

# 支持流媒体的应用层组播研究\*)

张 谓 吴慧中 肖 亮

(南京理工大学计算机科学与技术学院 南京 210094)

**摘 要** 在当前没有 IP 组播支持以及大多数端主机访问带宽有限的 Internet 网络条件下,普通的端主机提供实时流媒体应用服务缺乏有效的通信手段,应用层组播是解决这一问题的有效通信技术。针对网络电视、视频点播类型的大规模实时流媒体应用,本文提出一个两层应用层组播框架结构 ALMF,并对其中组播服务结点覆盖网络的构建和维护进行了研究,提出了一个覆盖网络路由组播模型 MOSSMRM 和基于智能蚁群的路由算法 IACSMR,随后将该路由算法和几种启发式算法进行了比较。仿真实验结果表明,IACSMR 算法是一个有效的应用层组播路由算法。

**关键词** 应用层组播,覆盖网络,路由算法,实时流媒体

## Research of Application Level Multicast for Streaming Media

ZHANG Xu WU Hui-Zhong XIAO Liang

(School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094)

**Abstract** Because most of end hosts in today's Internet have limited access bandwidth and the Internet does not support IP multicast, a general end host can hardly provide service of real time streaming media applications. Application layer multicast is a novel communication method to solve this problem. The paper puts forward a two-tier application layer multicast framework for large-scale real time streaming media applications such as IPTV and VOD. Then how to construct and maintain overlay network between multicast service nodes is discussed. After that an application level multicast routing mode for overlay network and a routing algorithm based on intelligence ant colony are devised. A simulation experiment compared the routing algorithm with some other heuristic algorithms and the result shows that the routing algorithm is efficient.

**Keywords** Application level multicast, Overlay network, Routing algorithm, Real time streaming media

随着 Internet 的不断发展,以流媒体为核心的新业务不断涌现,例如视频点播、视频会议、虚拟空间会议和网络电视等,组播技术是解决这些应用的关键技术之一。从 1989 年 Deering 提出 IP 组播开始,科研人员对 IP 组播展开了广泛的研究。IP 组播是对互联网“单播、尽力转发”模型的重要扩充,主要功能在路由器上实现。但由于 IP 组播在技术和管理上都存在许多问题,在互联网上尚未广泛部署。最近,有一些研究人员提出了应用层组播(ALM: Application Layer Multicast)的概念,其基本思想是指在应用层而不是网络层上实现组播能力,由端系统而不是路由器实现组播转发功能。在当前没有 IP 组播支持以及大多数主机访问带宽有限的 Internet 网络条件下,应用层组播是一种有效的组播通信技术<sup>[1-5]</sup>。

大规模的实时流媒体数据分发应用,例如网络电视、视频点播,一般存在单一的组播源节点。针对这类实时流媒体应用,本文提出一个应用层组播框架结构,并对其中的主要内容进行研究。

### 1 应用层组播框架

对于大规模的实时流媒体应用,仅基于单纯的算法在端主机之间构建应用层组播树,不能够满足应用需求,需要实施某种策略,才能构建满足大规模的实时流媒体应用需求的应

用层组播系统。借鉴 HMTF<sup>[2]</sup>的设计思想,本文提出一个两层应用层组播框架结构(Application Layer Multicast Framework,记为 ALMF),如图 1 所示。该结构采用 IP 组播技术和应用层组播技术相结合的方法来解决大规模的实时流媒体应用组通信问题。

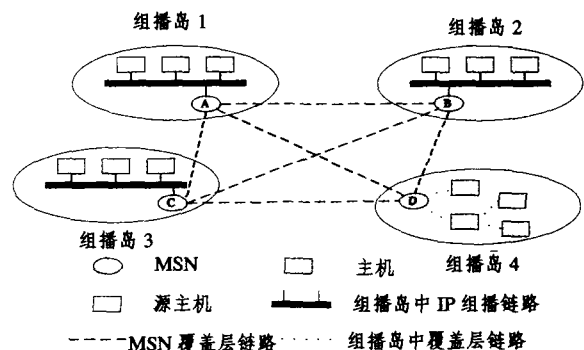


图 1 应用层组播框架 ALMF

在此引入组播岛的概念。组播岛可以是一个部署了 IP 组播服务的局域网,也可以是只提供单播服务的局域网(网内可由应用层组播提供组播服务)。MSN (Multicast Service Node) 可以是预先部署的服务器(当组播岛数量较大并且不

