

# Overlay 组播网络中的服务组合算法

潘 耘<sup>1</sup>, 余镇危<sup>2</sup>, 王银燕<sup>2</sup>

(1. 中国传媒大学计算机与软件学院, 北京 100024; 2. 中国矿业大学(北京)计算机系, 北京 100083)

**摘要:** 在 Overlay 组播路由中既需要考虑确保数据流能获得它所需要的服务, 还需要确保不同的服务按照合适的次序到达, 这是一个新的值得研究的问题, 称之为服务组合问题。该文研究了 Overlay 组播网络中的服务组合问题, 建立了相应的优化模型, 设计了解决该模型的启发式算法。大量的仿真表明了该模型和算法的有效性。

**关键词:** Overlay 组播网络; 服务组合; 扩展最短路径算法

## Service Composition Algorithm on Overlay Multicast Network

PAN Yun<sup>1</sup>, YU Zhenwei<sup>2</sup>, WANG Yinyan<sup>2</sup>

(1. Computer Science & Software Institute, Communication University of China, Beijing 100024;

2. Department of Computer, China University of Mining and Technology (Beijing), Beijing 100083)

**【Abstract】** This paper investigates the service combination problem in overlay multicasting network by taking both the service assuring and order of service into consideration. It presents an optimization model and designs a heuristic algorithm on the service combination problem. Large amount of simulations suggest the model and the algorithm are practical and effective.

**【Key words】** Overlay multicast network; Service composition; Extend short path algorithm

### 1 概述

许多Internet的应用, 如视频会议、远程教学等, 都需要基础网络层对组播通信的支持。虽然基于IP的组播技术<sup>[1]</sup>提出已经十多年了, 但许多问题仍未解决<sup>[2]</sup>: (1)每个分组所需要的状态维护问题; (2)安全问题; (3)扩展的地址分配问题; (4)可靠性、拥塞控制、流量控制问题; (5)部署效率问题, 等等。因此, 目前只有为数不多的ISP能够提供商业的IP组播服务。

根据端到端的原则<sup>[3]</sup>, 应该尽力把功能的实现推向高层, 除非在底层实现这一功能时, 所能够获得的性能效益远远超出了底层实现所带来的额外的复杂性代价。最近, 在这一概念的影响下, 许多研究者把组播研究的眼光放在了应用层<sup>[4,5]</sup>。在应用层实现组播技术, 所有的组播功能像成员管理、数据复制等都交给应用层节点来完成, 这个方法无须当前Internet框架结构的改变, 很容易部署, 还有许多优点, 如适应性强、鲁棒性、客户化能力强等, 因此很容易解决IP组播面临的问题。

如何在应用层实现组播, 目前有两种被广泛接受的方案: 一种是由终端节点组成的虚拟网络, 在其上建立组播路由, 这种方案称为应用层组播<sup>[4]</sup>; 另一种是在网络中间使用了一定的结构(称为代理, 可以实现一定的组播功能), 虚拟网络由终端节点与代理节点共同构成, 这种方案称为Overlay组播<sup>[5]</sup>。文献[6]对比了IP层组播、应用层组播和Overlay组播的性能(如图1所示), 得出了这样的结论: (1)Overlay组播的性能可以比得上IP组播的性能; (2)相比应用层组播, Overlay组播是大规模组通信的一个很好选择; (3)应用层组播善于解决立即需要配置的问题, 而Overlay组播是一个较长的解决方案, 它值得进一步研究。

Overlay 组播不仅性能好、支持长期大规模的组播应用,

而且可把一些通用的新业务添加在代理上, 不但让代理完成数据的转发和复制, 还可以进行一些特殊的服务, 使得从源端到目的端的数据流可得到这些特殊服务, 这样使得 Overlay 组播的应用前景更加广阔, 更值得进一步研究和推广。

