应用层组播研究进展*)

叶保留 李春洪 姚 键 顾铁成 陈道蓄
(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 组播技术是一种针对多点传输和多方协作应用的组通信模型,有高效的数据传输效率,是下一代 Internet 应用的重要支撑技术。早期的组播技术研究试图在 IP 层提供组播通信功能,但 IP 组播的实施涉及到对现有网络基础设施的调整,因此,大规模应用受到限制。近两年来,随着 Peer-to-Peer(P2P)研究的兴起,基于应用层的组播技术也逐渐受到广泛关注。应用层组播协议将组成员节点自组织成覆盖网络,在主机节点实现组播功能,为数据多点并发传输提供服务。将组播功能从路由器迁移到主机上能有效解决许多与 IP 组播有关的问题,但同时也带来了一些新的挑战。本文分析了目前应用层组播研究的主要内容及技术特点,描述了协议设计所涉及的关键技术及面临的主要挑战,总结了现有工作及相关进展。

关键词 IP 组播,应用层组播,覆盖网络,组播树

State-of-the-art in Application Layer Multicast Research

YE Bao-Liu LI Chun-Hong YAO Jian GU Tie-Cheng CHEN Dao-Xu
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract Multicast provides an efficient way for one-to-many and multi-party communication. It is a key technique for the next-generation Internet applications. The conventional work on multicast emphasizes on providing multicast mechanism in IP protocol layer. However, the deployment of IP multicast is constrained by the modification to the operational Internet infrastructure and is hampered by a lot of unsolved technical problems. In the recent two years, the research trend on Peer-to-Peer (P2P) gives new light on multicast, and Application Layer Multicast (ALM) begins to attract wide attention. ALM protocols organize a set of hosts into an overlay tree for data delivery and all multicast related functions are implemented within end system. The shift of multicast functions from routers to hosts has the potential to address most problems associated with IP multicast, but also presents new challenges. This paper analyses the characteristics of ALM research from distributed system perspective, describes the technologies and difficulties involved in ALM design, and summarizes current work and progress.

Keywords IP multicast, Application layer multicast, Overlay network, Multicast tree

1 引言

组播(Multicast)能节省网络资源,减少网络管理要求,为数据的多点并发传输提供了有效途径。组播技术是下一代 Internet 应用如视频会议、远程教育、数据复制、网上游戏等系统的关键支撑技术。

为保证网络协议设计的通用性和适应性,Internet 体系建议在网络层仅提供基本的报文传输功能,并将其他一些重要功能(如差错处理、拥塞控制、流控制)交由终端系统实现。文[2]从端到端的角度给出了新增功能的部署原则:除非在低层实现所获得的性能收益能超过所需的额外开销,新增功能应尽可能放在上层实现。Deering[1]根据这一原则,从协议设计效率出发提出了 IP 组播体系,即在 IP 层增加对组播通信的支持,将具体的组播功能(如组成员管理、组播报文路由)实现于路由器。IP 组播功能完全集成在路由器上,传输链路分支节点(组播路由器)根据自身维护的组状态信息自动复制报文,避免产生重复报文。因此,IP 组播是组通信环境中实现数据分发的最有效手段。然而,相关技术研究虽经历了十多年的发展,但由于自身存在的一些缺陷,IP 组播技术并未能得到广泛应用。首先,从技术角度,组播要求路由器维护每个组播会话的状态信息,增加了路由设计的复杂度和服务过载,相关研究仍存在一些热点问题(如域间路由问题),许多协议也未能标准化;其次,从市场角度,IP 组播采用 UDP 作为传输协议,提供的是"尽力而为"(best-effort)型服务,缺乏对可靠性及拥塞控制的有效处理,而且还打破了传统基于"输入流"的计量机制,因此,许多 Internet 服务提供商(ISP)仍不情愿在广域环境下提供对组播服务的支持。最后,从应用角度,IP 组播的实施涉及到对现有网络基础设施的更新,进一步限制和阻碍了 IP 组播应用的大规模开展。

近几年来,P2P 覆盖网络技术的出现为基于无集中控制结构的分布式应用提供了新的思路,同时也引发了对"IP 层是否为部署组播功能的最佳位置"这一问题的再思考。结合 IP 组播的发展现状,研究人员试图借鉴 IP 组播的基本设计思想,在应用层建立组播机制,作为 IP 组播的替代实现技术。

*)本文工作得到国家重点基础研究发展计划(973)项目(2002CB312002)和国家"八六三"高技术研究发展计划项目(2001AA113050)的资助。
叶保留 博士研究生,主要研究方向为分布式计算与并行处理。李春洪 博士研究生,主要研究方向为分布式计算与并行处理。姚键 博士研究生,主要研究方向为分布式计算与并行处理。顾铁成 博士研究生,主要研究方向为分布式计算与并行处理。陈道蓄 教授,博士生导师,主要研究领域为分布式计算与并行处理。