\*) 高等学校博士学科点专项科研基金(No. 20020288024)、长三角重大科技联合攻关项目(BE2004400)。张 谓 博士研究生,主要研究方向为分布式虚拟现实、应用层组播;吴慧中 教授,博导,主要研究方向为虚拟现实与系统仿真;肖 亮 讲师,博士,主要研究方向为分布式虚拟现实、图像建模理论。

断变化时,这种方法代价较大),也可以动态选择组播岛中一个主机充当。MSN 之间维护一个覆盖网络,负责组播岛之间的数据分发。MSN 接收到数据后,除了向所在组播岛分发,还要沿 MSN 覆盖网络分发数据。以图 1 为例,源主机首先利用 IP 组播服务向其所在组播岛 1 上的目的主机(包括 MSN A)发送数据,A(源 MSN)接收到数据后,利用 MSN 之间的覆盖网络将数据分发到 MSN B、C、D。MSN B、C 接收到数据后,利用 IP 组播服务分别向其所在组播岛 2、组播岛 2 上的目的主机发送数据。由于组播岛 4 不支持 IP 组播,则 MSN D 采用应用层组播将数据转发到各目的节点。在整个数据分发过程中,MSN 之间覆盖网络性能的优劣直接影响实时流媒体应用的性能,因此构建和维护 MSN 之间的覆盖网络是 ALMF 中的首要问题。

## 2 构建和维护 MSN 之间的覆盖网络

构建和维护 MSN 之间的覆盖网络,采用集中式方法比采用分布式方法更为方便,由维护覆盖层结构而导致的网络负载较小。虽然采用集中式方法存在单点失效(Single Point of Failure)的问题,即集中点(Rendezvous Point,简称 RP)失效会导致整个系统不能正常工作,但这一问题可以通过使用备份集中点来解决。对于源具体组播应用的覆盖网络,采用源具体组播树比共享组播树更合适。构建源具体组播树可以采用直接构建树的方法或者基于网优先的方法,采用后者可以有效地减少由维护覆盖层结构所导致的计算负载。

综上所述,本文采用了集中式方法以及基于网优先的策略来构建和维护 MSN 之间的覆盖网络(源具体应用层组播树)。不同于 Narada 采用非完全虚拟图来表示用于维护 MSN 之间关系的网 MESH<sup>[4]</sup>,ALMP 采用完全虚拟图来表示 MESH,从而简化了维护 MESH 的过程。

### 2.1 主机加入

当主机 H 想加入组时,它首先访问集中点 RP。RP 将组中已存在的 $\{MSN_0, MSN_1, \dots, MSN_i\}$ 的地址发送给 H, H 可以通过 mcast 组播测试工具或其他方法来检测是否属于支持 IP 组播的组播岛。如果 H 属于某个支持 IP 组播的组播岛,则将 H 加入该组播岛。否则,可以根据应用的具体需求选定策略(例如根据 H 的本地路由器是否支持 IP 组播及组中现有 MSN 数目等)将 H 作为  $MSN_{i+1}$  或者加入不支持 IP 组播的组播岛。具体方法本文不做讨论。

当 H 作为  $MSN_{i+1}$  时,MSN 之间的 MESH 发生变化, $MSN_{i+1}$  通过 Ping 工具获取与其他 MSN 之间的近似反应时间,并将 MSN 之间的反应时间以及本地网络带宽信息发送给 RP。RP 收到上述信息后,在已有的 MESH 上添加新的链路 $\{(MSN_0, MSN_{i+1}), (MSN_i, MSN_{i+1}), \dots, (MSN_i, MSN_{i+1})\}$ 。随后 RP 调用组播路由算法基于 MESH 计算组播树,并将组播树的变化通知到相应的 MSN。

### 2.2 主机离开

当主机 H 想要退出组时,分为 4 种情况:①H 属于支持 IP 组播的组播岛, H 是该组播岛的 MSN。首先 H 向集中点 RP 发出退出请求,同时推荐组播岛内一个主机 H'(如果存在)作为替代 MSN。RP 收到请求后,更新 MESH。随后 RP 调用组播路由算法基于 MESH 重新计算组播树,并将组播树的变化通知到相应的 MSN。最后向 H 返回同意退出消息,