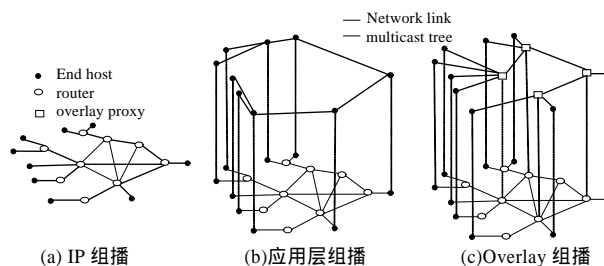


图1 3种组播实现功能的比较

要进行带服务的路由, 也引出了新的问题。服务可以多种多样, 而每组数据流对服务有不同的要求, 接收这些服务需要遵守一定的次序, 例如先加密, 后解密, 而不是相反。在 Overlay 组播路由中需要考虑既要保证数据流能获得它所需要的服务, 还要保证能按照既定的服务次序获得服务, 这是一个新的值得研究的问题, 称之为服务组合问题。可以证明, 这也是一个带约束的 Steiner 树问题, 因此是 NP-hard 的。

本文首先提出了Overlay组播网络中服务组合路由模型; 在此基础上提出了一个求解该问题的启发式算法, 该算法扩展了Floyds最短路径算法, 使得其适用于Overlay组播网络的特殊性, 该算法的复杂度为 $O(n^3)$ ; 并对该算法进行了分析和模拟, 效果较好。

**基金项目:** 教育部博士点基金资助项目(20030290003)

**作者简介:** 潘 耘(1974-), 女, 博士、副教授, 主研方向: 网络性能评价, Overlay 网络; 余镇危, 教授、博导; 王银燕, 博士生

**收稿日期:** 2006-11-15 **E-mail:** panyun13@126.com

## 2 Overlay 组播网络中的服务组合路由模型

Overlay组播网络可以表示为：设网络 $G(V,E,P)$ ,  $V$ 是节点(表示终端节点和代理节点的组合)的集合,  $E$ 是边(表示节点之间的虚拟连接)的集合, 边权表示节点之间的组播代价, 边代价函数 $c_1(e)$ , 其中 $e \in E$ ;  $P$ 是代理节点的集合, 代理节点还提供了一些服务, 为了简化模型, 约定每一个代理节点仅承担一种服务, 当某个代理节点承担 $m(m \geq 2)$ 种服务时, 可简单地将此节点看作是由 $m$ 个节点组成的完全图, 并且设定此完全图中的每条边的代价为0, 每个节点仅仅承担这 $m$ 种服务的一种。为了完成这些服务, 代理节点还要付出一定的代价, 如对CPU的占用, 用 $c_2(p)$ , 其中 $p \in P$ 。假设 $P$ 共能提供 $K$ 类服务。

给定组播源点 $s \in V - P$ , 组播终点集 $D \subseteq V - P$ , 以及本次组播路由的服务需求 $\langle k, L \rangle$  (即需要 $k(k \leq K)$ 类服务, 并按次序约束关系 $L$ 对组播会话进行服务,  $L = \{l_1, l_2, \dots, l_k\}$ , 其中 $1 \leq l_i \leq k, 1 \leq i \leq k$ )。对 $s$ 要求一棵满足服务需求 $\langle k, L \rangle$ 的代价最小的组播树 $T \subseteq G$ , 使得:  $T$ 以 $s$ 为根节点, 且 $D \subseteq V_T$ , 这里 $V_T$ 表示树 $T$ 的节点集合, 这时组播代价由边代价和中间处理节点的代价来决定。模型形式化如下:

给定网络 $G(V,E,P)$ , 组播源点 $s \in V - P$ , 组播终点集 $D \subseteq V - P$ ,  $\langle k, L \rangle$ 为需满足的服务要求。求解服务组合的组播路由问题相当于求解如下约束规划问题:

$$\min_{T \subseteq G} \sum_{e \in E_T} c_1(e) + \sum_{p \in P_T} c_2(p) \quad (1)$$

$$s.t. \text{ path}_T(s,d) \text{ 满足服务需求 } \langle k, L \rangle, \forall d \in D$$

其中:  $T$ 为 $G$ 的子树;  $\text{path}_T(s,d)$ 为 $T$ 中从 $s$ 到 $d$ 的唯一路径;  $E_T$ 为树 $T$ 的边集合;  $P_T$ 为树 $T$ 中提供服务的代理节点的集合。

## 3 模型求解方法的讨论

### 3.1 模型的求解算法

求一棵能满足服务需求的组播树, 传统的启发式算法不能直接应用。必须进行一定的改进, 使其满足服务需求, 为此扩展了最短路径算法Floyd。用 $P^{(i)}$ 来表示能承担第 $i$ 类服务的代理节点集合, 能完成服务需求 $\langle k, L \rangle$ 的代理节点, 可以表示成:  $(P^{(l_1)}, P^{(l_2)}, \dots, P^{(l_k)})$ 。具体算法描述如下:

(1)以 $s$ 为根,  $P^{(l_1)}$ 中的节点为叶子, 在 $V - D$ 中求解一棵树 $T_1$ 。树的边由 $s$ 到 $P^{(l_1)}$ 中的节点的最短路径组成。并在叶子节点处标记根到叶子的代价与叶子节点的处理代价之和;

(2)以 $T_1$ 树中的每个叶子节点为根(即集合 $P^{(l_1)}$ 中的节点), 在集合 $V - D - \{s\}$ 中, 以 $P^{(l_2)}$ 中的节点为目的节点扩展树 $T_1$ 为 $T_2$ , 边集合的构成: 先从 $P^{(l_1)}$ 中每个节点到 $P^{(l_2)}$ 中每个节点的最短距离和 $P^{(l_1)}$ 中节点处所标记的代价之中找出最短路径, 然后将最短路径上的边加入树 $T_2$ 中。同时在每个叶子节点处标记根节点处的代价(即旧值)与根到叶子节点之间的代价以及叶子节点的处理代价之和;

(3)对树 $T_2$ 进行剪枝, 剪去不属于 $P^{(l_2)}$ 的叶子节点;

(4)按服务次序 $L$ 重复(2), 直到生成树 $T_k$ ;

(5)以 $T_k$ 树中的每个叶子节点(即集合 $P^{(l_k)}$ 中的节点)为根, 在集合 $V - \{s\}$ 中, 以 $D$ 为目的节点集合将树 $T_k$ 扩展为 $T$ , 边集合的构成: 先从 $P^{(l_k)}$ 中每个节点到 $D$ 中每个节点的最短距离和 $P^{(l_k)}$ 中节点处所标记的代价之中找出最短路径, 然后将最短路径上的边加入树 $T_k$ 中;

(6)对树 $T$ 进行剪枝, 剪去不在组播终点集中的叶子节点;

(7)现在从整个树 $T$ 角度看, 得到一棵能满足服务需求

$\langle k, L \rangle$ 的次优树。

图2中给出了有满足两类服务, 服务次序为先1后2的组播树的生成过程, 假定源节点 $s$ 为节点1, 目的点集 $D = \{4, 9, 10\}$ 。提供两类服务,  $P^{(1)} = \{2, 5\}$ ,  $P^{(2)} = \{3, 6, 8\}$ ,  $k = \{1, 2\}$ ,  $L = \langle 1, 2 \rangle$ 。

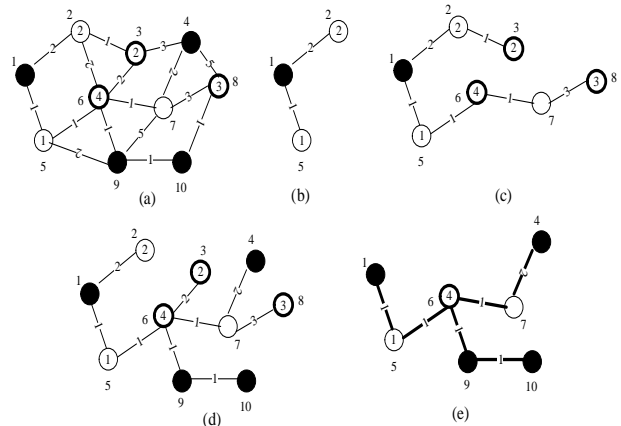


图2 组播树的生成过程

在源点到目的节点的路径上完全有可能存在代理节点没有承担服务, 只作为普通组播节点存在。为了区别, 用 $p/1$ 表示承担服务,  $p/0$ 表示不承担服务, 其中 $p \in P_i$ 。图3给出一个例子来说明这种情况, 假定源节点 $s$ 为节点1, 目的节点集 $D = \{4, 11, 13\}$ 。提供两类服务 $P^{(1)} = \{2, 8, 9, 10\}$ ,  $P^{(2)} = \{3, 5, 12\}$ 。在源点到目的11的路径为: 1, 6, 7, 8/0, 9/1, 12/1, 11, 其中节点8不承担服务, 9、12承担服务。

