

# 一种异构环境下覆盖多播网络路由算法\*

吴家皋<sup>1,2+</sup>, 叶晓国<sup>1,2</sup>, 姜爱全<sup>1,2</sup>

<sup>1</sup>(东南大学 计算机科学与工程系,江苏 南京 210096)

<sup>2</sup>(计算机网络和信息集成教育部重点实验室(东南大学),江苏 南京 210096)

## A Routing Algorithm in Heterogeneous Overlay Multicast Networks

WU Jia-Gao<sup>1,2+</sup>, YE Xiao-Guo<sup>1,2</sup>, JIANG Ai-Quan<sup>1,2</sup>

<sup>1</sup>(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

<sup>2</sup>(Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096, China)

+ Corresponding author: Phn: +86-25-83793073, Fax: +86-25-83792757, E-mail: jgwu@seu.edu.cn, http://www.seu.edu.cn

Received 2003-09-16; Accepted 2004-05-08

**Wu JG, Ye XG, Jiang AQ. A routing algorithm in heterogeneous overlay multicast networks. *Journal of Software*, 2005,16(6):1112-1119. DOI: 10.1360/jos161112**

**Abstract:** Due to the difficulties in deployment of IP multicast, the overlay multicast networks (OMN) are being increasingly recognized as a viable alternative to construct a general multicast service infrastructure. To settle the heterogeneity of bandwidth required by real-time multimedia application, the multicast routing problem in heterogeneous OMN is discussed. A new network model of heterogeneous OMN is described by extending the degree-constrained model. Adopting the strategy of layered bandwidth allocation, a heuristic routing algorithm, called layered compact tree (LCT) is proposed to build a minimum delay radius multicast tree in heterogeneous OMN. The properties of LCT are also proved and analyzed in theory. The simulation results show that LCT could reduce both hops and network resource usage of the tree effectively, and keep a little increase in delay radius while the allocated bandwidth decreases.

**Key words:** overlay multicast networks (OMN); routing algorithm; heterogeneity; bandwidth allocation; layered compact tree (LCT) algorithm

**摘要:** 由于IP多播在部署上的困难,覆盖多播网络(overlay multicast networks,简称OMN)作为构建通用的多播服务平台的另一可行途径正不断为人们所认可.针对实时多媒体应用对带宽需求的异构性,研讨了异构环境下OMN的路由问题.通过对度约束模型进行扩展,描述了一种新的适应异构环境的OMN网络模型.采用分层的带宽分配策略,提出了一种异构环境下构造OMN最小延时半径多播树的启发式算法——分层的压缩树(layered compact

---

\* Supported by the National Natural Science Foundation of China under Grant No.90104009 (国家自然科学基金); the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2001205 (江苏省自然科学基金)

**作者简介:** 吴家皋(1969—),男,江苏苏州人,博士生,助理研究员,主要研究领域为网络服务体系结构,覆盖网络,多播;叶晓国(1975—),男,博士生,主要研究领域为高性能网络体系结构及协议,主动网络技术,网络拥塞控制;姜爱全(1970—),男,博士生,主要研究领域为高性能网络体系结构及协议,移动互联网的QoS.

tree,简称 LCT) 算法,并对其性质进行了理论证明和分析.仿真实验结果表明,随着分配带宽的减少,LCT 算法能够有效地降低多播树的高度和网络资源使用量,并保持较低的多播树延时半径增幅.

**关键词:** 覆盖多播网络;路由算法;异构性;带宽分配;分层的压缩树算法

**中图法分类号:** TP393

**文献标识码:** A

以实时多媒体应用(如网络会议、网上直播和网络教室等)为代表的新型网络应用,对多播通信服务提出了迫切的要求.但是,由于技术和经济方面的原因<sup>[1]</sup>,直到现在,全网范围内的 IP 多播<sup>[2]</sup>服务尚未完全部署.于是,人们提出了覆盖多播(overlay multicast)的概念<sup>[3-10]</sup>,希望直接应用层提供多播服务.覆盖多播的思想是由端系统而不是核心路由器实现多播通信的所有功能,其最大的优势在于无须改变现有的 IP 网络,可灵活部署.覆盖多播按照系统结构可分为基于主机的(host-based)<sup>[3-6]</sup>和基于代理服务器的(proxy-based)两类<sup>[7-10]</sup>.前者完全由主机(应用)构成,后者则由多播服务节点(MSN)和客户(client)组成.MSN 负责构造核心多播分发树,Client 必须先连接到 MSN 上,方可进行数据的收发.Proxy-based 系统也被称为覆盖多播网络(OMN)<sup>[9]</sup>,其体系结构如图 1 所示.由于 MSN 比主机具有更高的处理能力、稳定性和安全性.而一个 MSN 可以同时服务于多个 Client.因此,我们认为,OMN 适合于构建通用的多播服务平台,并能提供更为强大、灵活和可扩展的服务.本文以 OMN 作为研究的系统环境.