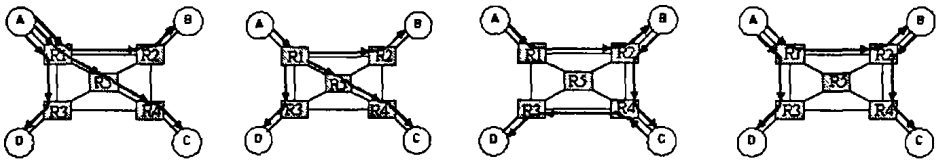
应用层组播研究旨在直接在组成员之间建立数据传递树，将组播相关功能(包括组成员管理、报文复制、数据分发等)实现于终端主机，在网络层仍采用 IP 单播实现数据传输，从而取消对网络基础设施(如组播路由器)的依赖。应用层组播的配置、部署毋需更改现有 Internet 基础设施，可解决 IP 组播中所面临的大多问题，近两年来受到学术界的广泛关注。文[3～12]列举了有关应用层组播研究的部分工作。

本文结合应用层组播的基本机制，分析了应用层组播研究存在的一些热点问题和技术难点，介绍了相关工作进展。本文第 2 节描述了应用层组播的基本模型及其特征；第 3 节概述了应用层组播协议的构造方法，并对相关协议进行了比较；第 4 节对协议设计所涉及到的关键因素和协议质量评价的主要技术指标进行了分析；最后总结了应用层组播研究现状，并提出了今后可能的研究方向。

2 应用层组播概述

2.1 数据传输模型

应用层组播的基本思想是屏蔽底层物理网络的拓扑细节、将组成员节点直接自组织成一个逻辑覆盖网络(Overlay Network)，并在应用层提供组播路由协议来构建和维护该网络，为数据传输提供高效、可靠服务。应用层组播将所有组播功能完全集成在主机中，由应用层软件具体实现。图 1 给出了应用层组播与传统网络层数据传输模式的对比示意图[6]。图中主机 A 为发送源，主机 B、C、D 为接收者集合，$R1$～$R5$ 为路由器。图 1(a)显示了单播方式下的报文传输路径，该方式下发送者为每个接收者分别建立从发送源到目的地的传输路径，因而在上游路径易于产生重复报文，而且越靠近发送源冗余量越大(如图 1(a)中的 A—$R1$ 段)，容易造成链路拥塞、形成服务瓶颈。图 1(b)描绘了典型的 IP 组播流程，为加入组播组，接收者与组播路由器联系，由路由器负责组成员的管理和组播树的创建与维护。传输过程中，数据沿着组播树流向接收者，报文复制由中间路由器根据需要自动处理。由于组播路由器拥有链路级的组播树分枝路径信息，因此，基于路由器的报文复制、转发机制使得任何物理链路上只可能存在一个数据拷贝，可有效避免冗余传输，节省网络带宽，因而有很好的协议性能。应用层组播(如图 1(c)、图 1(d))将所有组播功能迁移到终端主机，传输网络的构建完全独立于 IP 层的物理链路，数据流向及报文复制由终端主机控制，具体的报文传输仍依赖于 IP 单播传输机制。这种基于主机控制的报文复制方法使得在覆盖网络分支节点所对应的物理链路(如图 1(d)中的 A—$R1$ 段)及中间转发节点与路由器之间的迂回路径(如图 1(d)中的 B—$R2$ 段)上可能会存在同一报文的多个副本，由此可见，应用层组播对带宽的利用率不及 IP 组播。图 1(c)和图 1(d)显示了两种不同的应用层组播传输模式。



a 单播（顺次直接单播）　b IP 组播　c 应用层组播（令牌组播）　d 应用层组播（一般情形）

图 1　典型数据传输模型

从部署位置上来看，应用层组播试图将组播功能推向应用上层，通常可用以下两种系统结构来实现应用层组播[8]：

1)基于 Peer-to-Peer 体系结构。这是一种基于动态节点的覆盖组播技术，系统中组成员节点状态动态变化且完全分布，节点之间通过特定算法、协议自组织成控制网络和数据分发树。每个节点仅维护自身参与组的状态信息，所有组播相关功能以软件实体形态集成于参与组播会话的节点中。基于 P2P 结构的组播协议设计强调如何在动态网络环境下提供有效的组播传输机制，保证协议的可靠性，提高协议的自适应性。文[3～9,11,12]针对 P2P 环境提出了各种不同的解决方案。

2)基于代理的体系结构。这是一种基于固定节点配置的覆盖式组播技术，一般由增值服务提供商根据一定策略在 Internet 的某些位置部署应用层代理节点，代理节点之间的数据传输路径和传输方式也由增值服务商预先确定。终端主机通过接入距自身最近的代理节点获取数据。代理方案实质上是一种中继技术，在终端主机和代理之间建立临时通道，也常被称做隧道化方式(tunneling approach)。该方法的主要优点是代理节点相对固定，组播树比较稳定，容易针对具体的应用需求提供特定的 QoS 保证。但所提供的组播服务缺乏伸缩性，容易在代理节点形成服务瓶颈。此外，还常常需要 ISP 的支持。文[10,13]的主要工作都借鉴了该思想。