RP 删除关于 H 的信息, H 断开与其他 MSN 之间的所有连接。接着 H 使用 IGMP 消息向组播路由器通知它退出组播组。②H 属于支持 IP 组播的组播岛, H 是该组播岛内的普通主机, H 使用 IGMP 消息向组播路由器通知它退出组播组。③H 属于不支持 IP 组播的组播岛, H 是该组播岛的 MSN, 在组播岛内重新计算剩余主机(如果存在)的组播树, 将组播树的根主机 H'推荐给 RP, 作为替代 MSN。RP 收到请求后,更新 MESH。随后 RP 调用组播路由算法基于 MESH 重新计算组播树,并将组播树的变化通知到相应的 MSN。最后向 H 返回同意退出消息,RP 删除关于 H 的信息, H 断开与其他 MSN 之间的所有连接。④H 属于不支持 IP 组播的组播岛, H 不是该组播岛的 MSN, 在组播岛内重新计算剩余主机的组播树。

### 2.3 主机失效

所有 MSN 定期地测量相互之间反应时间以及本地带宽,并传送到 RP, RP 会根据参数的变化计算组播树。如果组播树的性能提高超过一个阈值,则更新组播树。如果 RP 在一段时间间隔内没有收到  $MSN_j$  的消息,则认为  $MSN_j$  失效,在 MESH 上删除所有和  $MSN_j$  相连的边, RP 在  $MSN_j$  所在组播岛中重新选出一个主机(如果存在)作为替代 MSN。随后 RP 调用组播路由算法基于更新的 MESH 重新计算组播树,并将组播树的变化通知到相应的 MSN,删除关于  $MSN_j$  的信息。不支持 IP 组播的组播岛内主机失效可采用类似的方法解决,支持 IP 组播的组播岛内主机失效可由 IGMP 协议处理。

## 3 MSN 覆盖网络的路由模型

构建覆盖网络的路由策略是针对具体的应用而定的,没有统一的标准,常用的标准包括带宽的使用、端到端数据传输延迟和组播节点的“度”等。国内外已有研究人员提出了一些针对视频会议类型应用的应用层组播路由模型及相应的组播路由算法, Shi 提出了 MDDL 和 LDRB 模型以及解决该模型的启发式算法 CT 和 BCT<sup>[6,7]</sup>; 参考了 MDDL 和 LDRB 模型,潘耘提出了一个多约束路由模型 MRRBDL 以及相应的组播路由遗传算法<sup>[8]</sup>。除了上述模型之外,还有一些针对具体问题的应用层组播路由模型及算法,这些模型大多与 MDL 和 LDRB 类似,因此不一一列举。

MDDL 和 LDRB 模型的应用背景是应用层组播方案 ALMI<sup>[3]</sup>。ALMI 是由华盛顿大学提出的一个针对小规模、多数据源的一个应用层组播方案,典型应用是视频会议系统。该方案基于树优先的策略,在组播成员之间维护一棵共享组播树,所以 MDDL 和 LDRB 模型的端到端数据传输延迟优化目标具体而言是优化共享组播树直径,并不适合组播源节点具体的应用。MRRBDL 模型的端到端数据传输延迟优化目标具体而言是优化组播源节点到目的节点的最长简单路径。由此可见,该模型是面向组播源节点具体的应用,并不像其文章中所说的那样适合 ALMI 方案。另外,该模型的负载均衡优化目标函数并不适当。根据对上述模型的分析,本文提出一种适用于网络电视、视频点播等大规模实时流媒体应用的应用层组播路由模型。

网络电视类型应用是组播源节点具体的流媒体应用,必须可以接受的质量支持实时视频,这通常需要较大的带宽。以虚拟会堂应用为例,为了有效表达会议发言者身体姿势和

面部表情以及可能的相互凝视,需要以 15fps 的频率传输 320 × 240 分辨率的实时视频,经 MPEG-4 编码后,产生平均 100kbps 的网络流量<sup>[8]</sup>。绝大多数 MSN 由普通的主机充当,使用的访问网络是由 ISP 提供的高速 DSL 连接(典型的上行带宽为 200~300kbps,下行带宽为 1Mbps)。可以看出,一般的 MSN 只能支持几个下行 MSN,也就是说 MSN 的“出度”受约束。

从应用的角度看,网络电视类型应用的一个重要性能指标是实时视频从源节点到各目的节点的延迟;另外一个性能指标是 MSN 节点的负载平衡,要避免由于负载过于集中而带来的节点处理瓶颈。从网络优化的角度看,可由两个性能指标来评价:①压力度,用于说明网络物理链路带宽的利用率,定义为流经物理链路的相同报文的数量;②资源占用率,用于说明网络物理链路所消耗的资源,定义为所有参与组播会话的物理链路所消耗的资源之和。大多数 MSN 不知道网络的拓扑结构,如果要优化路由器之间的物理链路的压力度,必须获取网络的拓扑信息,本文对此不做讨论。而优化 MSN 访问链路的压力度可由 MSN 的负载平衡来实现,资源占用率的优化可由优化生成树的整体延迟来实现。

