覆盖多播路由的算法及协议研究综述*)

吴家皋
(南京邮电大学计算机学院 南京 210003)
(计算机网络和信息集成教育部重点实验室(东南大学) 南京 210096)

摘 要 虽然IP多播的性能优势无可否认,但是它却面临着部署上的困难。近年来,覆盖多播作为提供多播服务的另一可行途径正不断为人们所认可。本文对覆盖多播路由的算法和协议进行了综述研究,提出了通用的覆盖多播网络模型,对覆盖多播中的路由优化问题进行了分类。在此基础上,介绍了当前重要的覆盖多播路由算法和协议,并对它们的性能参数、所解决的路由问题、系统结构和控制方式等技术特点进行了全面的分析和讨论。另外,本文还指出了覆盖多播路由中一些有待进一步研究的问题。

关键词 多播,覆盖多播,路由问题,算法,协议

A Survey of Routing Algorithms and Protocols in Overlay Multicast

WU Jia-Gao[1,2]
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210096)[1]
(Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096)[2]

Abstract Though the performance advantage of IP multicast is undeniable, the deployment of IP multicast has met many difficulties. In recent years, overlay multicast is being increasingly recognized as a viable alternative. In this paper, an overview of overlay multicast routing algorithms and protocols were provided. A general overlay multicast network model was proposed, and the routing optimal problems in overlay multicast were identified and classified. Based on above work, some recent important overlay multicast routing algorithms and protocols were described, and a set of properties of them, including performance metric, routing problem solved, system structure and control method etc., were fully analyzed and discussed. In addition, some further research problems in overlay multicast routing were also pointed out.

Keywords Multicast, Overlay multicast, Routing problem, Algorithm, Protocol

1 引言

随着以网络视频会议、视频点播、远程教育、多方在线游戏和虚拟现实等为代表的新型应用的大量涌现,人们对多播通信服务提出了迫切的需求。然而,由于技术和经济的双重原因[1],直到目前,全互联网范围内的IP多播服务尚未部署完成。为此,人们转而希望在应用层解决IP多播面临的问题,提出了覆盖多播(Overlay Multicast)的概念[2]。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(如组管理、成员管理、分组复制和转发等),下层网络只需提供基本的端到端单播传输服务。因此,覆盖多播也被称为应用层多播或端系统多播。覆盖多播最大的优势在于无需改变现有的IP网络基础设施,易于部署。另外,利用端系统灵活的计算能力,覆盖多播还易于实现可靠、安全和服务定制等高层功能。但是,由于覆盖多播网络是在下层IP网络之上构建的虚拟的、逻辑的网络,这就可能在IP层形成重复分组和重复路径,从而增加覆盖多播路由的传输延时和占用带宽。图1给出了IP多播和覆盖多播路由的比较示意图,其中端系统A为数据源,B,C,D为目的。从图中可以看出,对于IP多播,每个数据分组在物理链路上只转发一次,且能沿最短路径到达目的;而对于覆盖多播,则在物理路径A-1-3上出现了重复分组(增加了带宽占用量),在物理路径3-2-B上形成了重复路径(增加了D的接收延时)。因此,覆盖多播路由的性能通常不及IP多播。这一问题对于大多数新型应用,特别是对于延时和带宽敏感的实时多媒体应用,就显得尤为突出。如何构造满足各种应用需求的覆盖多播路由是当前覆盖多播研究的热点之一。

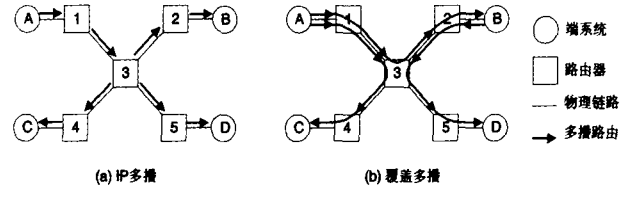

图1 IP多播和覆盖多播路由比较示意图

对这一新兴研究方向已有及时的分析和总结,无疑对掌握研究动态、洞察问题本质、发现新的研究方向具有重要的指导意义。目前已报道的有关覆盖多播的研究综述[2,3],主要从不同的技术方案层面对当前的覆盖多播路由算法及协议进行分类和比较,对覆盖多播网络模型还缺乏统一的指导。同时,过于注重技术细节,将不利于全局把握研究状况。与以上

*)本文受到国家自然科学基金重大研究计划项目(90604003)和南京邮电大学攀登计划项目(NY206009)资助。吴家皋 博士,助理研究员。

思路不同，本文以覆盖多播的网络模型基础，覆盖多播路由算法和协议的性能优化目标（延时、带宽等）为切入点，试图揭示这些算法和协议背后蕴含的科学问题，从问题域的高度，对覆盖多播路由的研究有一个新的认识。