由于 OMN 是在下层 IP 网络基础设施之上构建的虚拟、逻辑的网络,其路由和下层 IP 网络的路由通常不一致,这就可能造成 OMN 系统的延时增加和资源浪费.这一问题对于延时和带宽敏感的实时多媒体应用就显得更为突出.如何构造满足实时多媒体应用的 OMN 多播路由问题是当前 OMN 研究的热点和难点之一.在文献[9]中,Shi 等人提出了一个集中式的贪心路由算法——压缩树(compact tree)算法,目标是使多播树的任意两个 MSN 之间的最大延时(直径)最小化.Banerjee 等人在文献[11]中提出了一种分布式路由算法.以 Client 数对从源(source)到 MSN 的延时加权,该算法可用于解决加权平均延时最小化树等路由问题.在文献[7]中,Chawathe 等人提出了一个覆盖多播路由协议——Gossamer,其目标也是使多播树的平均延时最小化.但是与文献[9,11]直接构造多播树的树优先(tree-first)路由策略不同,Gossamer 采用网格优先(mesh-first)策略:首先,MSN 之间生成一个网状拓扑(mesh),然后在 mesh 之上运行类似于 DVMRP 协议<sup>[12]</sup>的应用层路由协议构造反向最短路径树.以上算法和协议的共同之处是,它们都是基于度约束(degree-constrained)网络模型的.“度”是一个 MSN 最多能同时向其他 MSN 复制和转发分组的数目.例如,在一个以平均带宽  $B$  传输媒体流的多播组中,一个转发能力为  $C$  的 MSN 的度为  $d = \lfloor C/B \rfloor$ ,即该 MSN 最多只能同时向  $d$  个 MSN 转发分组.度约束网络模型是对实际问题的简化和抽象,它适合所有 MSN 都以相同带宽转发媒体流的多播会话,称为同构的多播会话.

实际上,由于 MSN 的需求带宽由其所服务的 Client 决定,而 Client 对带宽的个性化要求,使各个 MSN 的需求带宽表现出较大的差异性.此时,若仍以相同的带宽进行多播传输显然不合理.为此,我们提出可以根据 MSN 的需求带宽的大小,通过分层或压缩编码等技术,将媒体流进行带宽转换后再转发.这样既满足了 Client 的服务需求又节省了系统的带宽资源,从而形成了异构的多播会话环境.在异构的多播会话中,多播树上每个分支可能传输不同带宽的流,不能简单地用度来抽象.因此,基于度约束的网络模型不再适用,必须提出新的适合异构环境的 OMN 网络模型.本文以异构的 OMN 多播环境为出发点,研究了面向实时多媒体应用的最小延时路由问题及其算法.有关异构环境下 OMN 的最小延时路由算法尚未见到全面报道,研究此问题无疑具有较大的实际意义.研究表明,我们提出的启发式算法能为在异构环境下构造节省带宽资源的最小延时多播树提供有效的途径.

本文第 1 节介绍我们提出的异构环境下的 OMN 网络模型及问题描述.第 2 节给出该问题的启发式算法,并对其性质进行了理论分析和证明.该算法的仿真性能评价在第 3 节中进行.第 4 节给出本文的研究结论.

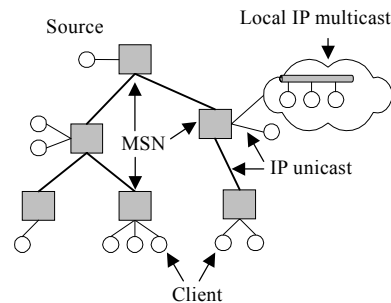


Fig.1 The Architecture of OMN

图 1 OMN 体系结构图

## 1 网络模型及问题描述

首先,我们假设 Client 总能连接到最近的 MSN,它们之间的通信延时可以忽略不计.其次,由于 Client 本身并不参与核心多播树的生成,因此可不考虑.这样,OMN 可以表示成一个完全有向图  $G=(V,E)$ ,其中  $V$  是顶点的集合, $E=V \times V$  是边的集合,对于  $\forall v \in V$  表示一个 MSN, $\forall e(u,v) \in E, u,v \in V$ ,表示顶点  $u,v$  之间的逻辑信道, $c(u,v)$  表示边  $e(u,v)$  的代价权重,这里用顶点  $u,v$  之间端到端的单播延时表示.

为了适应异构多播环境,我们对度约束模型予以扩展.对于  $\forall v \in V$ ,我们用一个四元组表示其属性  $v=(CapB(v),UsedB(v),ReqB(v),AllocB(v))$ ,其中, $CapB(v)$  表示  $v$  最大的带宽转发处理能力, $UsedB(v)$  表示  $v$  已经使用的转发带宽(初始化为 0), $ReqB(v)$  表示  $v$  的需求带宽, $AllocB(v)$  表示  $v$  的分配带宽,即  $v$  的实际接收媒体流的带宽, $v$  的父节点必须以此带宽向  $v$  转发分组.