由上述分析可知,网络电视类型应用需求是多元的,需要综合性能优化的源节点具体组播树,因此本文提出一个多目标优化组播路由模型 MOSSMRM(multiple object, source specific, multicast route model)。MOSSMRM 优化目标是:①优化组播源 MSN 到目的 MSN 的延迟,一方面要求优化源 MSN 到目的 MSN 的最长延迟,另一方面要求优化生成树的整体延迟(也就是优化源 MSN 到各目的 MSN 的平均延迟);②优化 MSN 的负载平衡,形式化描述如下:

给定一个无向完全图  $G=(V, E)$ ,  $d_{\max}(v) \in N$  是每个顶点  $v \in V$  的度上限,表示 MSN 的处理能力与带宽能力上限; $c(e) \in Z^+$  是每条边的  $e \in E$  的代价,表示两个 MSN 之间的数据传输延迟;在图  $G$  中寻找一棵生成树  $T$ ,满足

$$\begin{cases} \min \text{delay}_{\max}(T) \\ \min D_{\chi}(T) \\ \min \cos t(T) \\ s. t. d_T(v) \leq d_{\max}(v) \end{cases} \quad (1)$$

公式(1)中,  $\text{delay}_{\max}(T)$  表示树  $T$  组播源 MSN 到目的 MSN 的最长延迟;  $d_T(v)$  表示树  $T$  上节点  $v$  的度,  $\chi(T)$  表示  $d_T(v)/d_{\max}$  所组成的向量,其方差  $D_{\chi}(T)$  能有效表示具有不同访问链路承载能力的 MSN 负载分布是否均衡;  $\cos t(T) = \sum_{e \in T} c(e)$  表示生成树的整体延迟。

因为带有度约束的最小生成树问题是 NP 完全问题,而该问题是 MOSSMRM 模型求解的子问题,所以得出 MOSSMRM 模型求解是 NP 完全问题。对于 NP 完全问题,我们知道有效率的非确定算法,但是不知道是否存在有效率的确定性算法,同时不能证明它不存在有效率的确定性算法。当面对一个必须解决的 NP 完全性问题时,一般是采用启发式算法来找到问题的一个近似解。

智能蚁群算法是一种具有群体智能特征的启发式算法<sup>[10]</sup>,它是由意大利 M. Dorigo 等人首先提出的,其基本思想是模拟蚂蚁依赖信息素进行通信而显示出的社会行为。在 Agent 总体的基础上,由一个贪心算法指导下的自催化过程引导每个 Agent 的行动。该算法模拟了蚂蚁觅食时的行为,按照启发式思想,通过信息素的诱导作用,逐步收敛到问题的最优解。蚁群算法结合了问题求解的快速性、全局最优特征

以及在优化过程初期解的合理性等特性。

智能蚁群算法在解决组合优化类 NP 完全问题求解方面表现出了突出的适用特征,典型的为 TSP 问题的求解。M. Dorigo 所发表的一篇经典论文系统论述了解决 TSP 问题的蚁群协作方法。实验研究的结果表明,蚁群系统的性能要优于其它诸如模拟退火和进化计算等启发式算法。TSP 问题是 MOSSMRM 模型求解问题的一个特例,问题的求解过程具有相似性,所以本文尝试用智能蚁群算法来求解 MOSSMRM 模型问题。

#### 4 基于智能蚁群的组播路由算法

MOSSMRM 模型是一个多目标组合优化问题,求解方法之一是加权组合法。为每个目标函数引入权重系数,化多目标优化为单目标优化,如式(2)所示。

$$\begin{cases} \min f = \lambda_1 \text{delay}_{\max}(T) + \lambda_2 D_{\chi}(T) + \lambda_3 \cos t(T) \\ s. t. d_T(v) \leq d_{\max}(v) \end{cases} \quad (2)$$

式中,  $\lambda_1 + \lambda_2 + \lambda_3 = 1.0$ 。

智能蚁群算法的具体求解步骤可见参考文献[10],本文在此不再赘述。将智能蚁群算法应用到 MOSSMRM 模型求解中,组播路由算法(记为 IACSMR)定义人工蚂蚁为具有下列特征的智能体:

①在周期  $(t, t+n-1)$  中,第  $k$  个蚂蚁构建一棵组播树  $T_k$ ;记组播树上已有满足度约束的 MSN 集合为  $T_{\text{agg}}$ ,未加入组播树的剩余 MSN 集合为  $L_{\text{agg}}$ ;该蚂蚁在选择  $t+i, i \in [0, n-1]$  时刻的出发点时可以跳跃到组播树上  $T_{\text{agg}}$  中的任意节点,蚂蚁的目的 MSN 是  $L_{\text{agg}}$  中的任意节点;