在本文中，我们首先提出了通用的覆盖多播网络模型，并对覆盖多播路由问题进行了分类，指出了其不同于 IP 多播的新特点。在此基础上，我们介绍了当前重要的代表性覆盖多播路由算法和协议，并对其技术特点进行了全面的分析和讨论。最后，本文讨论了一些有待解决的问题并指出了将来的研究方向。

2 网络模型及路由问题

假设下层 IP 网络能向端系统提供透明的端到端单播路由，则覆盖多播的网络模型可用一个完全有向图 $G=(V, E)$ 来描述，其中 V 是节点的集合，表示端系统；$E=V\times V$ 是边的集合，表示虚拟链路，每一虚拟链路对应于一条下层 IP 网络的物理路径。设 R^+ 为正实数集合，R^* 为非负实数集合，则 $\forall v\in V$，可定义下列属性参数：输入带宽能力函数 $B_{in}(v): V\rightarrow R^+$，表示节点最大的接收带宽；输出带宽能力函数 $B_{out}(v): V\rightarrow R^+$，表示节点最大的转发带宽；代价函数 $C_v(v): V\rightarrow R^+$，表示节点的代价，如端系统设备的费用等。$\forall e\in E$，可定义如下属性参数：带宽容量函数 $B_c(e): E\rightarrow R^+$，表示边上两节点之间可获得的最大带宽，它由该边上的端系统节点的输入、输出带宽能力及其对应的下层物理路径的瓶颈带宽决定，即，$\forall e=(u, v)$，有 $B_c(u, v)=\min(B_{out}(u), bn(u, v), B_{in}(v))$，其中，$bn(u, v)$ 为边 (u, v) 对应的物理路径的瓶颈带宽；延时函数 $D_e(e): E\rightarrow R^+$，表示边上两节点之间的端到端分组传输延时；抖动函数 $J_e(e): E\rightarrow R^*$，表示边上两节点之间的端到端分组传输延时抖动；分组丢失率函数 $P_e(e): E\rightarrow R^*$，$P_e(e)\in[0,1)$，表示边上两节点之间的端到端分组传输丢失率；代价函数 $C_e(e): E\rightarrow R^+$，表示边的代价，如网络资源占用量等。

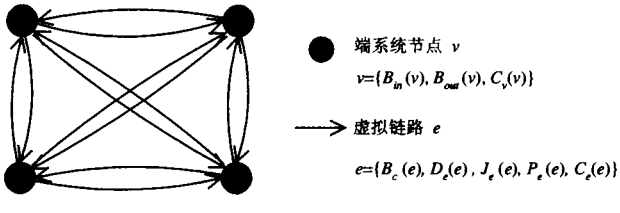

图 2 覆盖多播网络模型示意图

图 2 给出了一个覆盖多播网络模型的示意图，该模型由 4 个端系统节点组成，它们之间通过 12 条有向边（虚拟链路）构成一完全有向图，对于每个端系统节点和每条虚拟链路分别定义相应的属性参数。与 IP 网络模型不同，这里的边是虚拟链路，它可能与其他边共享端系统节点和下层物理链路的带宽，将增加覆盖多播网络中带宽资源管理的复杂性。但是，考虑到端系统的接入带宽和 CPU 处理能力的限制，当端系统成为虚拟链路的带宽瓶颈时，B_c 将由端系统的带宽能力属性决定，这将简化上述模型。进一步，对于媒体流的平均带宽为 b 的多播组，若 $\forall v\in V$，$b\leqslant B_{in}(v)$，则节点 v 的输出能力带宽可用“度”表示，定义度约束属性 $d_{\max}(v)=\lfloor B_{out}(v)/b\rfloor$，即节点 v 最多只能同时向 $d_{\max}(v)$ 个下游节点转发分组。此时，覆盖多播网络就可用度约束模型来描述。但是，在上述简化条件不成立时，多播应用通常采用端到端的测量来估计虚拟链路的可用带宽。因此，本文下面将从端系统的角度来讨论和描述带宽参数。

覆盖多播路由问题的一般形式描述如下：给定一个完全有向图 $G=(V, E)$，一个源（或根）节点 $s\in V$，以及目的节点的集合 $V-\{s\}$。对于一组约束条件 C 和一个可能优化目标 O，构造一棵 G 的最优生成树，使其满足 C。

根据上述模型，我们发现覆盖多播路由问题的优化目标和约束条件均可分为 2 类：节点优化和树优化、节点约束和树约束。其中，节点优化（约束）针对节点相关的属性参数，如节点的带宽或度、节点的代价等；树优化（约束）针对与多播树相关的性能参数，如树的延时直径（或半径）、树的平均延时、树的代价等。将这些约束条件和优化目标进行组合，可形成下面各种覆盖多播路由问题。