另外,记  $ReqB_{\max} = \max_{v \in V} ReqB(v)$ ,  $ReqB_{\min} = \min_{v \in V} ReqB(v)$  分别表示  $V$  中最大和最小的需求带宽.

该模型区别于度约束模型,在以下两方面进行了扩展:一是用节点最大的带宽转发处理能力  $CapB(v)$  代替了度,从而使该模型具备了处理不同带宽分组转发的能力,二是引入了需求带宽  $ReqB(v)$  和分配带宽  $AllocB(v)$  两个参数, $ReqB(v)$  能够反映 MSN 需求带宽的异构性, $AllocB(v)$  则提供了对异构性施加控制策略的接口.不同的带宽分配策略,对多播路由的性能具有重要的影响(在本节后面将有相应讨论).该模型在  $AllocB(v)=ReqB(v)=B=$  常数时,可以定义  $d(v)=\lfloor CapB(v)/B \rfloor$ ,即与度约束模型等价.

在新的模型中,构造多播树必须注意由异构性而引起的新问题.首先是带宽分配问题,虽然我们可以通过降低节点的分配带宽以节省系统资源,但是分配带宽不能小于需求带宽,否则就无法满足节点对服务的需求;其次是节点成树次序问题,我们注意到,高分配带宽的节点可作为低分配带宽节点的父节点,反之则不成立.因为高带宽的流可以转换成低带宽流,而低带宽流一般不能还原为高带宽流;最后,与度约束模型类似,仍然要考虑节点处理能力的限制问题.根据以上分析,在异构环境下,构造任何多播树  $T$  必须满足以下约束条件:

**约束 1 (服务需求约束).**  $AllocB(v) \geq ReqB(v), AllocB(v) \in [ReqB_{\min}, ReqB_{\max}]$ .

**约束 2 (成树次序约束).**  $AllocB(v) \geq AllocB(u)$ , 对于  $\forall u \in Childof(v)$ . 其中  $Childof(v)$  表示节点  $v$  在多播树上孩子节点的集合.

**约束 3 (处理能力约束).**  $CapB(v) \geq UsedB(v)$ , 其中  $UsedB(v) = \sum_{u \in Childof(v)} AllocB(u)$ .

基于该模型,下面给出异构 OMN 环境下最小延时路由问题的描述.为了简化问题,我们这里只考虑单源的情况,设节点  $\forall r \in V$  为多播树  $T$  的根,令  $AllocB(r) = ReqB_{\max}$ ,即假设  $r$  总能以  $ReqB_{\max}$  发送数据.对于  $\forall v \in V, v \neq r$ ,定义  $\delta(v)$  为从  $r$  到  $v$  在  $T$  上的延时.  $Delay_T = \max_{v \in V, v \neq r} \delta(v)$  表示  $T$  上从树根到其他节点的最大延时,称为树的半径.

**问题 1.** 给定一个完全有向图  $G=(V,E), \forall e(u,v) \in E$ , 其权重为  $c(u,v), \forall v \in V, v=(CapB(v), UsedB(v), ReqB(v), AllocB(v))$ , 对于  $\forall r \in V$  为树根,构造一棵  $G$  的有向生成树  $T$ , 使得  $Delay_T$  最小化,并且  $T$  中的任意节点应同时满足约束 1~约束 3.

显然,该问题是 NP-Hard 问题,因为在  $AllocB(v)=ReqB(v)=B=$  常数时,该问题转换为度约束的最小半径树问题,而这个问题已被证明是 NP-Hard 的<sup>[9]</sup>.

容易看到,在该问题中,分配带宽  $AllocB(v)$  是一个重要的参数.下面,我们分析不同的带宽分配策略对多播树延时的影响.对于有  $n$  个节点的多播组,当对于  $\forall v \in V, AllocB(v)=ReqB_{\max}$  时,即按最大需求带宽分配实际接收带宽,则总的发送带宽为  $(n-1)ReqB_{\max}$ , 此时带宽消耗量最大,这种方案显然不合理.通过降低  $AllocB(v)$  在满足服务需求约束(约束 1)的前提下可以减少带宽使用.同时,随着  $AllocB(v)$  的下降,每个节点  $v$  能接纳的最大的孩子数将增加(约束 3),从而降低树的高度(hops),有利于减小树的半径.当  $AllocB(v)=ReqB(v)$  时,则达到了最小带宽分配,即按需分配.这似乎意味着带宽分配越低越好,但事实上并非如此,考虑到节点成树次序约束(约束 2)的存在,随着  $AllocB(v)$  接近于  $ReqB(v)$ , 系统的异构性就显示出来了,这时节点成树受到不同带宽的影响,有可能造成不能按延时最优的节点次序构造多播树.这时,最小的带宽分配和最小半径树的构造就形成了一对矛盾.最终的带宽

分配策略必须在带宽和延时之间取得折衷.这一问题非常复杂,我们只能通过启发式算法和仿真实验来进行评价,严格的理论分析尚未得出.