②令  $P_{ij}^k(t)$  表示第  $k$  只蚂蚁在  $t$  时刻由出发 MSN  $i$  到目的 MSN  $j$  的寻径概率,  $\eta_{ij}$  表示 MSN  $i, j$  之间描述贪婪式启发搜索特性的能见度,  $T_{ij}(t)$  表示  $t$  时刻 MSN  $i, j$  之间描述正反馈的信息素浓度;寻径概率是能见度和信息素浓度二者的函数,如式(3)所示:

$$P_{ij}^k(t) = \begin{cases} \frac{[T_{ij}(t)]^{\alpha} \times [\eta_{ij}]^{\beta}}{\sum_{m \in T_{\text{agg}}} \sum_{n \in L_{\text{agg}}} [T_{mn}(t)]^{\alpha} \times [\eta_{mn}]^{\beta}}, & i \in T_{\text{agg}}, j \in L_{\text{agg}} \\ 0, & \text{其他} \end{cases} \quad (3)$$

式中,  $\alpha, \beta$  为控制信息素和能见度之间相对重要性的参数。若  $\alpha=0$ ,则意味着不再考虑信息素水平,算法就演变为有多重起点的随机贪心算法。

令  $\text{delay}(s, j)$  表示 MSN  $j$  添加到组播树  $T_k$  后到源节点  $s$  的最短距离,  $c(i, j)$  表示 MSN  $i$  和 MSN  $j$  之间的距离(延迟)。由于  $d_T(i)/(d_{\max}(i) \times d_{\max}(j))$  综合表示了 MSN  $i$  的负载率和 MSN  $j$  的负载能力,所以可作为 MSN 负载平衡的贪心启发式搜索策略。

针对优化目标函数公式(2),本文采用  $\text{delay}(s, j), d_T(i)/(d_{\max}(i) \times d_{\max}(j)), c(i, j)$  来计算 IACSMR 算法中的能见度  $\eta_{ij}$ ,如下所示:

$$\eta_{ij} = 1 / (\lambda_1 \text{delay}(s, j) + \lambda_2 d_T(i) / (d_{\max}(i) \times d_{\max}(j)) + \lambda_3 c(i, j))$$

③一个周期完成后,第  $k$  只蚂蚁在其经过的边  $(i, j)$  上留下信息素,记为  $\Delta T_{ij}^k$ ,计算公式见式(4):

$$\Delta T_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{若 } (i, j) \in T_k \\ 0, & \text{其他} \end{cases} \quad (4)$$

式中,  $L_k = \lambda_1 \text{delay}_{\max}(T_k) + \lambda_2 D_{\chi}(T_k) + \lambda_3 \cos t(T_k)$ 。在

传统的蚁群算法中  $Q$  是一个常量,不能适应由于应用规模变化而引起解的综合性能指标  $L_k$  变化。为了适应 MSN 规模以及端到端延迟的不同,有必要将信息素  $Q$  的大小与  $L_k$  关联起来。本文采用了  $\min L_k$  和系数  $\sigma$  来控制信息素  $Q$  的大小,  $\sigma$  一般为信息素初始值  $ph$  的几十分之一。实验表明,该方法是可行的。

当  $m$  只蚂蚁均构建完组播树之后,更新边  $(i, j)$  上的信息素,计算公式见式(5):

$$T_{ij}(t+n) = \rho T_{ij}(t) + \sum_{k=1}^m \Delta T_{ij}^k \quad (5)$$

式中,  $1-\rho$  表示在周期  $(t, t+n-1)$  中边  $(i, j)$  上信息素的挥发系数。

IACSMR 算法具体步骤如下:

- Step1 初始化:  
 $t := 0; NC := 0;$   
 为每条  $(i, j)$  边的信息素浓度设置一个初始值  $T_{ij}(t) = ph,$   
 $\Delta T_{ij} := 0;$   
 将  $m$  个蚂蚁放在源 MSN  $s$  上;
- Step2 for  $k := 1$  to  $m$  do {  
 $T_{agg} = \{s\}, L_{agg} = V - T_{agg};$   
 for  $l := 1$  to  $|V|$  do {  
 以概率  $P_{ij}^k(t)$  从  $L_{agg}$  中选择 MSN  $j$  与  $T_{agg}$  中的 MSN  $i$  相连;  
 $T_{agg} = T_{agg} \cup \{j\}, L_{agg} = L_{agg} - \{j\};$   
 $d_{T_k}(i) ++;$   
 if  $(d_{T_k}(i) == d_{max}(i))$   
 $T_{agg} = T_{agg} - \{i\};$
- Step3 由 Step2 得到  $m$  棵组播树 Tree;  
 $Q = \sigma \cdot \min L_k;$   
 for  $k := 1$  to  $m$  do {  
 $\Delta T_{ij} = \begin{cases} Q, & \text{若 } (i, j) \in T_k \\ 0, & \text{其他} \end{cases}$   
 记录  $m$  棵组播树中的最优解;
- Step4 for 每条边  $(i, j)$  {  
 $T_{ij}(t+n) = \rho T_{ij}(t) + \sum_{k=1}^m \Delta T_{ij}^k$   
 $t := t+n$   
 $NC := NC+1$
- Step5 while  $(NC < NC_{max} \&\& \text{解非停滞状态})$  {  
 转到 Step2;  
 记录到本次循环为止的最优解;}  
 输出最优组播树;

## 5 仿真实验

为了验证所提出的 IACSMR 算法,采用随机生成的完全图表示 MESH, MESH 边的代价为一个范围内的随机整数,每个节点的度约束为大于 2 的一个随机整数,通过仿真对 IACSMR 算法所得到的组播树性能进行分析。

### 5.1 算法的性能分析

取 MESH 边的代价为  $[6, 30]$  中的随机整数,每个节点的度约束为  $[2, 3]$  中的随机整数,对组播节点数目  $n = \{5, 10, 20, 30, 50\}$  分别进行 10 次组播路由计算,计算所得的组播树平均性能。

当  $\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$  时, MOSSMRM 模型简化为求解带度约束最短路问题,本文将 IACSMR 算法和一种贪心算法 G-CT 进行性能比较。该贪心算法参考了 Shi 提出解决 MDDL 模型的 CT 算法<sup>[6,7]</sup>,伪代码如下所示:

```

 $T_{agg} = \{s\}, L_{agg} = V - T_{agg};$ 
for  $greedy_i := 1$  to  $|V|$  do {
     $greedy_{cost}[greedy_i] = 0;$  //记录树上 MSN 到源 MSN 的最短代价
for  $greedy_{j_i} := 1$  to  $|V| - 1$  do {
    从  $L_{agg}$  中寻找距离组播树上源节点最近的 MSN  $j$  连接到组播树上 MSN  $i \in T_{agg}$ , 即
     $\min(greedy_{cost}(i) + c(i, j)), i \in T_{agg}, j \in L_{agg};$ 
     $greedy_{cost}(j) = greedy_{cost}(i) + c(i, j);$ 
     $T_{agg} = T_{agg} \cup \{j\}, L_{agg} = L_{agg} - \{j\};$ 
     $d_{T_k}(i) ++;$ 
    if  $(d_{T_k}(i) == d_{max}(i))$ 
         $T_{agg} = T_{agg} - \{i\};$ 
    
```

两种算法性能比较结果如表 1 所示。

表 1  $\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$  组播树平均性能比较

性能指标 \ MSN 数	5	10	20	30	50
$delay_{max}(T)$	24.7	23.2	27.5	26.2	29.0
贪 $delay_{max}(T)$	27.0	26.8	30.1	30.6	32.3

由表 1 可以看出, IACSMR 算法所得组播树的性能指标  $delay_{max}(T)$  优于 G-CT 算法。其原因为 IACSMR 算法既采用了  $delay(s, j)$  作为随机贪心启发指标, 又结合了  $delay_{max}(T)$  作为信息的正反馈指标, 而 G-CT 算法只以  $delay(s, j)$  作为贪心启发指标, 因此 IACSMR 算法所得解的  $delay_{max}(T)$  性能优于 G-CT 算法。上述结果是针对平均性能而言, 对于一次组播路由计算而言, 有时 G-CT 算法所得到的组播树  $delay_{max}(T)$  和 IACSMR 算法一样优化。

当  $\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 0$  时, MOSSMRM 模型简化为求解节点负载均衡问题。本文将 IACSMR 算法和一种贪心算法 G-BCT 进行性能比较, G-BCT 参考了 Shi 提出解决 LDRB 模型的 BCT 算法<sup>[6,7]</sup>, 具体步骤不再阐述。性能比较结果如表 2 所示。

表 2  $\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 0$  组播树平均性能比较

性能指标 \ MSN 数	5	10	20	30	50
$D_x(T)$	0.15	0.213	0.883	1.427	3.038
贪 $D_x(T)$	0.15	0.213	0.271	0.360	0.494