（1）节点约束：具有节点属性约束的多播路由问题，如度（带宽）约束的多播树等。

（2）树约束：具有树性能参数约束的多播路由问题，如延时约束的多播树等。

（3）多树约束：具有多个树约束的多播路由问题，如延时和代价约束的多播树等。

（4）节点和树约束：具有节点和树约束的多播路由问题，如度（带宽）和延时约束多播树等。

（5）节点优化：节点属性优化的多播路由问题，如最小（或平均）带宽、最大多播树等。

（6）树约束节点优化：具有树约束且节点属性优化的多播路由问题，如延时约束的剩余度（带宽）、平衡多播树等。

（7）树优化：树性能参数优化的多播路由问题，如延时（或代价）最小多播树等。

（8）节点约束树优化：具有节点约束且树优化的多播路由问题，如度（带宽）约束的延时（或代价）最小多播树等。

（9）树约束树优化：具有树约束且树优化的多播路由问题，如延时约束的代价最小多播树等。

（10）节点和树约束树优化：具有节点和树约束且树优化的多播路由问题，如度（带宽）和延时约束的代价最小多播树等。

问题（1）容易在多项式时间内解决，如对于度约束多播树的构造，可从源节点开始，每次选择任一未成树的节点加入到树上任意有剩余度的节点，直到所有节点都加入树为止。问题（7）可分为延时半径（源到最远节点的延时）最小树、延时直径（最远两节点之间延时）最小树、代价最小树等问题。不难发现，从源节点到所有目的节点构成的星型（Star）树，其延时半径最小；Hassin 等[4]则提出了在多项式时间内寻找延时直径最小树的算法；而代价最小生成树可以采用 Prim 算法解决。另外，问题（7）的算法可用于解决问题（2）。所以，问题（2）和（7）都是多项式问题。根据 Wang 等的结论[5]，问题（3）属于多个“相加型”约束组合问题，是 NP 完全的。问题（9）中的延时约束的代价最小生成树问题也是 NP 难问题[6]。Garg 等[7]证明了问题（5）中的最小带宽最大树的构造是 NP 难的，则平均带宽最大树也是 NP 难的。现已证明，度约束的延时（平均延时、半径、直径）最小以及代价最小生成树问题都是 NP 难[8]或 NP 完全的[9,10]。Shi 等[9]还证明了延时约束度平衡生成树问题是 NP 完全问题。由此易知，问题（4）、（6）、（8）、（10）都具有 NP 完全复杂度。覆盖多播路由问题的分类汇总，如表 1 所示。

表 1 覆盖多播路由问题分类表

	无优化	节点优化	树优化
无约束	—	(5)节点优化 NP 完全复杂度	(7)树优化 多项式复杂度
节点约束	(1)节点约束 多项式复杂度	—	(8)节点约束树优化 NP 完全复杂度
树约束	(2)树约束 多项式复杂度	(6)树约束 节点优化 NP 完全复杂度	(9)树约束树优化 NP 完全复杂度
	(3)多树约束 NP 完全复杂度		
节点和 树约束	(4)节点和树约束 NP 完全复杂度	—	(10)节点和树 约束树优化 NP 完全复杂度

在上述问题中，问题(1)过于简单，没有单独研究的价值。问题(2)、(3)、(7)和(9)与节点的属性无关，可以归结为 IP 多播路由问题的特例，不是覆盖多播路由研究的重点。因此，当前覆盖多播路由算法和协议重点解决的问题在于(4)、(5)、(6)、(8)和(10)。

3 路由算法及协议

目前，有关覆盖多播路由算法和协议的研究十分活跃，已提出了一系列解决方案，所涉及的技术非常广，可从不同的角度进行分类。

按系统结构可分为基于主机的(Host-based)和基于代理服务器的(Proxy-based)两类。其中，前者直接由参与多播组的主机(应用)构建多播路由；后者则在主机之间加入多播服务器，由服务器组成核心多播网络，主机通过接入服务器获得多播服务。由于在 Proxy-based 的系统中，主机不参与多播路由的构建，因此上述两种系统结构在解决覆盖多播路由问题时是等价的。

按多播路由的构造方式，可分为网格优先(Mesh-first)和树优先(Tree-first)两类。其中，前者首先由端系统构成一个网状的拓扑结构(Mesh)，然后基于 Mesh 构造多播树；后者则由端系统直接构成一棵多播树，而不依赖于 Mesh。

按算法和协议的控制方式，可分为集中式、分布式和层次式 3 类。其中，集中式方法由一控制中心负责覆盖多播网络状态信息的收集和路由的计算；分布式方法将状态信息和算法分布到各个端系统节点中，端系统能自组织地构建多播路由；层次式方法则通过构建多层次的多播路由，分解控制的复杂性。考虑到算法和协议的控制方式对解决覆盖多播的路由问题有着根本的影响，下面我们将从这一视角对当前有代表性的覆盖多播路由算法和协议进行分类介绍。