## 2 问题解决及算法分析

为了研究不同带宽分配对树的最小半径的影响.我们提出了一种分层的启发式多播树生成算法.该算法采用“分而治之”的思想,首先,将所有节点按  $ReqB(v)$  排序,进行分层的带宽分配,层次越高,所分配的带宽越大,同一层内节点的  $AllocB(v)$  都相同.然后,按节点所在的层次由高到低分层地构造多播树.由此可见,构造该算法要分 3 个步骤:(1) 构造带宽分配策略;(2) 构造层内路由算法;(3) 结合(1)(2)构造完整的算法.

### 2.1 分层的带宽分配策略

对于给定的需求带宽集合  $\{ReqB(v)\}$ ,总可以找到一个分层方法  $H=(L,\{LayerB(l)\})$ ,其中  $L$  是分层数, $LayerB(l)$  是第  $l$  层的带宽, $l \in [1,L]$ ,使得  $ReqB_{\max}=LayerB(L)>LayerB(L-1)>\dots>LayerB(1) \geq ReqB_{\min}$ ,则基于  $H$  的带宽分配策略是  $\forall v \in V$ ,

$$AllocB(v) = \begin{cases} LayerB(l), & \text{若 } LayerB(l-1) < ReqB(v) \leq LayerB(l), l > 1 \\ LayerB(1), & \text{若 } ReqB(v) \leq LayerB(1) \end{cases}$$

可以看出,当  $L=1$  时,  $\forall v \in V, AllocB(v)=ReqB_{\max}$ .随着  $L$  的增加,分层带宽间隔将越来越密,当  $L$  足够大时,一定能使  $AllocB(v)=ReqB(v)$  成立.所以,我们可以选择不同的分层数  $L$ ,比较不同  $L$  下构造的树的最小半径,就能分析不同的带宽分配对树的延时的影响.

### 2.2 层内路由算法

由于同一层内的节点构成同构的多播会话,因此层内节点的最小半径树生成算法,原则上可以选择任何基于度约束模型的相关算法.这里,我们采用修改了的压缩树算法(compact tree,简称 CT)<sup>[9]</sup>.CT 是一个集中式的贪心算法,目标是生成一棵直径最小的树.CT 的算法思想类似于 Prim 的最小生成树算法,每次选择使直径增加最小的节点加入树.已经证明,CT 算法得到的多播树直径是最优解的  $O(\log n)$  近似,其上限为

$$\lambda_{CT} = 2\epsilon \log_{d_{\min}} n, 2 \leq d_{\min} \leq n-1,$$

其中  $\epsilon$  表示任意两节点之间边最大的延时, $d_{\min}$  表示节点最小的度, $n$  为节点数.CT 算法的时间复杂度为  $O(n^3)$ .由此可见,CT 算法具有良好的性能代价比,有一定的代表性.另外,本文主要是研究异构环境对 OMN 多播路由的影响,所以采用 CT 算法是合理的选择.虽然 CT 算法是面向优化树直径的,但是,我们很容易将其移植到求解本文中的树最小半径问题,只要每次选择具有最小  $\delta(v)$  且满足约束条件的节点加入树即可.

### 2.3 LCT算法

**算法 1.** 分层的压缩树(layered compact tree,简称 LCT)算法伪代码.

1. Input:  $G=(V,E)$
2. Edge cost:  $c(u,v)$  for  $u,v \in V$
3. node proprieties:  $CapB(v), ReqB(v)$  for  $v \in V$
4. Layer method:  $H=(L,\{LayerB(l)\})$
5. root node:  $r \in V$
6. Output:  $T$  with the smallest  $Delay_T$
7.  $V(l)=\{\}$  for each  $l \in L$
8. for each  $v \in V, v \neq r$
9. for  $l=L$  down to 1
10. if  $(l>1$  and  $ReqB(v)>LayerB(l-1)$  and  $ReqB(v) \leq LayerB(l)$ )
11. or  $(l=1$  and  $ReqB(v) \leq LayerB(1))$  then
12.  $AllocB(v)=LayerB(l)$

```

13.       $V(l)=V(l)\cup\{v\}$ 
14.      break
15.   $T=\{W=\{r\},P=\{\}\}$ 
16.  for  $l=L$  down to 1
17.      for each  $v\in V(l)$ 
18.           $\delta(v)=\infty$ 
19.          for each  $t\in W$ 
20.              if  $UsedB(t)+AllocB(v)\leq CapB(t)$  and  $\delta(t)+c(v,t)<\delta(v)$  then
21.                   $\delta(v)=\delta(t)+c(v,t)$ 
22.                   $p(v)=t$ 
23.      while( $V(l)$  is not null)
24.          let  $u$  be the vertex in  $V(l)$  with the smallest  $\delta(u)$ 
25.           $W=W\cup\{u\},P=P\cup\{(p(u),u)\}$ 
26.           $V(l)=V(l)\setminus\{u\}$ 
27.           $UsedB(p(u))=UsedB(p(u))+AllocB(u)$ 
28.          for each  $v\in V(l)$ 
29.               $\delta(v)=\infty$ 
30.              for each  $t\in W$ 
31.                  if  $UsedB(t)+AllocB(v)\leq CapB(t)$  and  $\delta(t)+c(v,t)<\delta(v)$  then
32.                       $\delta(v)=\delta(t)+c(v,t)$ 
33.                       $p(v)=t$ 