由表 2 可以看出, IACSMR 算法所得组播树的负载均衡指标比 G-BCT 算法低, 原因是在度约束范围较小的情况下, G-BCT 算法得到组播树负载均衡指标近似于最优。而由于 IACSMR 算法基于概率, 解空间较大时难以收敛到最优解, 一般只得到较优解。

当  $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 1$  时, MOSSMRM 模型简化为带度约束的最小生成树问题。本文将 IACSMR 算法和带度约束的 SPH 算法<sup>[11]</sup>进行性能比较, 性能比较结果如表 3 所示。

表 3  $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 1$  组播树平均性能比较

性能指标 \ MSN 数	5	10	20	30	50
$\cos t(T)$	39.2	68.5	117.4	164.5	255.1
贪 $\cos t(T)$	39.2	69.7	119.2	168.2	260.0

由表 3 可以看出, IACSMR 算法所得组播树的整体延迟性能比带度约束的 SPH 算法优化, 其原因为 IACSMR 算法既采用了与 SPH 算法类似的贪心启发策略, 又结合了生成树代价进行信息素正反馈。

由上述分析可知, IACSMR 算法在只考虑一个优化指标时解的该指标性能是优化的。根据应用的具体需求, 并考虑消除各目标函数值数量级的差别, 赋予  $\lambda_1, \lambda_2, \lambda_3$  不同的值, 使用 IACSMR 算法求解, 则可以得到一个综合性能优化的解。

### 5.2 算法的复杂度讨论

本文提出的 IACSMR 算法时间复杂度为  $O(NC_{max} \cdot m \cdot n^3)$ 。当  $n$  较小时, 能以较快的速度找到性能较好的优化解; 当  $n$  增大时, 一般需要相应地增大迭代次数  $NC_{max}$  和蚂蚁数目  $m$ , 以便寻找全局最优解, 但这却降低了算法的收敛速度。另一方面, 算法中的参数  $\rho, Q, \alpha, \beta$  的取值也影响着算法的取

值空间和收敛性。有许多文章对解决寻找全局最优解和提高收敛速度之间的矛盾进行了研究,可以借鉴这些文章提出的方法对 IACSMR 算法进行改良。

**结论** 针对网络电视、视频点播类型的大规模实时流媒体应用,本文提出一个两层应用层组播框架结构 ALMF,并对其中 MSN 覆盖网络的构建和维护进行了研究,提出了一个 MSN 覆盖网络路由由组播模型 MOSSMRM 和相应的基于智能蚁群的路由算法 IACSMR。仿真实验表明,IACSMR 算法是一个有效的应用层组播路由算法。我们下一步的努力方向是基于 ALMF 框架构建一个具体的组播应用,在该应用中实现 MOSSMRM 模型和 IACSMR 算法并在真实的互联网环境中进行测试。

**参考文献**

- 1 Yeo C K, Lee B S. A survey of application level multicast techniques[J]. Computer Communications,2004,1547~1568
- 2 Zhang Beichuan, Sugih Jamin. Host Multicast: A Framework for Delivering Multicast to End Users[C]. In: Proceedings of IEEE

- INFOCOM, 2002. 1366~1375
- 3 Pendakaris D, Shi S. ALMI: an application level multicast infrastructure[C]. In: Anderson T, ed. The 3rd USENIX Symposium on Internet Technologies and Systems. San Francisco, CA, USA: USENIX Association, 2001. 49~60
- 4 Chu Yang-hua, Rao S G. A Case for End System Multicast[J]. Selected Areas in Communications, 2002, 20(8): 1456~1471
- 5 Luo Chong, Li Jiang. DigiMetro - An Application-Level Multicast System for Multi-party Video Conferencing[C]. In: IEEE Globecom, 2004
- 6 Shi S, Turner J. Dimension Server Access Bandwidth and Multicast Routing in Overlay Networks[C]. In: 11th International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV01), June 2001
- 7 Shi S, Turner J. Routing in Overlay Multicast networks[C]. In: Proceedings of IEEE INFOCOM, 2002
- 8 潘耘,余镇危,等. 求解应用层组播路由问题的遗传算法[J]. 小型微型计算机系统, 2005, 26(1): 55~58
- 9 Hosseini M, Georganas N D. Design of a multi-sender 3D video-conferencing application over an end system multicast protocol [C]. In: Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA, November 2003
- 10 吴启迪,汪毓. 智能蚁群算法及应用[M]. 上海科技教育出版社, 2004. 4
- 11 Bauer F, Varma A. Degree-Constrained Multicasting in Point-to-Point Networks[C]. In: IEEE INFOCOM'95, 369~376