2.2 技术特征

应用层组播是建立在应用层之上的一种虚拟网络，与 IP 组播相比，协议设计的主要差异包括：1)报文转发位置。应用层组播的数据转发节点可以是覆盖网络中的终端主机或专有服务器，而 IP 组播的报文转发必须由核心路由器来处理。2)网络拓扑的创建方法。应用层组播树是由节点直连而成的覆盖网络，该网络的构建和优化常依据在部分或全部节点之间建立的一些度量尺度，完全隐藏物理网络拓扑特征；构造 IP 组播树的所有路由协议(包括域内路由、域间路由协议)都依赖于具体的网络链路，与 IP 链路信息密切相关。3)组管理模式。IP 组播的组成员关系信息分布于组播路由器，组管理由路由器负责；而应用层组播的成员关系可由系统中的会聚点(RP)集中控制，也可由各节点根据自身维护的局部状态信息协作管理。应用层组播由于节点及网络状态动态变化，一般通过建立一个与数据传输网络有别的控制拓扑来管理，要求控制拓扑能在异常条件下替代传输网络提供数据传输服务。

从功能特征来看，IP 组播设计强调时效性，有较好的传输效率和资源利用率。但受路由器存储空间限制，因此，可跨越空间位置、而不能跨越时间维传输信息[3]。通常要求接收者同步接收数据。IP 组播非常适合具有严格时间要求、且能容忍一定丢包率的实时应用(如视频服务、网络游戏)。应用层组播将组播功能实现于上层，具体的报文传输仍由传统的 IP 单播机制负责。由于覆盖网络与物理网络分离，因此协议效率不及 IP 组播。但应用层组播可充分利用节点的存储能力，结合 IP 层和应用层的 QoS 机制，有效避免和解决 IP 组播中存在的拥塞控制、可靠传输等问题，提供端到端的服务质量保证。

应用层组播的实现不需要网络中路由器提供任何特殊支持，可以类库形式被调用或直接内嵌于应用代码之中，因而操作方便、易于广泛部署。此外，应用层组播能同时跨越时间维和空间维进行数据传输，更适合能容忍一定程度延迟的高可靠应用，如电子邮件、文件传输。

3 协议构造方法

应用层组播将组播功能实现于终端主机中，组成员的动态性对组播有很大影响。应用层组播协议不仅要提供有效的数据组播分发树，还要针对节点的动态性提供可靠的组管理算法。协议设计强调在动态网络环境下维持网络的稳定性。应用层组播路由协议设计面临的主要问题是，如何在广域环境下，针对节点的动态性、在节点上建立必要的状态信息，并根据这些信息构建优化的组播路由协议。图 2 根据协议构造算法的差异对现有应用层组播路由协议的构造方法进行了分类。

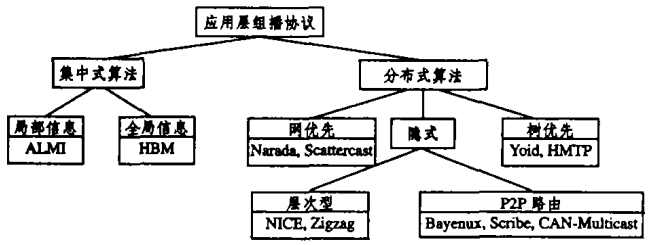


图 2 应用层组播协议分类

3.1 集中式算法

集中式算法在协议中引入全局会话控制点，集中管理所有组成员的状态变化（加入/离开）及成员之间的位置（距离）关系信息，负责组播树的创建和维护，并定期计算、优化覆盖拓扑。集中式方法具有较好的可靠性，而且能减少控制过载，易于维护组播树的一致性和效率。但由于控制点要维护全局信息，受单点失败的限制，因而缺乏可伸缩性，比较适合小规模稀疏型组通信应用。根据协议优化所依赖信息的不同，集中式算法又可分为基于全局信息和基于局部信息两种组播树优化策略。ALMI[5]、HBM[14]都属于集中式应用层组播协议。

ALMI 的每个会话包括一个会话控制器和若干个会话成员。会话控制器控制覆盖网络的全局信息，具体处理组成员的注册请求和维护组播树。为保证协议效率，会话控制器根据所有组成员反馈的最新链路信息定期重新计算并生成最小生成树（MST），该组播树是一个具有双向链路的共享树（shared tree）。为降低控制负载，ALMI 协议采取基于局部信息的优化算法，对每个节点所监视的邻居节点数进行限制。因此，ALMI 构建的组播树是次优的。