```

该算法的执行步骤如下:(1) 先根据  $H$  进行分层的带宽分配,带宽分配的原则按不小于  $ReqB(v)$  的  $LayerB(l)$  中,最接近于  $ReqB(v)$  的带宽分配,从而保证了服务需求约束(代码 7~14 行).(2) 按所分的层次,由高向低循环调用修改的 CT 算法成树,这样保证了高带宽的节点总是先于低带宽的节点成树,满足节点成树次序约束(代码 15~33 行).在每次向树中加入新的节点时,都要检查节点的处理能力约束,只有在处理能力满足的情况下,才能加入(代码 20 行和 31 行).所以,我们有:

**定理 1.** LCT 算法同时满足约束 1~约束 3.

证明:由 LCT 算法的步骤即可得证. □

下面我们讨论 LCT 算法的上限.定义  $\lambda_{LCT(L=1)}$  和  $\lambda_{LCT(L>1)}$  分别为分层数为  $L=1$  和  $L>1$  时,LCT 算法求出的树半径上限.定义  $d_{\min}=\min_{v\in V} \lfloor CapB(v)/ReqB_{\max} \rfloor$ ,为节点最小度约束.假设  $2\leq d_{\min}\leq n-1$ ,则有如下定理.

**定理 2.**  $\lambda_{LCT(L>1)}\leq\lambda_{LCT(L=1)}$ .

证明:

(i) 当  $L=1$  时, $\forall v\in V, AllocB(v)=ReqB_{\max}$ ,类似于 CT 算法,可以证明 LCT 算法得到的树的半径上限为  $\lambda_{LCT(L=1)}=\varepsilon \log_{d_{\min}(L=1)} n$ ,其中  $\varepsilon$  为任意两节点之间边的最大延时, $2\leq d_{\min}(L=1)=d_{\min}\leq n-1$ .即在最坏情况下,节点按最小度约束  $d_{\min}(L=1)$  生成完全树,此树的高度为  $\log_{d_{\min}(L=1)} n$ ,而树的边权重为  $\varepsilon$ .

(ii) 当  $L>1$  时,根据 LCT 算法,在最坏情况下,每一层的节点都按该层的最小度约束生成完全树,对于  $\forall l\in [1,L], d_{\min}(l)=\min_{v\in V(l)} \lfloor CapB(v)/AllocB(v) \rfloor, V(l)$  为第  $l$  层节点集合.

因为  $V(l)\subseteq V$ ,有  $\min_{v\in V(l)} CapB(v)\geq\min_{v\in V} CapB(v)$ ,而  $AllocB(v)\leq ReqB_{\max}$ ,

所以  $d_{\min}(l)\geq d_{\min}(L=1)$ .我们取  $d_{\min}(L>1)=\min_{l\in[1,L]} d_{\min}(l)$ ,按  $d_{\min}(L>1)$  生成完全树,则  $\lambda_{LCT(L>1)}\leq\varepsilon \log_{d_{\min}(L>1)} n$ .

因为  $d_{\min}(L>1)\geq d_{\min}(L=1)$ ,

所以  $\varepsilon \log_{d_{\min}(L>1)} n\leq\varepsilon \log_{d_{\min}(L=1)} n=\lambda_{LCT(L=1)}$ ,故  $\lambda_{LCT(L>1)}\leq\lambda_{LCT(L=1)}$ . □

由此可见,LCT 算法获得的分层的多播树的半径上限比不分层的要小,说明分层策略有利于减小树的半径

值.但这只是理论上的上限值,实际树的半径是多种因素共同作用的结果.

最后分析 LCT 算法的时间复杂度.LCT 算法可分为两个阶段,第 1 个阶段为带宽分配阶段,其时间复杂度为  $O(n \cdot L)$ .第 2 个阶段为路由生成阶段,该阶段主要由两层循环构成,外层的循环按分层数循环,复杂度为  $O(L)$ ,里面的循环是同层节点按 CT 算法生成树,其复杂度  $O(|V(l)|^3)$ ,与各层的节点数有关,则第 2 阶段的时间复杂度为  $O(L \cdot |V(l)|^3)$ .所以,LCT 的算法时间复杂度为  $O(n \cdot L + L \cdot |V(l)|^3)$ .假设在节点均匀分布于各层的情况下,有  $|V(l)| = n/L$ .此时,LCT 的算法复杂度为  $O(n \cdot L + n^3/L^2)$ .当  $n \gg L$  时, $O(n \cdot L + n^3/L^2) \sim O(n^3/L^2)$ ,即与  $n^3$  成正比,与  $L^2$  成反比.