(上接第 33 页)

在一段时间内成功的路由请求占总组播请求的比率,称为请求成功率。三种算法请求成功率的比较结果如图 1 所示。可以看出,运行 F 算法和 G 算法得到的组播路由树对网络状态不精确有着更好的适应性, F 算法的请求成功率大于另外两种算法。P 算法由于没有网络状态不精确问题而体现出较差的性能。

三种算法所求得的组播树满足用户 QoS 约束的可信度的比较结果如图 2。仿真结果表明, F 算法的满足用户 QoS 约束的可信度明显优于另外两种算法,并且随着网络拓扑的复杂化,优化目标值没有明显的波动,说明算法的扩展性很好。G 算法仅仅考虑了 QoS 中延迟一个参数,而 P 算法没有考虑 QoS 参数,所以结果不理想。



图 1 请求成功率

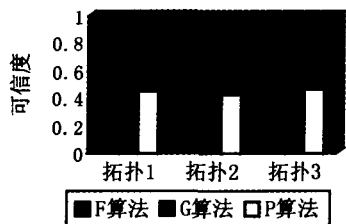


图 2 满足用户 QoS 约束的可信度

以拓扑 2 为例并以 P 算法为基准,比较三种算法的用户效用和网络方效用,结果如表 2 所示。相较而言, F 算法能更好地保证用户和网络方效用的最大化,这是因为该算法引入了博弈理论和动态 Pareto 最优解集,更好的保证了多个优化目标的同时最大化。

**结束语** 本文在充分考虑了 NGI 网络状态不精确和用

户柔性 QoS 需求的前提下,设计了一种基于 FCA 的 QoS 组播路由算法。引入模糊数学的理论适应网络状态不精确的需要,采用用户效用和网络方效用的博弈建立选路原则,利用 FCA 的寻优能力,找出在给定条件下用户效用、网络方效用和满足用户 QoS 需求可信度同时达到最大的组播路由树。仿真结果证明了该算法的可行性和有效性。

表 2 用户效用和网络方效用的比较

组播节点比例	F 算法 : G 算法 : P 算法	
	用户效用	网络方效用
16.2%	1.206 : 1.206 : 1	1.183 : 1.183 : 1
32.4%	1.303 : 1.225 : 1	1.211 : 1.15 : 1
48.6%	1.215 : 1.18 : 1	1.253 : 1.126 : 1
64.8%	1.253 : 1.197 : 1	1.149 : 1.117 : 1
81.0%	1.174 : 1.129 : 1	1.24 : 1.203 : 1
97.3%	1.28 : 1.209 : 1	1.177 : 1.114 : 1

**参考文献**

- 1 Wang Zheng, Jon C. Quality of Service for Supporting Multimedia application. IEEE journal on selected areas in communication, 1996, 14(7): 1228~1234
- 2 Charikar M, Naor J, Schieber B. Resource Optimization in QoS Multicast Routing of Real-Time Multimedia. IEEE/ACM transaction on networking, 2004, 12(2): 340~348
- 3 Moussaoui O, Ksentini A, Naimi M, Gueroui A. Multicast Tree Construction with QoS Guaranties. Springer LNCS 3754, 2005. 96~108
- 4 Korkmaz T, Krunz M. Bandwidth-Delay Constrained Path Selection Under Inaccurate State Information. IEEE/ACM transaction on networking, 2003, 11(3): 384~398
- 5 Zappala D. Alternate Path Routing for Multicast. IEEE/ACM transaction on networking, 2004, 12(1): 30~43
- 6 koyama A, Nishie T, Arai J, Barolli L. A New Quality of Service Multicast Routing Protocol Based on Genetic Algorithm. IEEE ICPADS'05, 2005. 1~6
- 7 Wang Junwei, Wang Xingwei, Huang Min. A Hybrid Intelligent QoS Multicast Routing Algorithm in NGI. IEEE PDCAT', 2005. 723~727
- 8 喻海飞,汪定伟. 食物链算法及其在供应链中的应用. 系统仿真学报, 2005, 17(5): 1195~1199
- 9 张品, 李民乐, 王晟. 模糊参数下多播 QoS 路由及分解. 计算机学报, 2006, 29(2): 279~285
- 10 谢识予. 经济博弈论. 上海: 复旦大学出版社, 2002
- 11 Ping C, Tianling D. A fuzzy genetic algorithm for QoS multicast routing. Computer communications, 2003, 26: 506~512