3.1 集中式

3.1.1 Shi 算法[9]

Shi 等研究了 Proxy-based 构架下的两种覆盖多播路由问题：度约束的延时直径最小生成树(MDDL)和延时直径约束的剩余度平衡生成树(LDRB)问题，并证明它们都是 NP 完全的。对于 MDDL 问题，作者给出了一个启发式贪心算法—压缩树算法(CT)。该算法类似于 Prim 生成树算法，在遵守度约束条件的前提下，逐个选择使当前树直径最小的节点加入树，直到所有节点完成加入为止。CT 算法能在 $O(n^3)$ 时间找到具有 $O(\log n)$ 近似度的解，其中 n 为组大小。对于 LDRB 问题，作者也提出了系列近似算法的算法，但这些算法的基本思想都是从平衡的度分配方案出发，反复进行优化-松弛的叠代过程，其中叠代的压缩树算法(ICT)具有较高的多播会话接纳率。以上算法都需要维护全局的拓扑信息，其状态维护开销为 $O(n^2)$。

另外，针对实时多媒体应用对带宽需求的异构性，我们对 CT 算法进行扩展，采用分层的带宽分配策略，提出了一个异构环境下构造最小延时半径覆盖多播树的集中式算法—分层的压缩树算法(LCT)[11]。

3.1.2 Malouch 算法[12]

Malouch 等研究了在主机(Host)和代理服务器(Proxy)混合构架下的多播路由问题。考虑到 Proxy 比 Host 有更大的转发带宽(可用度)，引入 Proxy 有助于减少多播源到目的节点传输延时。但是，使用 Proxy 也会带来额外的费用。为此，作者提出研究有度和延时半径约束的代价最小生成树问题(代价为 Proxy 的费用)。该问题仍是 NP-hard 的，作者提出的算法也类似于 CT 算法。所不同的是在选择优先加入节点的策略上，该算法同时考虑了延时和度两个因素，并通过调解一启发参数在两者之间进行折衷，从而提高了算法的性能和灵活性。

3.1.3 ALMI 协议[13]

ALMI 是采用集中控制的覆盖多播路由协议。该协议通过一个会话控制器(SC)管理组成员节点的加入或离开。每个节点监测自己到其邻居节点的网络状态信息(如延时等)，并报告给 SC。SC 根据收集到的信息构造多播路由。ALMI 的目标是最小化树的代价。为了减少控制开销，ALMI 通过限制每个节点的邻居数(度)，只维护局部的网络拓扑，从而使其状态维护开销在 $O(n)$ 量级。在此基础上，ALMI 运行最小生成树算法，构造具有度约束的代价最小的多播共享树。虽然在初始阶段，算法会因为只有局部信息而导致路由性能下降，但是通过动态邻居优化选择和数据汇聚，路由性能将随时间逐步改善。

3.1.4 Garg 算法[7]

Garg 等研究了覆盖多播路由中的最小带宽最大化问题。在一般物理网络拓扑下，即使完全知道下层物理链路带宽，当通过单一路径(即一棵多播树)从源节点向目的节点分发数据时，该问题也是 NP 难的。为此，作者提出了一个近似度为 2 的集中式算法，其主要步骤是：①寻找能保证所有端系统节点连通的最大链路带宽 h；②以带宽不小于 h 的链路构造包括所有端系统节点的生成树 S；③将 S 上的任一链路以一对 1/2 带宽的相反链路代替，则能构造 S 的一欧拉路径，且其带宽为 $h/2$。该算法需要全局的物理网络拓扑信息，其状态维护开销为 $O(n'+e')$，其中 n' 为端系统及路由器节点的总和，e' 为物理链路总数。

3.2 分布式

3.2.1 Narada 协议[14]

Narada 协议是最早提出覆盖多播路由协议之一，它采用 Mesh-first 路由构造方案。在该协议中，所有组成员节点将构成一虚拟网络拓扑(Mesh)。当新节点加入时，它首先通过汇聚点获得组成员列表，然后从中随机选择邻居节点，请求加入 Mesh。在 Mesh 之上，Narada 运行类似于 DVMRP 的多播路由协议，采用反向最短路径(RSP)算法进行数据转发。Mesh-first 方案的优点是不同的数据源可以共享 Mesh，易于构造有多个源的多播树。Narada 能按不同应用的需求，采用不同的参数多播路由。当用延时为参数并限制节点在 Mesh 上的邻居数时，Narada 能构造有度约束平均延时最小多播

树。另外，面向网络视频会议应用，Chu 等[15]还提出最短的最宽多播树算法，其思想类似文[5]，首先优化带宽，当有多条带宽相同的路径时，选择延时最短的。

但是，Narada 要求每个节点维护全部组成员的状态信息，其状态维护开销达到了 $O(n)$，因此它只适合小规模的多播应用。

3.2.2 Gossamer 协议[16]

Gossamer 协议是 Proxy-based 覆盖多播系统 Scattercast 的核心路由协议。Gossamer 也属于 Mesh-first 类协议，其路由机制与 Narada 类似。所不同的是，在组成员发现方面，Gossamer 采用聊天式(gossip-style)协议，通过组成员间随机的报文交互，获得其他组成员的信息，而 Narada 则通过汇聚点获得全部组成员列表；在多播路由性能参数方面，Gossamer 只考虑延时参数，构造有度约束的平均延时最小多播树；而 Narada 则同时考虑了带宽和延时两个参数。