HBM 将组成员分成核心成员和非核心成员两类，核心成员形成数据分发树的主干节点，非核心成员只能作为叶子节点。组播中的所有活动都由一个集中汇聚点（RP）控制，汇聚点维护会话中所有成员的分类信息及相互距离。覆盖拓扑的优化完全依赖于 RP 中的全局信息，较 ALMI 而言，较容易得到最优化的网络拓扑。

3.2 分布式算法

分布式算法中没有集中的会话控制点，组成员的管理、数据传输网络的控制通过各节点维护的局部信息协作完成。分布式算法可避免集中式算法受单点约束的限制，有更好的可伸缩性。然而，在动态分布环境中，频繁的加入/离开操作、节点及网络条件的异常变化使组播树极为动态。因此，基于分布式算法的组播协议设计更为复杂和困难，通常需要协调和衡量协议设计所引起的每节点维护状态信息量、控制过载、通信代价等开销。为适应系统的动态性，保证组播传输的有效性，分布式算法常将组成员组织成控制拓扑和数据拓扑两种具有不同功能目标、但均覆盖所有成员的网络结构。控制拓扑一般是基于局部信息的连通图，负责监视节点变化、维护网络的连通性。数据拓扑是一个开环树，主要为数据传输提供高效的组播路径。现有协议大多把数据拓扑看作是控制拓扑的一个真子集。根据控制拓扑与数据拓扑建立顺序的不同，分布式算法又可分成网格优先（mesh-first）、树优先（tree-first）、隐式（implicit）建立等 3 种不同方法[16]。

3.2.1 *网格优先法* 首先将组成员自组织成一个连通的覆盖 mesh，该 mesh 中任意两个节点之间存在多条路径，每个节点都要维护全部（或部分）其他成员信息，并借助一定算法来避免和检测网络分区。然后通过路由算法在 mesh 上建立以指定源为树根、涵盖所有组成员的单独树（per-source tree）。网格优先法通常显式生成 mesh 拓扑，而组播树的建立依赖于具体的路由算法和 mesh 拓扑，并且 mesh 的选择直接影响着组播树的性能。文[8]认为，高质量的 mesh 应满足以下两个属性：

1）任意节点对之间的 mesh 路径质量（带宽、延迟）应与其对应的单播路径质量类似。

2）每个节点应只拥有少量 mesh 邻接节点。

属性 1 与组播树的数据传输效率密切相关，属性 2 决定了节点需维护的状态信息量及平均控制负载。网格优先方法可为不同的发送源分别建立优化的单独树，还允许在 mesh 上建立抽象的组管理功能和负载均衡的路由算法，具有较好的数据传输效率。该方法的一个主要缺点是，由于在控制网络上建立了多源覆盖树，每个节点都需要花很大代价来维护、优化多个组播树。图 3(a)给出了基于网格优先方法的覆盖拓扑结构示意图。Narada[8]、Scattercast[10]都采用了基于网格优先法的协议构造方法。

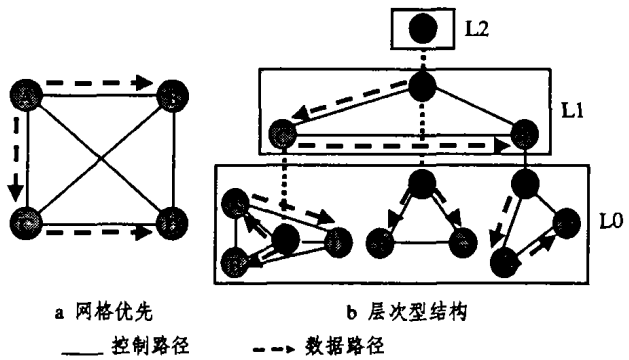


图 3 覆盖式组播网络

Narada 被认为是最早的应用层组播协议之一。新成员首先通过 RP 获取部分成员，并试图作为这些成员的邻接节点加入 mesh。组成员定期与 mesh 中的邻接节点交换信息，及时检测失败节点，维护 mesh 的连通性。Narada 采用逆向路径转发路由算法（如 DVMRP）在 mesh 上为各个发送源分别建立数据分发树。为提高数据传输质量，Narada 采用分布式启发算法，由每个节点周期性探测与其他成员的相互位置关系，并根据探测结果决定是否添加新链路，从而优化 mesh 拓扑。Narada 中每个成员所需维护状态信息为 $O(N)$，缺乏可伸缩

性，适用于中小规模稀疏型组播应用。

Scattercast 是一种基于代理的组播体系结构，组播服务通过各策略配置点间的代理(SCX)协作实现。协议将组成员分成若干个正交组，每个组由一个 SCX 提供服务。新成员通过最近的 SCX 加入组播会话，SCX 之间采用 Gossamer 协议定位彼此，并根据一定的应用层语义知识建立和优化覆盖 mesh。Scattercast 借助周期性的心跳消息(heartbeat message)检测 mesh 分区。SCX 通过执行逆向最短路径路由算法在 mesh 上为每个发送源分别建立单独树。为提高组播树效率，SCX 定期探测彼此距离，寻找有用连接。Scattercast 还可根据一定的应用层语义对组成员分组，并对组播传输内容作自适应调整，能有效解决组播会话的异构性。

