过去研究中已提出的组播路由算法, 文中将带宽约束参数引入进来, 以使路由算法更能够适应网络环境的变化。另外, 网络组播应用是实时的, 因此 QoS 路由的研究既要考虑到如何有效地利用网络资源, 又要考虑满足路由计算的实时要求。基于上述考虑, 提出了“破圈回滚”组播路由法, 破圈法最早是由数学家 Rosenstiehl 和管梅谷(GUAN Meigu)分别于 1967 年、1975 年提出的, 用于解决最小代价树的生成问题。破圈法操作简洁, 非常适合于基于源路由的操作思想与环境。在实际网络设备以及仿真实验中, 网络节点间的信息是以邻接矩阵的形式进行存储的, 而应用破圈的思想, 可以非常快地确定出“圈”的存在而予以打破, 进而回避了路由生成过程中“环”发生的可能。随着网络节点数的增多, 它的优势更明显。

第 5 部分的仿真对比实验说明: 这里所提出的 CRMA 算法比较均衡地考虑到了组播树代价和延迟、带宽等约束因素, 把用于生成最小 Steiner 树的算法和用于生成最短路径的 Dijkstra 算法结合起来, 从而获得了一棵确保组播带宽的“最佳平衡”组播树。另外, CRMA 算法的优势还在于它比较适合于构造组播目的节点较多的环境中, 此时其效率可以得到最好体现。与 CRMA 算法相比, BSMA、DMST 和 DCMST 则在满足延迟约束的前提下, 更侧重于组播树的代价开销最优, 它们用所获得的最优代价去弥补路径延迟和执行速度上的“衰减”。

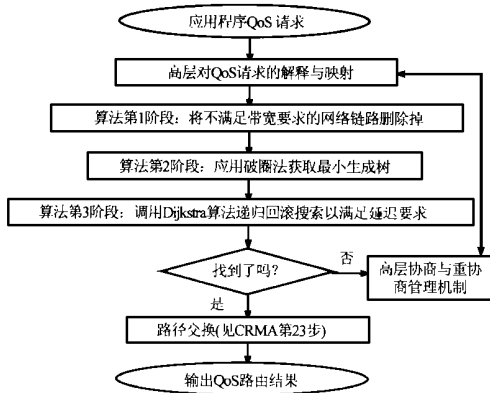


图1 CRMA 算法流程图

4.1 破圈回滚组播路由算法描述

算法分为三个阶段(见图1的算法流程图),它由带宽约束满足筛选、最小代价树生成以及延迟约束满足等三个阶段组成,伪代码实施如下:

4.1.1 算法输入定义 破圈回滚组播路由算法 CRMA(Cycle breaking Rollback Multicast Algorithm)的输入变量给定如下,即有算法 CRMA($G, S, D, \Delta_{bandwidth}, \Delta_{delay}$)的导入,其中:

$G = \langle V, E \rangle$, $link = (bandwidth_{ij}, delay_{ij}, cost_{ij})$;
 S 为源节点, D 为组播目的节点集合, $D = \{d_1, d_2, \dots, d_n\}$;
 $\Delta_{bandwidth}$ 为组播带宽约束, Δ_{delay} 为组播延迟约束。

4.1.2 带宽满足阶段(算法第一阶段) 该阶段主要用于删除不满足带宽约束要求的边 ℓ , 算法描述如下:

```

(1) for  $\forall i \in V$  do
(2)   for  $\forall j \in V(i \neq j)$  do
(3)     if  $bandwidth_{ij} < \Delta_{bandwidth}$  then  $bandwidth_{ij} = \infty$ 
(4)     else 进入第二阶段

```

4.1.3 最小生成树的产生(算法第二阶段) 运用破圈法处理网络图,即以每个节点为核心,考察是否存在关于该节点的圈,若有圈存在,则将该圈中最大代价的边删除掉,算法描述如下:

```

(5)  $G_0 := G; k := 0;$ 
(6) for  $\forall i \in V$  do
(7)   if  $G_k$  不含圈 cycle then goto (15)
(8)   else begin
(9)     for  $\forall \ell \in E(cycle)$  do  $\{ \ell_k \in E(cycle): C(\ell_k) := \max(C(\ell)) \}$ ;
(10)     $G_{k+1} := G_k - \ell_k;$ 
(11)     $k := k + 1;$ 
(12)  end begin
(13)  end if
(14) end for

```

(获得了最小生成树 MST, 进入第三阶段)

4.1.4 延迟满足阶段(算法第三阶段) 调用 Dijkstra 最短距离算法,判断上面生成的最小树中的每一条 $S \rightarrow D$ 的路径是否满足延迟约束 Δ_{delay} 的要求。

```

(15) for  $\forall d_i \in D$  do begin