3.2.3 Yoid 协议[17]

Yoid 是早期提出的 Tree-first 覆盖多播路由协议，它采用有度约束的共享树方案。新成员节点通过汇聚点获得部分树上节点信息，并从中随机选择一满足度约束条件的节点作为父节点，加入多播树。然后，树上节点将动态发现并改变当前父节点，以提高路由性能。为了避免因多播树动态变化而产生回路(Loop)，每个节点都保持一个从根节点到自己树上路径的节点序列，称为根路径(root path)。协议可以根据 root path 中是否有重复节点，来发现是否有 Loop 发生。另外，每个节点还维护一个独立于多播树的 Mesh 结构，以提高路由的鲁棒性。在 Yoid 中，每个节点只需维护树上邻居节点的状态信息，考虑到 Mesh 的开销，其总的状态维护开销为 $O(d+m)$，其中 d 为节点最大的孩子节点数(度约束值)，m 为节点连接 Mesh 的最大边数。这与节点总数无关，因而大大提高协议的可扩展性。Yoid 是一个通用的覆盖多播协议，应用可以根据自己需求，利用 Yoid 提供的协议机制实现不同的优化目标。Yoid 中的许多基本协议机制，如局部状态维护、动态优化和 root path 等为后来的大多数 Tree-first 路由协议和算法所借鉴和采用。

3.2.4 Overcast 协议[18]

Overcast 协议主要是面向网络内容分发应用提出的，它采用 Proxy-based 结构，其优化目标是使从数据源节点到所有目的节点的平均带宽最大化。该协议采用分布式 Tree-first 路由方案，新节点初始都接入到源节点(树根)，然后采用动态优化策略调整其在树上的位置。Overcast 的优化原则是节点在不减少其从源获得的带宽的前提下，尽量远离源节点，以减少对源节点带宽的占用，为其他节点保留更多的可用带宽。

另外，Kim 等也研究了平均带宽最大化覆盖多播树的构建问题，并提出了 Kim 协议[19]。与 Overcast 协议的局部优化算法不同，Kim 协议通过对网络模型进行抽象和假设(边缘带宽假设、公平贡献假设)能保证其带宽优化算法具有全局最优性。Kim 协议中，每个节点维护其祖先节点状态的开销为 $O(\log n)$。

3.2.5 Banerjee 算法[8]

Banerjee 等研究了 Proxy-based 覆盖多播系统中的核心路由协议，也提出了两种 NP-hard 问题：度约束的平均延时最小生成树和度约束的延时半径最小生成树。所不同的是，这里的平均延时引入了客户数加权因子，使路由优化目标能更合理地反映客户的整体情况。针对上述问题，作者提出了一个分布式路由算法。该算法分为 3 个步骤：初始优化、局部优化和随机优化。其中，局部优化是该协议的核心，即在多播树上两跳(层)范围以内，通过树上节点的移动和交换操作来实现路由性能优化，主要包括：孩子节点提升、父子节点交换、2 层内节点交换/移动、1-2 层内节点交换等。同 Yoid 协议一样，Banerjee 算法也只维护局部状态，并采用 root path 避免路由回路。

3.2.6 Switch-Tree 协议[20]

Switch-Tree 协议的基本思想是，所有新加入的节点首先连接到源节点(根)，然后节点动态选择新的父节点，以优化路由性能。当需要优化多播树的代价时，节点倾向于选择比当前父节点距离更近的节点；当需要优化多播树的平均延时时，节点倾向于选择能减少其在树上的延时的节点。考虑对节点度的限制，Switch-Tree 能为构造有度约束的平均延时(或代价)最小生成树提供解决方案。该协议提出了一种范围可变的动态优化模式，包括：邻居交换、1 跳范围交换、2 跳范围交换和任意跳范围交换等。研究发现，随着交换范围的扩大，多播树的性能将有所提高，但协议的状态维护开销却呈指数增大，为 $O(d^h)$，其中 h 为交换范围的跳数。一般认为，1 或 2 跳范围交换具有较高的性价比。

3.2.7 HMTP 协议[21]

主机多播树协议(HMTP)也是 Tree-first 的。该协议的目标是构造度约束的代价最小共享树，其核心包括：节点加入算法和动态优化算法。加入算法的思想是按深度优先的方式搜索树上的局部最优的加入点，其基本过程是：初始令根节点为当前试探父节点(p_t)，新加入节点首先查询并探测 p_t 及其孩子节点，从中选择延时距离最近的节点。若该节点为 p_t，则向其发出加入请求，否则令该节点为新的 p_t，并重复前面的过程。为进一步提高路由性能，HMTP 同样采用动态优化算法，即每个节点的 root path 上随机选择一节点，令其为 PP，重新运行加入算法。