3.2.2 *树优先法* 基于树优先法的组播协议首先根据各节点的局部信息创建一个共享树，随后，各组成员从覆盖树中寻找非邻接节点，根据一定算法建立并维护到这些节点的控制链路，从而使控制拓扑得到加强。与网格优先方法不同，树优先法中的控制网络与组播树分离，组播树不一定是 mesh 的子集。该方法的优点是赋予用户对组播树更多的直接控制能力，可以控制节点的最大出度，选择适合的父节点。树优先法的主要缺点是对组播树的回路检测和避免处理比较复杂，而且对组播树的优化也比较困难。Yoid[3]、HMTP[12]都采用了树优先方法。

Yoid、HMTP 都能集成现有 IP 组播技术并向非 IP 组播区域提供组播服务，两者组成员加入机制非常相似。Yoid 为应用层组播的实现提供了完整的应用框架，可看作是一个协议集。包括拓扑管理协议 YTMP，传输控制协议 YDP，身份识别协议 YIDP，传输层协议 yTCP、yRTP、yMTCP、YMRTP。Yoid 的控制 mesh 和组播树由 YTMP 分别创建，但两者的优化目标不同。前者优化主要是为了提高健壮性，而后者以提高协议效率为目的。控制 mesh 除维护正常的组管理功能外，还对网络分区情形下的数据传输服务提供支持。为加入组播树，新成员首先通过 RP 获取部分组成员列表，然后从中选择具有可用度且不会产生闭环路径的节点作为父节点。

HMTP 主要处理组播的配置问题，协议将组播服务覆盖区分成若干个正交区域，并为每个区域选择一个指定成员(DS)。所有的 DS 通过 HMTP 自组织成双向共享树。HMTP 为每个组会话定义一个引导点 HMRP，新成员首先通过 HMRP 获取树根地址，并从根节点开始由上向下逐层寻找，选取一个距自己最近、且有可用度的节点作为自己的父节点。为检测闭环，HMTP 中的每个成员都要维护到根节点的路径。HMTP 不要求显式生成 mesh，协议让每个节点周期性探询和缓存组播树中部分成员信息来避免、解决网络分区。

3.2.3 *隐式法* 隐式法中控制拓扑和数据拓扑的定义没有严格的先后顺序，成员之间也不需要额外交互。协议的控制拓扑一般要求满足一定的属性需求，数据传输路径通常要求是建立在报文转发规则之上、能平衡控制拓扑属性的开环组播路径。根据控制拓扑建立方法的不同，隐式法又可分为层次型、P2P 路由等几种类型。

层次型方式。协议将组成员分成若干层，并由下向上为每个层顺序编号(最低层编号为 L_0)，每层都包含若干个簇，所有成员都一定属于 L_0，图 3(b)描述了层次型网络结构示意图。层次型拓扑构建通常从 L_0 层开始按如下方法递归定义：协议依一定规则将 L_i 层的组成员划分成若干个具有一定大小限制的簇，并选取簇的图论中心节点作为簇头，L_i 层的所有簇头形成 L_{i+1}。随后根据该层次结构定义控制和数据传输的覆盖拓扑。新成员加入系统时首先从最高层向下顺序在每层探测距自己最近的成员，并加入 L_0 相应簇。

NICE[4]、Zigzag[17]的层次型拓扑构建思想基本相同，都直接利用该拓扑做组管理机构，但两者的数据传输机制不同。NICE 中每个成员都可直接在层次拓扑上建立以自己为根节点的组播树，状态信息交换只在簇内进行，每节点维护的状态信息最多为 $O(k\log_k N)$(k 为决定簇大小的常数)。Zigzag 的每个簇内还包括一个副簇头，簇头用于监视簇内成员关系，副簇头负责数据分发。此外，Zigzag 为规范组播树的创建定义了 C-rules。与 NICE 相比，Zigzag 最大优点是能将节点的状态信息量和失败恢复过载限制在常量水平。