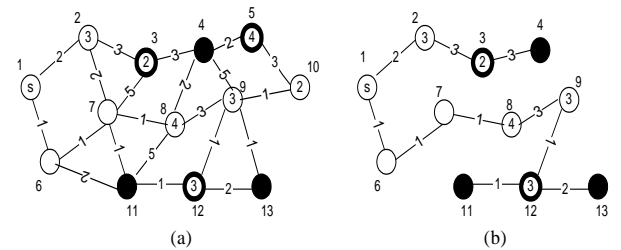


图3 一个特殊的例子

### 3.2 算法的复杂性分析

根据3.1节所描述的算法, 逐步分析其计算复杂性。开始时调用Floyds算法计算任意两个节点之间的最短路径, 而Floyds算法的复杂性为 $O(n^3)$ 。步骤(1)的复杂度为 $O(|P^{(l_1)}|)$ ; 步骤(2)的复杂度为 $O(|P^{(l_1)}| |P^{(l_2)}|)$ ; 步骤(3)和步骤(6)的剪枝算法的复杂度为 $O(n)$ ; 步骤(5)的复杂度为 $O(|P^{(l_1)}| |D|)$ 。因为集合 $P^{(l_i)}$ 和集合 $D$ 均是 $V$ 的子集, 即这些集合中元素的个数都以 $n$ 为上界。每步的计算复杂性列表如表1所示。

表1 计算复杂性列表

步骤	功能	计算复杂度
预备阶段	调用Floyds算法, 计算每对节点之间的最短距离	$O(n^3)$
(1)	求解初始近似解	$O(n)$
(2)	迭代: 从第 $i$ 个近似解得到第 $i+1$ 个近似解	$O(n^2)$
(3)	剪枝: 减去第 $i+1$ 个近似解中的非组播叶子节点	$O(n)$
(4)	控制: 控制从(2)~(3)的迭代	$O(n)$
(5)	最后一次迭代计算, 求解最后的近似最优组播树	$O(n^2)$
(6)	剪枝: 功能同(3)	$O(n)$
总计	求解满足服务次序约束的近似最优组播树	$O(n^3)$

因此, 整个算法的计算复杂性跟Floyds算法的相同。从

计算复杂性的角度来讲,算法其余步骤并没有增加——尽管具体的计算量确实增加了。

#### 4 仿真

为了测试算法的性能,采用BRITE工具来生成带有AS区域的拓扑图,在AS图上随机地选择代理节点,代理节点之间的边由KMST算法<sup>[1]</sup>生成。在实验中,网络有如下属性:(1)无向图;(2)路由器节点的个数为100,路由器之间的边按Waxman模型来产生,其中连接边数 $m=4$ , $\alpha=0.15$ , $\beta=0.2$ ,边的费用在 $[1,5]$ 上均匀分布;(3)AS为10个,AS的分配按均匀分布;(4)选择20%的组播代理节点,均匀分布在各个AS域中,节点所对应的服务代价在 $[1,5]$ 上均匀分布,共5类服务,随机地放在代理节点上;(5)每个代理节点上可接入若干个终端节点(满足 $[1,5]$ 之间的均匀分布)。

在以上的仿真网络中随机地选择5%、10%、15%、20%、25%、30%的终端节点为组播会话中的基本节点(包括组播源端节点和目的节点),服务由1个变化到3个,次序分别为 $\langle 1 \rangle$ 、 $\langle 1,2 \rangle$ 、 $\langle 1,2,3 \rangle$ 。随着基本节点个数和服务需求的变化,观察解的质量(即生成的代价)。仿真结果如图4,可以看出随着基本节点数的增多,代价增大;服务种类增多,代价越大。

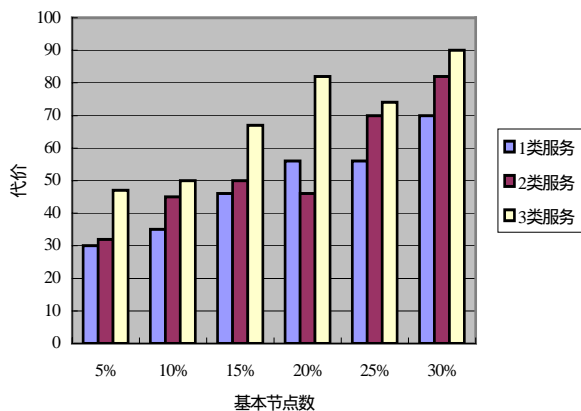


图4 服务需求、基本节点数、代价之间的关系

为了对比该算法与最优解之间的关系,按Waxman模型(连接边数 $m=4$ , $\alpha=0.15$ , $\beta=0.2$ )生成了20个节点的网络,边的代价带宽在 $[1,10]$ 上均匀分布,中间处理节点的代价在