3.2.8 Hostcast 协议[22]

Hostcast 协议的优化目标也是最小化树的平均延时，其加入算法和动态优化机制与 HMTP 协议基本类似。但是，与其他 Tree-first 协议不同的是，该协议在构造 Tree 的同时，也构造 Mesh。Tree 用于数据传输，Mesh 用于 QoS 参数的测量和协议控制。由于每个节点要求维护 2 跳范围内的 Mesh，Hostcast 的状态维护开销为 $O(d^2)$。

3.2.9 TBCP 协议[23]

树构建控制协议(TBCP)是又一个 Tree-first 协议，其独特之处在于将组合优化方法引入节点加入算法中。具体方法是：潜在父节点将新加入的节点在自己的孩子节点中进行局部优化组合，以寻找代价(或延时)最小的组合方案。当该方案满足度约束时，新节点加入成功；否则进行新一轮局部优化组合。

3.3 层次式

3.3.1 NICE 协议[24]

NICE 协议是一个自组织的层次化覆盖多播路由协议，面向大规模 Host-based 多播应用。该协议的思想是将整个多播树分层几个层次，从底向上依次为第 0，1，2，…层，每层组成员节点分成多个簇(Cluster)。每个 Cluster 只能包含[k，$3k-1$]个节点，其中 k 为 Cluster 大小常数。每个节点都属于第 0 层，从最底层开始，每个 Cluster 选出一个节点作为领导节点(leader)，参与上一层 Cluster 的构造，直到选出最高的

leader为止。新节点的加入过程，从最高层leader开始，不断向下层寻找距离最近的leader，直到第0层完成加入。Cluster内成员采用星型结构连接到leader(延时最小)。NICE协议最大的状态维护开销为$O(k\log_k n)$，平均开销为$O(k)$。

3.3.2 ZigZag协议[25]

ZigZag协议与NICE类似。所不同的是ZigZag采用外领导节点(forgein head)，作为下层节点的父节点，构成了一棵"锯齿型"层次树。这样，就减轻了NICE中leader节点的失效而造成的路由分裂问题。ZigZag协议在状态维护方面，平均开销与NICE相当，但其最大开销为$O(\log_k n)$，优于NICE。

3.4 其他协议

覆盖多播路由协议和算法的种类很多，本文只是面向覆盖多播路由优化问题，涉及了其中一部分。为了提高协议的可扩展性，支持大规模多播应用，目前已提出了一类基于结构化P2P网络的覆盖多播路由协议[26~28]。但这类协议采用的是逻辑标识路由，不能直接映射为物理网络上的QoS性能参数。另外，考虑到因端系统失效而造成的多播树分裂和数据丢失问题，HMB协议[29]提出在多播树外增加冗余虚链路(RVL)，数据同时在树和RVL上转发，以提高多播树的鲁棒性。TAG协议[30]则利用下层IP网络拓扑信息辅助构造多播树，以提高路由性能。

4 比较与分析

上述覆盖多播路由算法及协议的分类比较列表，如表2所示。

表2 覆盖多播路由算法和协议的分类比较表

协议算法	性能参数	路由问题	系统结构	控制方式	建树方式	树类型	状态维护
Shi	度约束，延时直径最小	(8)	Proxy-based	集中	Tree-first	共享的	$O(n^2)$
	延时直径约束，度平衡	(6)					
LCT	带宽约束，延时半径最小	(8)	Proxy-based	集中	Tree-first	基于源的	$O(n^2)$
Malouch	度和延时半径约束，代价最小	(10)	Host和Proxy-based	集中	Tree-first	基于源的	$O(n^2)$
ALMI	度约束，代价最小	(8)	Host-based	集中	Tree-first	共享的	$O(n)$
Garg	最小带宽最大	(5)	Host-based	集中	Tree-first	基于源的	$O(n'+e')$
Narada	度约束，平均延时最小	(8)	Host-based	分布	Mesh-first	基于源的	$O(n)$
	带宽最大(延时最小的)	(5)					
Gossamar	度约束，平均延时最小	(8)	Proxy-based	分布	Mesh-first	基于源的	$O(n)$
Yoid	度约束，通用优化目标	(8)	Host -based	分布	Tree-first	共享的	$O(d+m)$
Overcast	平均带宽最大	(5)	Proxy-based	分布	Tree-first	基于源的	$O(d)$
Banerjee	度约束，平均延时(或延时半径)最小	(8)	Proxy-based	分布	Tree-first	基于源的	$O(d)$
Switch-tree	度约束，代价(延时)最小	(8)	Host-based	分布	Tree-first	基于源的	$O(d^h)$
HMTP	度约束，代价最小	(8)	Host-based	分布	Tree-first	共享的	$O(d)$
HostCast	度约束，平均延时最小	(8)	Host-based	分布	Tree-first	基于源的	$O(d^2)$
TBCP	度约束，代价(延时)最小	(8)	Host-based	分布	Tree-first	基于源的	$O(d)$
Kim	平均带宽最大	(5)	Host-based	分布	Tree-first	基于源的	$O(\log n)$
NICE	度约束，延时最小	(8)	Host-based	层次	Tree-first	共享的	最大：$O(k\log_k n)$ 平均：$O(k)$
ZIGZIG	度约束，延时最小	(8)	Host-based	层次	Tree-first	基于源的	最大：$O(\log_k n)$ 平均：$O(k)$