P2P 路由方式。Bayeux[7]、Scribe[9]、CAN-Multicast[6]都建立在 P2P 的路由设施之上的应用层组播协议。Bayeux 以应用层路由协议 Tapestry 为基础，采用显式途径为每个发送源分别建立和维护组播树。Tapestry 中，每个节点都有一个独立于位置和语义属性、由固定长度位序列组成的标识符。每个节点都维护一个多级邻接表，为消息传递提供增量路由服务。Tapestry 本身是一个基于后缀匹配的层次型路由协议。Bayeux 有良好的可扩展性和容错能力，协议还通过复制根节点和群簇技术为实现负载均衡、提高带宽利用率提供支持。为加入组播，新成员首先发送“JOIN”消息通过 Tapestry 路由到发送源，发送源仍采用 Tapestry 路由反馈“TREE”消息至该节点，并在反馈覆盖路径的每个节点记录请求者身份，更新转发表信息。请求离开的处理方式与此类似。Bayeux 中每个成员的加入/离开都要经过根节点，因此，在根节点受单点失败限制。Scribe 建立在 P2P 对象定位和路由协议 Pastry 之上，其控制拓扑与 Pastry 的控制拓扑相同，每个成员都包含路由表、邻接点集和叶子集入口等信息。Scribe 的组播树通常由 Pastry 覆盖网络的子集组成，协议为每个组定义一个唯一标识符，与该标识最近的成员成为该组的 RP，负责组成员的注册管理，新成员通过组标识发送消息到 RP 加入会话。组播会话的数据传输拓扑由不同组成员到 RP 的 Pastry 单播路径联合而成。Bayeux、Scribe 都支持单独树，但前者组播树包含从根节点到每个目标节点的路由信息，而后者则包含从每个目标节点到根的信息。与 Bayeux 相比，Scribe 中每个节点维护的状态信息较少，而且成员的请求操作由本地处理，因此，Scribe 有更好的伸缩能力。

CAN-Multicast 以覆盖网络 CAN(Content-Addressable Network)为基础，CAN 将组成员组织成 n 维笛卡儿坐标空间，每个成员占据一定空间。与其他协议不同的是，CAN-Multicast 不需要创建组播树，而是通过为每个会话分别建立 CAN 子网络来为数据传输提供服务。每个节点只需要维护自己所属组的信息，而与组大小、数据发送源无关。组播报文通过泛洪方法向邻接节点发送。协议采用哈希函数为每个组播组映射一个引导节点，新成员通过该引导节点加入会话组，随后的节点加入、离开操作由常规 CAN 维护算法处理。CAN 中每个成员只需维护 $2d$ 个邻节点信息(其中 d 为坐标维数)，但为每个组分别维护一个 CAN 覆盖网络需要一定的代价。

4 协议质量度量

4.1 覆盖效率

应用层组播树是一个建立在应用层之上覆盖网络，虚拟网络特征使得协议设计必须从物理网络和上层应用两个方面

考虑网络的覆盖效率。从网络角度，覆盖网络应尽量使物理链路上的冗余传输尽可能最少；从应用角度，覆盖网络应能满足具体应用的端到端延迟需求。通常，可通过以下三个参数来评估数据路径质量：

1)压力度(stress)。该参数以网络中某段物理链路(或路由器)作计量单位，统计发送至该链路上的相同报文数，组播协议对网络带宽的利用率与压力度成反比关系。

2)伸张度(stretch)。该参数以每个接收者成员作为统计单位，计算沿覆盖网络以数据源到该收者的路径长度与直接的单目路径长度的比值。伸张度实质上反映了数据传输的相对延迟损失。

3)节点度(node degree)。覆盖网络上与该节点直接、接收由该节点转发数据的组成员数。

节点度较大将有助于降低网络传输延迟，该参数主要受限于节点的出口带宽。

不同的应用层组播路由协议生成不同的覆盖式路径，因此，在数据传输质量上存在一定属性差异。以图 1 为例，设物理拓扑上每段路径的长度为一个单位值，表 1 对不同情形下的数据传输路径质量进行了分析。

表 1　不同传输方式下的数据传输质量(以图 1 为例)

	单播	IP 组播	应用层组播		
			令牌组播方式	顺次直接单播方式	一般情况
最大压力度	3	1	2	3	2
平均伸张度	1	1	1.83	1	1.67

从表 1 可以看出，IP 组播具有最佳压力度和伸张度属性，顺次直接单播方式与 IP 单播方式具有相同的数据传输质量。特别地，顺次直接单播是应用层组播的一种极端方式：发送源分别建立到各个接收者的数据传输路径，覆盖式网络是一个以发送源为树根、接收者集为直接孩子的两层结构。因此，数据传输拓扑与图 1(a)等价，其中发送源的出度为 O(N)(其中 N 为组成员总数)，最大压力度 O(N)，平均伸张度为 1。图 1(c)显示了应用层组播的另一种极端情形—令牌组播，此时最大压力度为 2，而平均伸张度为 O(N)。大多数应用层组播都采用了类似图 1(d)的折中方法，即根据带宽条件和访问延迟需求在伸张度与压力度之间权衡、优化算法协议。