(上接第99页)

的开销和时延更小,对网络资源的使用效率更高。然而,这个算法并不能对生成的多播树进行控制,往往会导致节点负载的不均衡和聚类间节点分布的不均衡,这使得自适应层次化算法不能很好地用于数据多播,而是更适用于控制信息的传输。

剪枝重置和构建顶层拓扑这两项操作可以克服自适应层次化聚类算法的缺陷。仿真表明,剪枝重置操作可以减少自适应层次化聚类树的层数;构建顶层拓扑操作则主要降低了多播树根节点的负担。虽然剪枝重置操作和构建顶层拓扑的操作增加了自适应层次化聚类树的开销和时延,但与DT协议生成的多播树相比,仍然有着显著的性能优势。

剪枝重置和构建顶层拓扑操作可以在完全的单层拓扑多播协议和完全的自适应层次化聚类算法之间得到折衷,从而

$[1,5]$ 上均匀分布。用穷尽方法求出最优值,与本文启发式算法计算的代价进行对比。对比值随组播节点变化趋势如图5所示,可以看出该算法的性能较好。

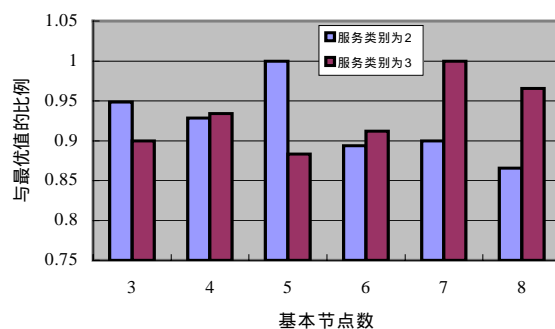


图5 该算法与最优值之间的关系

#### 5 总结

本文提出了Overlay组播网络中的服务组合问题,定义了相应的组播路由模型,并给出了求解该模型的一个启发式算法。最后,通过仿真说明该算法具有较好的数值效果。

#### 参考文献

- 1 Deering S. Multicast Routing in a Datagram Internetwork[D]. Stanford University, 1991-12.
- 2 Diot C, Levine B, Lyles J, et al. Deployment Issues for the IP Multicast Service and Architecture[J]. IEEE Network, 2000, 14(1): 78-88.
- 3 Reed S D, Clark D. End-to-end Arguments in System Design[J]. ACM Transactions on Computer Systems, 1984, 2(4): 195-206.
- 4 Chu S S Y, Rao S G, Zhang H. A Case for End System Multicast[J]. IEEE Journal on Selected Areas in Communication, 2002, 20(8): 1456-1471.
- 5 Jannotti J, Gifford D. Overcast: Reliable Multicasting with an Overlay Network[C]//Proceedings of the 4<sup>th</sup> Symposium on Operating System Design and Implementation, San Diego. 2000: 197-212.
- 6 Lao L, Cui J H, Gerla M. A Comparative Study of Multicast Protocols: Top, Bottom, or in the Middle[R]. Technical Report: UCLA CSD TR040054, <http://www.cs.ucla.edu/NRL/hpi/papers.html>, 2005-01.

融合二者的特点,更加适合于媒体数据多播应用。

#### 参考文献

- 1 Mathy L, Canonico R, Simpson S, et al. Scalable Adaptive Hierarchical Clustering[J]. IEEE Communications Letters, 2002, 6(3): 117-119.
- 2 Jin J W, Nahrstedt K. Large-scale Service Overlay Networking with Distance-based Clustering[C]//Proc. of ACM/IFIP/USENIX International Middleware Conference. 2003: 394-413.
- 3 Liebeherr J, Nahas M, Si Weisheng. Applicationlayer Multicasting with Delaunay Triangulation Overlays[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(8): 1472-1488.
- 4 Zegura E W, Calvert K, Bhattacharjee S. How to Model an Internetwork[C]//Proceedings of IEEE Infocom'96, San Francisco. 1996: 594-602.