由表2我们发现，不同的覆盖多播路由协议和算法所采用的技术方案不尽相同，它们在系统结构、控制方式、建树方式、树的类型、性能参数以及所解决的路由问题等方面都有着各自的特点。

在系统结构方面，由于代理服务器比用户主机具有更高的处理能力和稳定性，因此Proxy-based结构更适合构建通用的多播服务网络。但是，Proxy-based系统需要ISP的支持，其灵活性不如Host-based系统。

在协议和算法的控制方式方面，集中式方法通常能生成较高性能的多播路由，但它需要维护全局状态信息，可扩展性差；分布式方法只维护局部状态信息，可扩展性好，但这会影响路由算法的性能；层次式方法则在性能和开销之间进行折衷，不过它也会引入路由管理的复杂性。

在建树方式方面，Mesh-first方式有利于多播路由的重用和增加路由的鲁棒性，但维护Mesh本身需要额外的控制开销；而Tree-first方式则非常简单，能直接控制多播路由的QoS性能参数，但其路由的鲁棒性不如Mesh-first方式。

在树的类型方面，共享树适合有多个数据源的多播组，能减少对网络资源的占用；而基于源的树则更适合针对某个数据源优化其多播路由的性能。

在所解决的路由问题方面，与问题(8)相关的协议和算法最多，因为该类问题最直接地反映了覆盖多播的特征。而其他问题(如问题(4)、(10)等)的研究还有待加强和深入。

综上所述，在覆盖多播路由算法和协议的研究中，没有哪一种技术方案具有绝对优势，应该根据实际应用环境，选择具体的技术路线。

结束语 本文对覆盖多播路由算法及协议进行了综述研究。从路由优化问题的角度，提出了通用的覆盖多播网络模型，并对覆盖多播路由问题进行了分类。根据上述路由优化问题，介绍了当前一些重要的覆盖多播路由算法和协议，并对它们的性能参数、所解决的路由问题、系统结构和协议机制等技术特点进行了全面的分析和讨论。需要指出的是，覆盖多播路由是一个新的研究领域，还存在许多问题，有待将来解决：首先，性能和开销这对矛盾仍是覆盖多播路由研究中需要

面对的难题，为此应该对现有的各种覆盖多播路由协议和算法进行全面的、定量的比较研究，以便在性能和开销之间寻找最佳平衡点；其次，目前的覆盖多播路由协议和算法所涉及的路由性能参数相对比较简单，还缺乏对多 QoS 约束的覆盖多播路由问题的研究。特别是，当考虑到系统的异构性时，上述问题将更为复杂；最后，覆盖多播路由的可用性问题（例如，如何适应网络环境的动态变化、如何快速修复路由分裂以及防范恶意攻击等）也同样需要重视并进一步研究。

参 考 文 献

1 Diot C, Levine BN, Lyles B, et al. Deployment Issues for the IP Multicast Service and Architecture [J]. IEEE Network, 2000, 14 (1): 78～88

2 El-Sayed A, Roca V. A Survey of Proposals for an Alternative Group Communication Service [J]. IEEE Network, Jan/Feb 2003. 2～7

3 Banerjee S, Bhattacharjee B. A Comparative Study of Application Layer Multicast Protocols [A]. In: submitted for review, 2002

4 Hassin R, Tamir A. On the minimum diameter spanning tree problem [J]. Inform Processing Lett, 1995, 53: 109～111

5 Wang Z, Crowcroft J. QoS Routing for Supporting Resource Reservation [J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228～1234

6 Salama H F, Reeves D S, Viniotis Y. The Delay-constrained Minimum Spanning Tree Problem [A]. In: Proceedings of the International Symposium on Computers and Communications (ISCC '97) [C], June 1997

7 Garg N, Khandekar R, Kunal K, et al. Bandwidth Maximization in Multicasting. In: Proceedings of the 11th Annual European Symposium on Algorithms, September 2003. 242～253

8 Banerjee S, Kommareddy C, Kar K, et al. Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications [A]. In: Proceedings of the IEEE INFOCOM [C]. San Franciso, 2002. 1521～1531

9 Shi S, Turner J. Multicast Routing and Bandwidth Dimensioning in overlay networks [J]. IEEE Journal on Selected Areas in Communications, 2002, 20(8): 1444～1455

10 Bauer F, Verma A. Degree-constrained multicasting in point-to-point networks [A]. In: Proceedings of IEEE Infocom'95 [C], 1995. 369～376