### 3 仿真实验及结果分析

我们利用 Georgia 理工学院的网络拓扑生成器 GT-ITM<sup>[13]</sup>产生基于中转和桩(transit-stub)模型的网络拓扑作为实验的底层网络.该网络拓扑共包含 6 075 个随机产生的路由器节点.其核心由 5 个平均拥有 15 个路由器节点的相互连通的中转域构成,平均每个中转域节点与 10 个桩域相连,每个桩域平均包含 8 个路由器节点.域内节点的链路连接采用 Waxman 模型(取  $\alpha=0.6, \beta=0.7$ ).这里,我们假设相连路由器节点之间的通信延时正比于它们之间的距离,而任意两个路由器节点之间的通信采用延时最短路径路由.

我们将 MSN 节点随机地接入网络拓扑的路由器上,忽略接入延时.发送数据的源节点  $r$  随机选择.MSN 的需求带宽分布特征与具体的应用类型和 Client 的异构性等因素相关,精确的 MSN 需求带宽分布模型已经超出了本文的范围.这里,我们假设 MSN 的需求带宽范围在  $[ReqB_{min}, ReqB_{max}]$  之间随机平均分布.同时,采用线性平均分方法进行带宽的分配,令  $LayerB(l) = ReqB_{max} - \Delta(L-l), \Delta = (ReqB_{max} - ReqB_{min})/L, l \in [1, L]$ .不失一般性,我们取  $ReqB_{max} = 16$  (带宽单位),  $ReqB_{min} = 1$  (带宽单位),MSN 的需求带宽为  $[1, 16]$  之间的随机整数.另外,令所有 MSN 最大的带宽转发处理能力为  $2ReqB_{max}$ .

下面,我们对 LCT 算法产生的多播树的高度、半径和网络资源使用量等指标进行仿真.其中树的高度和半径反映了树的延时,树的网络资源使用量表示树的代价.由于随着分层数  $L$  的增加,多播树的平均分配带宽将下降,我们定义分配带宽下降率为  $\rho = (avgTB(L) - avgTB(1)) / avgTB(1)$ ,其中  $avgTB(L)$  表示分  $L$  层时树的平均分配带宽.每次实验我们选择不同的 MSN 数  $n$  和分层数  $L$ ,随机进行 500 次实验,并取相应指标的平均作为实验结果.

#### 3.1 树的高度

图 2(a)和图 2(b)给出了不同 MSN 数时,多播树的高度与分层数  $L$  以及分配带宽下降率  $\rho$  之间的关系曲线.由图可见,树的高度随着  $L$  和  $\rho$  的增加而减少.这正是我们的算法所预期的,随着  $L$  的增加,MSN 节点的分配带宽  $AllocB(v)$  将下降,则在  $CapB(v)$  不变的情况下,每个节点能接纳的最大孩子数将增加,从而降低了整个树的高度.

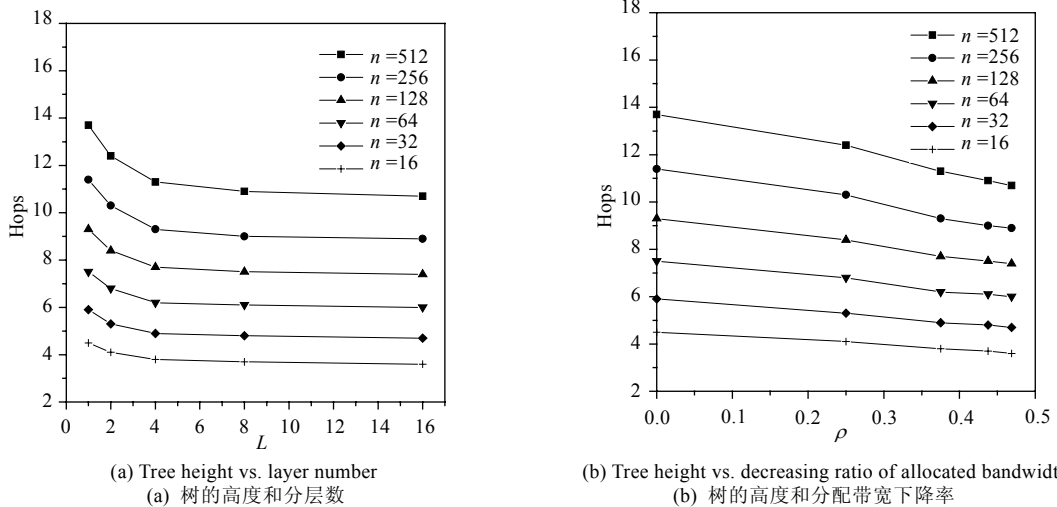
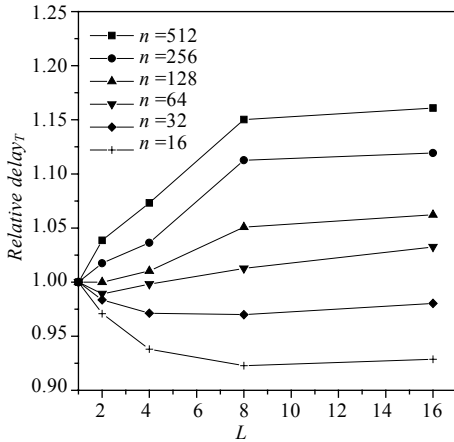