```

```

(16)  $relay\_node := d_i$ ; //  $relay\_node$  表示在回滚法中,当前处于活动工作状态的中继节点,该节点可能是组播目的节点,也可能是非组播节点;

```

```

(17) if  $delay(S \rightarrow d_i) > \Delta_{delay}$  then begin //  $delay(X \rightarrow Y)$ : 表示节点  $X$  到  $Y$  的延迟;

```

```

(18)  $relay\_node := pre(relay\_node)$ ; //  $pre(X)$ : 表示节点  $X$  的前驱节点;

```

```

(19) while  $(S \neq pre(d_i))$  do

```

```

(20)   if  $delay(S \rightarrow relay\_node) + Dijkstra(relay\_node, d_i) < \Delta_{delay}$ 

```

```

(21)   then begin // 调用贪心算法  $Greedy(path())$ , 当有多个最短路径满足延迟要求时,就以可用带宽作为第二选择标准(即把所有满足延迟条件路径的最小带宽按从高到底的顺序排列,取最大的);同样当有多个路径满足第二标准时,就以代价最小作为最终判别的标准;

```

```

(22)    $OPT\_PATH := Greedy(path(Dijkstra(relay\_node \rightarrow d_i)))$ ;

```

```

(23)    $path(pre(d_i) \rightarrow d_i) \xrightarrow{\text{置换成}} path(OPT\_PATH)$ ;

```

```

(24)    $deletepath(pre(d_i) \rightarrow d_i)$ ; // 将该边从原始路径中删除

```

```

(25)   end begin

```

```

(26) end if

```

```

(27) end while

```

```

(28) if  $Dijkstra(S \rightarrow d_i) > \Delta_{delay}$  then begin

```

```

(29)   返回一棵从  $S$  到  $d_i$  最小延迟路径,作为协商/重协商的参考点;

```

```

(30)   调用高层协商/重协商管理机制,进行:

```

```

    ①请求用户降低延迟约束要求;或②重新调用 CRMA;

```

```

(31)   end begin

```

```

(32) else begin

```

```

(33)    $addpath(Dijkstra(S \rightarrow d_i))$ ; // 将边加入到组播树中

```

```

(34)    $deletepath(pre(d_i) \rightarrow d_i)$ ;

```

```

(35)   end begin

```

```

(36) end if

```

```

(37) end begin

```

```

(38) end if

```

```

(39) end for

```

```

(40) 将所有非组播目的叶节点及其相连的边删除掉;

```

```

(41) return 组播树  $T = \langle V_T, E_T \rangle$ ;

```

(CRMA 算法结束)

该算法的时间复杂度是多项式时间,由以上三个阶段的复杂度组成,即 $O(|V|^2) + O(|V|^3) + O(|D||V|^3) = O(|D||V|^3)$, 其中 $|D|$ 为组播目的节点的个数.由此可见,CRMA 算法在计算复杂度上优于 BSMA,同时与 DMST 和 DCMST 算法的复杂度相当。

5 实验仿真

在 PG 466 计算机 Turbolinux 4.0 操作系统环境下,Java 编

程实现了网络图的手动和自动生成以及相关算法。网络节点的生成采用 B. Waxman 模型^[7], 在计算机屏幕 800×600 的空间内随机地产生, 模拟带宽请求为 2Mbps, 延迟约束设定为 0.08 秒。仿真以 China DDN 网络拓扑结构为基础, 为了验证所提算法对网络拓扑结构的适应性, 还引入了另外一种网络拓扑结构, 即网格状拓扑结构。在实验中, 把 CRMA 算法与 BSMA、DMST KPP、DCMST 算法在网络规模、网络拓扑结构、系统执行开销代价、对称与非对称网络负载等方面进行对比, 得到了相应的实验结果, 见图 2~ 9。围绕着 DCMST、DMST、BSMA 以及文中所提出的 CRMA 算法, 就网络规模、网络拓扑结构和对称/非对称链路负载等因素进行了仿真实验, 同时也给出了评价组播树总代价、系统执行时间、平均延迟和平均路径长度等实验指标。目前的网络拓扑结构是以 Internet 的 ISP 型 (Internet Service Provider) 拓扑结构占主导地位的, 因此, 图 4~ 9 的数据是围绕着上述 ISP 拓扑结构展开的, 这样可更具代表性。