11 吴家皋，叶晓国，姜爱全. 一种异构环境下覆盖多播网络路由算法. 软件学报，2005，16(6)：1112～1120

12 Malouch N M, Liu Z, Rubenstein D, et al. A Graph Theoretic Approach to Bounding Delay in Proxy-Assisted, End-System Multicast [A]. In: Proceedings of ACM NOSSDAV' 02 [C], 2002

13 Pendarakis D, Shi S, Verma D, et al. ALMI: An Application Level Multicast Infrastructure [A]. In: Proceedings of the 3rd USENIX Symposium on Internet Technologies & Systems [C], San Francisco, 2001. 49～60

14 Chu YH, Rao SG, Zhang H. A Case for End System Multicast [A]. In: Proceedings of the ACM SIGMETRICS [C]. Santa Clara, 2002. 1～12

15 Chu Y H, Rao S G, Seshan S, et al. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture [A]. In: Proceedings of ACM SIGCOMM [C], 2001. 55～67

16 Chawathe Y. Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service: [Ph D Thesis]. University of California, Berkeley, 2000

17 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/

18 Jannotti J, Gifford D, Johnson K, et al. Overcast: Reliable Multicasting with an Overlay Network [A]. In: Proceedings of 4th Symposium on Operating Systems Design and Implementation [C], 2000. 197～212

19 Kim M S, Lam S S, Lee D Y. Optimal Distribution Tree for Internet Streaming Media [A]. In: Proceedings of the 23rd IEEE ICDCS [C], May 2003

20 Helder DA, Jamin S. End-host Multicast Communication Using Switch-trees Protocols [A]. In: Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02) [C], 2002. 419～424

21 Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users [A]. In: Proceedings of the IEEE INFOCOM 2002 [C], New York, 2002. 1366～1375

22 Li Z, Mohapatra P. HostCast: A New Overlay Multicasting Protocol [A]. In: Proceedings of IEEE Int. Communications Conference (ICC) [C]. Alaska, 2003. 702～706

23 Mathy L, Canonico R, Hutchison D. An Overlay Tree Building Control Protocol [A]. In: Proceedings of 3rd Int Workshop on Networked Group Communication [C], London, 2001. 78～87

24 Banerjee S, Bhattacharjee B, Kommareddy C. Scalable Application Layer Multicast [A]. In: Proceedings of ACM SIGCOMM '02 [C], 2002. 205～217

25 Tran D A, Hua K A, Do T T. ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming [A]. In: Proceedings of IEEE INFOCOM 2003 [C], 2003

26 Druschel P, Castro M, Kermarrec A, et al. Scribe: A large-scale and decentralized application-level multicast infrastructure [J]. IEEE Journal on Selected Areas in Communications, 2002, 20 (8): 1489～1499

27 Ratnasamy S, Handley M, Karp R, et al. Applicationlevel multicast using content-addressable networks [A]. In: Proceedings of 3rd Int. Workshop on Networked Group Communication [C], London, 2001. 14～29

28 Zhuang S Q, Zhao B Y, Joseph A D, et al. Bayeux: An architecture for scalable and fault-tolerant widearea data dissemination [A]. In: Proceedings of Eleventh International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001) [C], 2001. 14～29

29 Roca V, El-sayed A. A host-based Multicast (hbm) Solutionfor Group Communications [A]. In: Proceedings of 1st IEEE Int Conf Networking [C], 2001. 610～619

30 Kwon M, Fahmy S. Topology-Aware Overlay Networks for Group Communication [A]. In: Proceedings of ACM NOSSDAV' 02 [C], 2002. 127～136

（上接第 3 页）

参 考 文 献

1 Weiser M. The computer for the twenty-first century. Scientific American, 1991, 265 (3): 94～104

2 徐光祐，史元春，谢伟凯. 普适计算. 计算机学报，2003，26(9)：1042～1050

3 Coen MH, Phillips B, Warshawsky N, et al. Meeting the computational needs of intelligent environments: the metaglue system. In: Nixon P, Lacey G, Dobson S eds. Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments. Berlin: Springer-Verlag, 1999. 201～212

4 Iachellov G, Abowd G D. Security requirements for environmental sensing technology. In: Proceedings of the 2nd Workshop on Ubicomp Security, Seattle, USA, 2003

5 Ranganathan K. Trustworthy pervasive computing: the hard security problems. In: Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications Workshops. Washington: IEEE Computer Society, 2004. 117～121

6 Lahlou S, Langheinrich M, Röcker C. Privacy and trust issues with invisible computers. Communications of the ACM, 2005, 48 (3): 59～60

7 Creese S, Goldsmith M, Roscoe B, et al. Research directions for trust and security in human-centric computing. In: Proceedings of the First Workshop on Security and Privacy in Pervasive Computing. Vienna, Austria, 2004

8 Satyanarayanan M. Pervasive computing: vision and challenges. IEEE Personal Communication, 2001, 8(4): 10～17

9 Rosenthal L, Stanford V. NIST information technology laboratory pervasive computing initiative. In: Proceedings of the IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. Washington: IEEE Computer Society Press, 2000. 30～36