4.2　组管理

IP 组播将组管理交由网络中位置相对固定的路由器负责，网络拓扑相对稳定，部分成员的“加入/离开”操作不会造成整个网络的波动。对应用层组播而言，组播功能在终端主机实现，覆盖控制拓扑及数据传输路径都是直接建立在组成员之间的动态网络，组成员的状态变化将带来全局网络拓扑的动态变化。因此，组管理必须能及时发现组成员的状态变化、根据变化情况调整控制网络和数据传输网络拓扑，避免网络分区，维护全局一致性。组管理反映了协议设计的可靠性、健壮性及自适应能力。通常用协议过载来衡量组管理质量，协议过载包括用于保持覆盖网络连接的控制报文和探测报文，协议过载主要体现在正常情况下的平均控制过载和节点异常失败所产生的额外处理开销来。平均控制过载主要产生在节点间周期性的状态信息交换，因此，与节点维护的状态信息密切相关。网格优先表方法由于成员间要周期性交换状态信息，因而有较高的平均控制过载。表 2 总结了部分应用层组播协议的控制过载情况。

表 2　应用层组播协议比较

组播协议	构造方法	树性质	最大路径长度	最大子树度	平均控制负载
HBM	集中式	共享树	无上界	无确界	O(N)
ALMI	集中式	共享树	无上界	无确界	O(N)
Narada	网格优先	单独树	无上界	无确界	O(N)
HMTP	树优先	共享树	无上界	O(最大节点度)	O(最大出度)
Yoid	树优先	共享树	无上界	O(最大节点度)	O(最大出度)
Scribe	隐式	单独树	O(logN)	O(logN)	O(logN)
Bayeux	隐式	单独树	O(logN)	O(logN)	O(logN)
CAN-multicast	隐式	单独树	$O(dN^{1/d})$	O(d)	常量(与 d 相关)
NICE	隐式	单独树	O(logN)	O(logN)	常量(与 k 相关)
Zigzag	隐式	单独树	O(logN)	O(k)	常量(与 k 相关)

4.3　路由效率

应用层组播的协议设计独立于网络的物理链路，网络控制拓扑和数据传输路径通过节点直接互连而成，数据复制由主机节点控制，而具体的数据传输依赖于 IP 单播链路。因此，路由效率同时受到应用层和网络层的限制。好的路由协议设计必须协调两层的关系，即同时考虑覆盖网络的设计及其与物理网络的关系，使协议具有较小的传输延迟和较高的资源利用率。压力度和伸张度虽提供了链路级的覆盖网络效率评价标准，然而，由于过于依赖物理网络的拓扑特征，因此很难为分析、计算这些性能指标建立准确的数学模型。但对组成员相对位置关系的充分考虑有助于构建高效的覆盖网络。现有工作中，Bayeux 和 NICE 可为“稠密型”组分布维持常量级的伸张度。

在应用层，组播树拓扑受节点的出口带宽(出度)能力约束，并且路由效率与协议构造方式密切相关。集中式算法协议和树优先协议创建的都是共享树，不能针对不同发送源优化，因此不太适合实时应用。网格优先协议虽能针对各个发送源建立单独树，但组播树依赖于 mesh，因此不一定能生成最优树。隐式协议和树优先协议由于都能控制节点出度，都非常适合高带宽应用，但隐式协议也非常适合“延迟敏感型”应用。从路由构造方法上来看，基于传统 IP 路由算法(如 DVMRP)的路径构造方法能较好反映物理链路特征，能降低协议的伸张度。而采用 P2P 对象定位路由协议的路径构造方法由于完全基于应用层语义，因而虽然能保证应用层的延迟统计(节点间的跳数)，但不一定有很好的伸张度属性。

4.4　可伸缩性

伸缩性反映了协议设计受规模的影响，决定了组播协议的应用规模，是组播协议设计强调的一个重要因素。应用层组播的可伸缩性主要体现在以下两个方面：1)节点的状态信息。由于报文转发功能交由动态节点完成，因此协议要通过节点保存部分(或全部)信息来维护网络完整性，及时修复网络分

(下转第 53 页)

产权的国产分布式数据库管理系统DM4上探讨了如何实现Java存储过程，并首次实现了Java虚拟机和JDBC驱动程序的内置，使数据库管理系统更好地从内部支持Java语言。但该系统尚存在不足之处，如何实现Java存储过程的分布式处理和虚拟机的安全共享内存执行，尚有一大段路要走。

参考文献

1 Louden K C. Compiler Construction Principles and Practice. PWS Publishing Company, 1997

2 Aho A, Jeffrey V, Ullman D. Principles of Compiler Design. Addison-Wesley, 1977

3 Ertl M A. Stack Caching for Interpreters. In: Proc. of the ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'95), ACM Press, June 1995. 315～327

4 Romer T H, Lee D, Voelker G M. Wolman A, Wong W, Baer J-L, Bershad B N. Levy H M. The Structure and Performance of Interpreters. ASPLOS VII, 1996

5 Proebsting T A. Optimizing an ANSI C Interpreter with Superoperators. In: Proc. Symp. on Principles of Programming Languages, 1995. 322～332

6 Miranda E. BrouHaHa---a portable Smalltalk interpreter. In: Proc. of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'87), ACM Press, Oct. 1987. 354～365