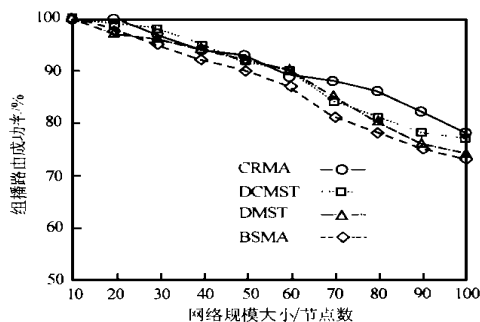


图2 网络拓扑结构: ISP internet 型组播节点数: 网络规模* 10%

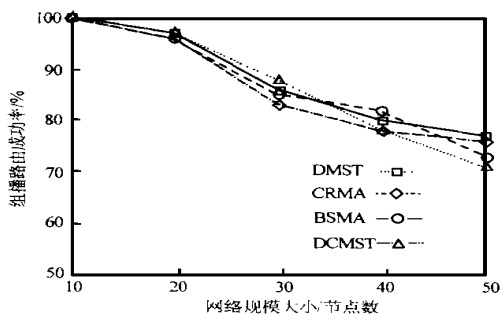


图3 网络拓扑结构: mesh 网格状组播节点数: 网络规模* 20%

图2和图3给出了 DCMST、DMST、BSMA 和 CRMA 等算法在两类不同网络拓扑结构下的实验结果。其中, 一类是目前广泛采用的 INTERNET 服务网络拓扑结构, 而另一类则是具有 mesh 网格状的拓扑结构。图2表明上述四种算法在不同的网络规模下, 随着网络节点数的增加, 其路由性能逐渐下降。但由 100 个节点组成的网络中, 其组播路由成功率依然可以维持在 75% 以上。图2还表明所设计的路由算法的路由成功率与其它三种路由算法相比, 当网络规模超过 60 时 (此时, 组播节点数为 $60 \times 10\% = 6$), 其成功率总体上讲, 是略高于其它路由算法的, 这得益于其简单明了的路由实现算法。

图3给出了网格状拓扑结构下的实验结果。从图中可以看出, 四种算法的路由成功率比较接近, 不分伯仲。当网格关节点数从 40 到 50 时, DMST 与 BSMA 算法较 CRMA 和 DCMST 成功率来得要高, 这是由于 DMST 和 BSMA 均采用从上到下的启发式探方法, 随着网络节点数的增加, 由于可选路径增加了, 因此其路径搜索的成功率也会随之增加。而 CRMA 算法和 DCMST 算法随着关节点数的增加, 其路由成功率略低于 DMST 和 BSMA 算法。

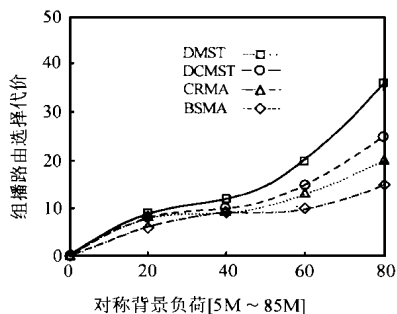


图4 网络由 20 外节点组成, 其中组播节点由 10 个节点组成

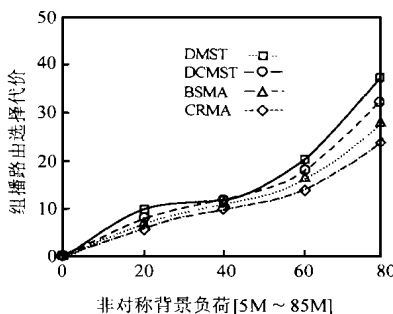


图5 网络由 20 个节点组成, 其中组播节点由 10 个组成

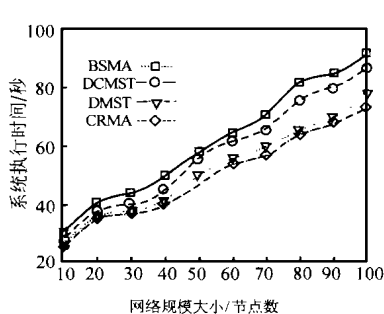


图6 网络拓扑结构: internet ISP 型组播节点数: 网络规模* 10%

图4和图5给出了四种算法在对称/非对称网络负载环境下组播路由代价的实验结果。图4和图5的网络容量假定为 100M, 其背景流量为 5M 至 85M, 从图4和图5可以看出四种算法均可应用在对称/非对称网络负载环境下, 而我们提出的 CRMA 算法, 在非对称环境下仅次于 BSMA, 而在对称环境下则次于 DCMST。

图6给出了系统执行四种算法的时间比较。可以看出, 随着节点数的增加, 四种算法的执行时间均呈上升趋势。在网络规模大小为 80 时, DCMST“骤增”, 而 CRMA 则有“下降”趋势。

这是因为随着网络节点数的增加, DCMST 要进行更多的回环检测和代价优化工作, 而相比之下, CRMA 采用了“破圈”法来优化代价, 避免了回环检测等处理开销, 随着节点数的增多, 这种趋势则更为明显。

图7表明, 用四种算法所生成的组播树中, 代价最大的是 DCMST, 以后依次是 BSMA、CRMA 和 DMST。图8表明了, 在延迟约束情况下, 它们所生成的组播树平均延迟比较接近, 这是四种算法都以路径满足延迟约束作为路由选择的最高标准, 这是造成其差别不明显的主要原因。图9给出了用四种算法所

生成的组播树平均路径长度的数值结果, BSMA 和 DCMST 的最优路径长度明显高于 DMST 和 CRMA 生成的路径长度, BSMA 和 DCMST 算法比较侧重于代价开销, 而我们所提出的

CRMA 使用了 *Greedy* 贪心算法和回滚 *Dijkstra* 算法来优化路径. 因此, 结果就是在四种算法所生成的组播树中, CRMA 的平均路径长度是最短的.