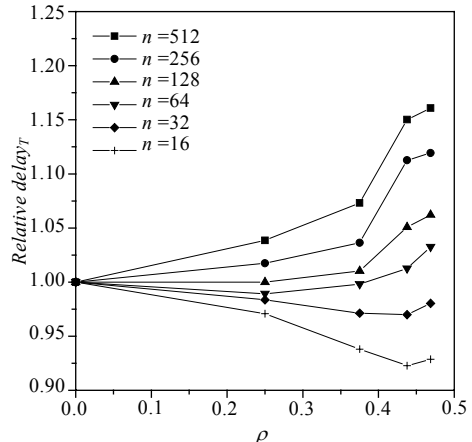
Fig.2  
图 2

### 3.2 树的半径

为了研究分层数对树的半径的影响,我们定义树的相对半径(*Relative Delay<sub>T</sub>*)为  $Delay_{T(L)}/Delay_{T(1)}$ ,即分层数  $L$  相对于分层数为 1 的半径.图 3(a)和图 3(b)给出了在不同 MSN 数时,相对半径和  $L$  以及  $\rho$  的关系曲线.从图中可以看到,当 MSN 数较小( $n \leq 32$ )时,随着  $L$  和  $\rho$  的增加,树的相对延时半径总体呈下降趋势,这也是我们所希望的结果.而当 MSN 数较大( $n > 32$ )时,树的相对延时半径随着  $L$  和  $\rho$  的增加反而会有所上升.这是因为,虽然随着  $L$  的增加,树的高度将有所下降,但是,由于受到约束 3 的限制,随着  $L$  的增加,每一层生成树可选择的最小延时边的范围将减小,从而影响了最佳边的选择.整个树的半径是树的高度和树的边长之间相互作用的结果.尽管如此,我们可以看到树的半径增加的幅度非常有限.例如,当  $L$  从 1 增加到 16 时,整个树的带宽分配减少了将近一半 ( $\rho=0.47$ ),而树的半径在  $n=512, 256, 128$  和 64 时,分别只增加了 16%,11%,6%和 3%,大大低于带宽的减少.



(a) Tree relative radius vs. layer number  
(a) 树的相对半径和分层数



(b) Tree relative radius vs. decreasing ratio of allocated bandwidth  
(b) 树的相对半径和分配带宽下降率

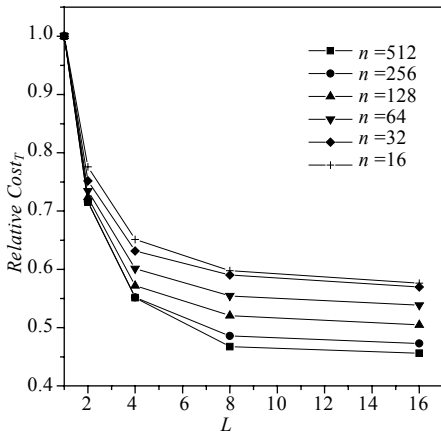
Fig.3  
图 3

### 3.3 树的代价

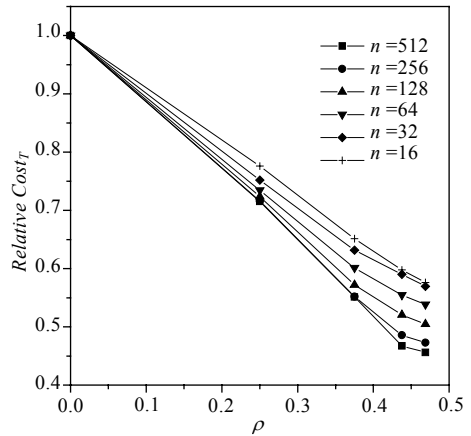
为了进一步证明 LCT 算法的有效性,我们用多播树的网络资源使用量作为算法有效性的评价指标.一棵多播树的网络资源使用量通常被认为是该树的代价,其定义为  $Cost_T = \sum_{i \in Links} dl_i \cdot bl_i$ ,其中,Links 是所有参与多播树数据传输的网络链路的集合, $dl_i$  和  $bl_i$  分别表示链路  $i$  的延时和被使用的带宽.同上,我们也定义树的相对代价 (*Relative Cost<sub>T</sub>*)为  $Cost_{T(L)}/Cost_{T(1)}$ .图 4(a)和图 4(b)给出了在不同 MSN 数时,相对代价和分层数  $L$  以及  $\rho$  的关系曲线.从图中我们看到,树的代价随着  $L$  和  $\rho$  的增加而减少.因此,利用分层策略,LCT 算法可以有效地减少多播树的网络资源使用量,从而说明该算法是有效的.