7 SQLJ reference information site. http://www.sqlj.com

8 Oracle JDeveloper information. http://www.oracle.com/tools/jdeveloper/suiteintro2.html

9 Information on IBM's SQLJ implementation. http://www.ibm.com/java

10 Sun Mircosystem5 Inc. The Java virtual machine specification. 2nd ed. [EB/oL]. http://java.sun.com/ldocs/books, 1999-09-01/2004-04-25

（上接第10页）

区。节点的状态信息量与覆盖网络控制算法相关，通常，基于集中式控制和泛洪方式的组管理的状态信息量较大，维护网络所需的通信过载也较大；而层次型结构和基于局部知识的组管理节点状态信息量较少，控制也相对简单。2)组播会话个数。该参数主要与组播树的构建方法相关。对共享树而言，所有数据源都通过该组播树发送数据，不需为每个会话维护单独的状态信息；而单独树由于要为每个数据源建立独立的数据分发树，节点必须为不同会话维护相应的状态信息，因此关于会话的状态信息与会话个数成正比。但从传输效率来看，由于单独树可以为每个发送源定制树，可以创建具有较小数据传输延迟的优化树。

从协议构造方式来看，网格优先协议由于创建的是单独树，且有较高的状态维护要求，故较适合小规模组播应用。而隐式协议一般都基于结构化设计，因而有较强的可伸缩能力，非常适合大规模应用。

总结 应用层组播技术将组播功能实现于应用层，避免了IP组播对网络基础设施的要求，解决了IP组播应用所面临的缺陷（如路由技术、组播地址空间），为组通信应用提供了一种新的解决途径。协议设计借鉴了IP组播的数据通信模型，节点间根据网络条件和组成员关系自组织成覆盖网络，并根据系统的动态变化作出自适应调整，为大规模动态环境（如P2P应用）提供了有效的数据传输服务。应用层组播的协议实现可集成现有网络设施对IP组播的支持，但大规模应用不要求网络层提供这些额外支撑，易于在Internet环境下部署。

应用层组播适应了新型分布式计算环境特别是P2P网络的需要。近年来，相关研究主要针对系统的动态性、结合协议设计的性能要求提出了一些解决方案，为相关技术的广泛应用提供了基础。进一步工作还应针对应用层的实现方式从软件部署方法、服务质量、系统安全等方面来增强和完善现有工作。

参考文献

1 Deering S. Multicast Routing in Internetworks and extended LANs. In: Proc. of ACM SIGCOMM. Aug. 1988

2 Saltzer J, Reed D, Clark D. End-to-end Arguments in System Design. ACM Transactions on Computer Systems, 1984, 2(4): 195～206

3 Fracis P. Yoid: Extending the Internet Multicast Architecture. April 2000, White Paper. http://www.aciri.org/yoid.

4 Bannerjee S, Bhattacharjee B, Kommareddy C. Scalable Application Layer Multicast. ACM SIGCOMM, Aug. 2002

5 Pendarakis D, Shi S, Verma D, Waldvogel M. ALMI: An Application Level Multicast Infrastructure. In: Proc. of 3rd Usenix Symposium on Internet Technologies & Systems, March 2001

6 Ratnasamy S, Handley M, Karp R. Application-level Multicast using Content-Addressable Networks. In: Proc. of 3rd Intl. Workshop on Networked Group Communication, Nov. 2001

7 Zhuang S Q, Zhao B Y, Joseph A D, Katz R H, Kubiatowicz J D. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination. In: Eleventh Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001), 2001

8 Chu Y, Rao S G, Seshan S, Zhang H. A Case for End System Multicast. In: the Proc. of ACM SIGMETRICS, June 2000

9 Castro M, Druschel P, Kermarrec A-M, Rowstron A. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in communications (JSAC), 2002

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

11 Leibeherr J, Nahas M. Application-layer Multicast with Delaunay Triangulations. In: Global Internet Symposium, Globecom, Nov. 2001

12 Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users. In: Proc. of IEEE INFOCOM 2002, New York, NJ, USA, June 2002

13 Finlayson R. The UDP Multicast Tunneling Protocol. work in progress, Draft-finlayson-umtp-07.txt, Sept. 2002

14 Roca V, El-sayed A. A Host-based Multicast (hbm) Solution for Group Communications. In: 1st IEEE Intl. Conf. on Networking, Calmar, France, July 2001

15 El-Sayed A, Roca V, Mathy L. A Survey of Proposals for an Alternative Group Communication Service. IEEE Network, Jan./Feb. 2003

16 Bannerjee S, Bhattacharjee B. A Comparative Study of Application Layer Multicast Protocols. http://www.cs.umd.edu/users/suman/publications.html

17 Tran D A, Hua K a, do T T. A Peer-to-Peer Architecture for Media Streaming. To appear in IEEE JSAC Special Issue on Advances in Overlay Networks