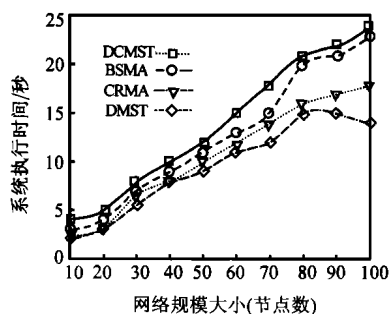


图 7 网络拓扑结构: internet-ISP 型
组播节点数: 网络规模* 10%

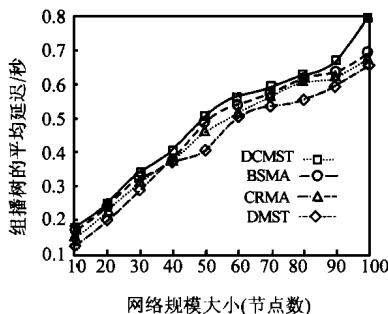


图 8 网络拓扑结构: internet-ISP 型
组播节点数: 网络规模* 10%

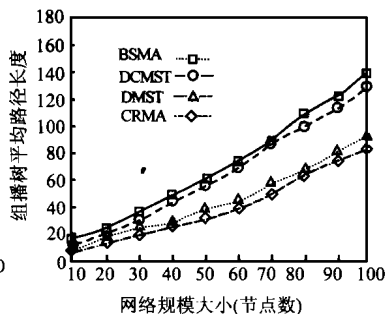


图 9 网络拓扑结构: internet-ISP 型
组播节点数: 网络规模* 10%

6 结论

本文在总结已有研究成果的基础上, 提出了“破圈回滚试探”路由算法, 大量的实验结果表明: 所提出的路由算法, 可以比较好地适应网络规模、网络拓扑结构和对称/非对称网络负载, 通过与其它三种有代表性的组播路由算法的实验比较, 可以认为: CRMA 算法实现比较简单, 且效率较高, 可应用在实时组播通信环境中.

参考文献:

- [1] Z. Wang and J. Crowcroft. Quality of service routing for supporting multimedia applications [J]. IEEE Journal on Selected Areas in Communications (JSAC), September 1996, 14(7): 1228-1234.
- [2] Q. Zhu, M. Parsa and J. Garcia Luna Aceves. A source based algorithm for delay-constrained minimum cost multicasting [J]. in Proceedings of IEEE INFOCOM 95, 1995: 377-385.
- [3] V. Kompella, J. Pasquale and G. Polyzos. Multicast routing for multimedia communication [J]. IEEE/ACM Transactions on Networking, 1993, 1(3): 286-292.
- [4] V. Kompella, J. Pasquale and G. Polyzos. Two distributed algorithms for the constrained steiner tree problem [J]. in Proceedings of the Second International Conference on Computer Communications and Networking (IC3N'93), 1993: 343-349.
- [5] H. Salama, D. Reeves, Y. Viniotis and T.-L. Sheu. Comparison of multicast routing algorithms for highspeed networks [A]. TR-29-1930 [C], IBM, September 1994.
- [6] D. Bertsekas and R. Gallager. Data Networks [M]. Prentice-Hall, 2nd ed., 1992.

- [7] B. Waxman. Routing of multipoint connections [J]. IEEE Journal Selected Areas on Communications, December 1988, 6(9): 1617-1622.

作者简介:



赵 键 1969 年生, 东南大学博士生, IBM 网络系统工程师, 现从事国家 973 信息技术和高性能软件项目, 主要研究方向: IP QoS 路由, 网络管理, 无线 AD HOC 路由, 主动网络理论与技术等.



吴介一 1941 年生, 东南大学教授, 博士生导师, CIMS/ERP 专家, 国家 973 信息技术和高性能软件项目负责人, 主要研究方向: IP QoS 路由, 网络管理, CORBA 分布计算等.

罗军舟 1960 年生, 东南大学博士, 教授, 博士生导师, 国家 973 信息技术和高性能软件项目负责人, 主要研究方向: Petri 理论, 网络体系结构, QoS 适应技术等.

张颀兵 1964 年生, 东南大学, 工程师, 主要研究方向: CSCW 技术, 工作流计算, CIMS 应用.