## 4 结论

本文对异构环境下 OMN 的路由问题进行了分析和研究,指出了基于度约束的网络模型的局限性,并描述了新的适合异构多播的 OMN 网络模型.采用分层的带宽分配策略,我们提出了一个异构环境下构造 OMN 的最小延时半径多播树的启发式算法——分层的压缩树算法,并从理论上证明了分层分配带宽将降低 LCT 算法生产的多播树的半径上限.仿真结果表明,在通过分层策略减少多播树的分配带宽的同时,LCT 算法可以明显地降低树的高度和网络资源使用量(代价).多播树的半径在 MSN 数较少时,也随分层数  $L$  的增加而下降.虽然在 MSN 数较大时,树的半径随  $L$  的增加有所上升,但是其升幅相对较小.因此,LCT 算法可以有效地解决异构环境下 OMN 中的最小延时路由问题.下一步工作,我们将从分布式和组成员动态管理等方面对 LCT 算法进行改进.



(a) Tree relative cost vs. layer number  
(a) 树的相对代价和分层数



(b) Tree relative cost vs. decreasing ratio of allocated bandwidth  
(b) 树的相对代价和分配带宽下降率

Fig 4  
图 4

**References:**

- [1] Diot C, Levine BN, Lyles B, Kassem H, Balensiefen D. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 2000,14(1):78–88.
- [2] Deering S, Hosting extensions for IP multicast. IETF, RFC 1112, 1989.
- [3] Chu YH, Rao SG, Zhang H. A case for end system multicast. In: Kurose J, Nain P, eds. *Proc. of the ACM SIGMETRICS 2000*. Santa Clara: ACM Press, 2000. 1–12.
- [4] Francis P. Yoid: Extending the multicast internet architecture. Technical Report, Berkeley: AT&T Center for Internet Research at ICSI (ACIRI), 2000. <http://www.aciri.org/yoid/>
- [5] Pendarakis D, Shi S, Verma D, Waldvogel M. ALMI: An application level multicast infrastructure. In: Anderson T, ed. *Proc. of the 3rd USENIX Symp. on Internet Technologies & Systems*. San Francisco: USENIX Press, 2001. 49–60.
- [6] Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users. In: Lee D, Orda A, eds. *Proc. of the IEEE INFOCOM 2002*. New York: IEEE Communication Society, 2002. 1366–1375.
- [7] Chawathe Y. Scattercast: An architecture for internet broadcast distribution as an infrastructure service [Ph.D. Thesis]. Berkeley: University of California, 2000.
- [8] Jannotti J, Gifford D, Johnson K, Kaashoek M, O’Toole J. Overcast: Reliable multicasting with an overlay network. In: Jones MB, Kaashoek F, eds. *Proc. of the 4th USENIX Symp. on Operating Systems Design and Implementation*. San Diego: USENIX Press, 2000. 192–212.
- [9] Shi S, Turner J. Multicast routing and bandwidth dimensioning in overlay networks. *IEEE Journal on Selected Areas in Communications*, 2002,20(8):1444–1455.
- [10] Stoica I, Adkins D, Zhuang S, Shenker S, Surana S. Internet indirection infrastructure. In: Paxson V, Balakrishnan H, eds. *Proc. of the ACM SIGCOMM 2002*. Pittsburgh: ACM Press, 2002. 73–88.
- [11] Banerjee S, Kommareddy C, Kar K, Bhattacharjee S, Khuller S. Construction of an efficient overlay multicast infrastructure for real-time applications. In: Roberts J, Shroff N, eds. *Proc. of the IEEE INFOCOM 2002*. San Francisco: IEEE Communication Society, 2002. 1521–1531.
- [12] Deering S. Multicast routing in internetworks and extended LANs. In: Landweber L, ed. *Proc. of the ACM SIGCOMM 1988*. Stanford: ACM Press, 1988. 55–64.
- [13] Zegura E, Calvert K, Bhattacharjee S. How to model an internetwork. In: Mukherjee B, ed. *Proc. of the IEEE INFOCOM 1996*. San Francisco: IEEE Communication Society, 1996. 594–602.