

# 应用层组播稳定性提高技术综述

苏金树 曹继军 张博锋

(国防科学技术大学计算机学院 长沙 410073)

**摘 要** 互联网上组通信应用的日益普及和传统 IP 组播发展面临的困境导致应用层组播逐步受到广泛关注. 将组播功能从路由器迁移到主机能够有效解决许多与 IP 组播相关的问题,但同时也带来一些新的挑战,如应用层组播稳定性问题. 文中概述了应用层组播的数据传输模型、组播树构造算法和协议性能评价标准,阐述了应用层组播稳定性问题产生的原因,提出了衡量应用层组播稳定性的标准,分析了影响应用层组播稳定性的因素,根据影响因素将应用层组播稳定性提高技术分类为降低节点离开事件发生频率的方法、缩小节点离开事件影响范围的方法以及缩短节点离开事件发生后组播树恢复时间的方法,并介绍了各种应用层组播稳定性提高技术,展望了该领域未来的研究工作.

**关键词** 组播;应用层组播;稳定性;影响因素;综述

**中图法分类号** TP393 **DOI 号**: 10.3724/SP.J.1016.2009.00576

## A Survey of the Research on ALM Stability Enhancement

SU Jin-Shu CAO Ji-Jun ZHANG Bo-Feng

(School of Computer, National University of Defense Technology, Changsha 410073)

**Abstract** Due to the increasing popularity of group communication applications in the Internet and the difficulties in deployment of IP multicast, Application Layer Multicast (ALM) begins to attract wide attention. The migration of multicast functions from routers to hosts has the potential to address most problems associated with IP multicast, but also faces with new challenges, such as the stability problem. In this paper, the authors firstly outline the data transmission model, multicast tree construction algorithms and protocol performance criterions of ALM. Then they analyze the reasons for the stability problem in depth and propose the criterion for evaluating the stability of ALM. The authors analyze the influencing factors of the ALM stability, which includes the frequency of node leave, the influencing scope of node leave and the time used to recovery ALM tree. According to the influencing factors, the technologies to enhance the stability of ALM can be classified as reducing the frequency of node leave, reducing the influencing scope of node leave and shortening the time used to recovery ALM tree after node leaves. Based on the classification, all the technologies of stability enhancement were overviewed and compared. And finally, we discuss some possible directions for future research in the area.

**Keywords** multicast; application layer multicast (ALM); stability; influencing factor; survey

收稿日期:2008-07-21;最终修改稿收到日期:2009-01-06. 本课题得到国家自然科学基金(906004006)、国家“九七三”重点基础研究发展规划项目基金(2009CB320503)和国家“八六三”高技术研究发展计划项目基金(2008AA01A325)资助. 苏金树,男,1962年生,博士,教授,博士生导师,主要研究领域为计算机网络、信息安全等. 曹继军(通信作者),男,1979年生,博士研究生,主要研究方向为高性能路由器技术、互联网组播技术等. E-mail: caojijun@nudt.edu.cn. 张博锋,男,1978年生,博士,助理研究员,主要研究方向为信息安全、互联网内容信息分类等.

## 1 引言

相对于单播而言,组播由于借助中间节点进行分布处理,从而实现了分布式并发传输模式,因此成为一种高效的数据传输机制。组播技术是互联网上组通信应用(例如视频会议、网络游戏、远程教育和网络电视等)的关键支撑技术。最早提出的组播方案是 Deering 提出的 IP 组播<sup>[1]</sup>。IP 组播在网络层实现组播功能,组内主机构成“主机组(host group)”并用 IP 组播地址标识,组播数据报文以“尽力转发”方式传输,组播路由和转发控制功能均由路由器完成。IP 组播虽然传输效率较高,但是由于技术和市场等原因,它在全球范围内的部署依然十分缓慢<sup>[2-3]</sup>。

面对 IP 组播发展的困境,Francis<sup>①</sup>和 Chu<sup>[4]</sup>分别于 1997 年和 1998 年独立地提出应用层组播思想,即由应用层实现组播功能。应用层组播技术在成员主机之间构建以网络层单播为基础的应用层覆盖网络(overlay network),组播路由和转发控制任务完全由成员主机承担。由于不需要额外的基础设施支持,因此易部署性成为应用层组播最大的优势。应用层组播自从被提出以来,国内外研究机构给予了相当的关注并且提出了大量的应用层组播协议<sup>[4-29]</sup>。目前,不少应用层组播系统已得到成功应用,例如 Coolstreaming<sup>[30-31]</sup>、PPLive<sup>[32]</sup>、PPStream<sup>②</sup>和 Sopcast<sup>③</sup>等。

IP 组播和应用层组播通常都以组播树为基础构建数据转发路径,但是它们各自的组播树却有着很大区别。IP 组播树的节点包括路由器和主机(路由器为组播树的内部节点,主机为组播树的叶节点),其边为路由器之间或主机与路由器之间的物理连接。应用层组播树的节点全部是主机,其边为主机之间的网络层单播连接。IP 组播树和应用层组播树构成元素的差异导致它们对主机节点的动态行为(节点的加入、退出和失效等)表现出不同的性质。由于新节点通常加入为组播树的叶节点,因此两种组播树都不会出现节点接收组播数据过程被中断的现象。当主机节点退出或失效时:(1)对于 IP 组播而言,由于主机节点是组播树叶节点,因此其它节点接收组播数据过程也不会被中断;(2)对于应用层组播而言,如果退出或失效节点是叶节点,那么情况和 IP 组播相同。如果退出或失效节点是内部节点,那么该节点下游的子孙节点接收组播数据过程将被中断,因而其子孙节点需要重新加入到应用层组播树以继续接收组播数据。可见,与 IP 组播不同,应用

层组播树会因为单个主机节点的退出或失效而被迫调整其它多个节点在组播树中的位置,该现象被称为由成员节点动态性引起的应用层组播稳定性问题,它严重影响了用户接收组播数据的连续性。因此,提高应用层组播的稳定性是提高应用层组播服务质量的必然要求。

本文主要综述了提高应用层组播稳定性方面的研究进展,并对各种研究成果进行了分类和比较,同时探讨了未来的研究方向。论文第 2 节概述应用层组播技术,简要介绍应用层组播数据传输模型,重点分析应用层组播树的构造方法,总结应用层组播性能评价标准;第 3 节提出应用层组播稳定性衡量标准并分析影响稳定性的因素;第 4~6 节对常见的稳定性提高技术基于类别逐一进行介绍和比较,从而获得对稳定性提高技术的研究现状的清晰认识;第 7 节给出全文总结和研究方向展望。

## 2 应用层组播基本问题

### 2.1 数据传输模型

应用层组播保持了互联网原有的网络层单播转发模型,组播功能完全由主机系统实现,即主机负责接收、复制和转发组播数据报文。根据在实现组播数据传输中任务分工的不同,应用层组播可以分为两个平面,即组播路由控制平面和组播数据转发平面。前者为组播数据转发提供依据,后者完成组播数据的传输。

#### 2.1.1 组播路由控制平面

应用层组播路由控制平面的核心功能是完成应用层组播树的构造、维护和优化。应用层组播树的构造主要是根据收集到的节点和覆盖网虚拟链路的相关信息(如传输延迟和节点支持的最大带宽等)构建满足一定约束条件并实现目标优化的组播树。应用层组播树的维护主要是根据节点的动态行为不断调整组播树的结构。应用层组播树的优化主要是根据随机探测的节点和虚拟链路信息自适应地改变组播树结构,以优化组播树的性能。

#### 2.1.2 组播数据转发平面

应用层组播数据转发平面的核心功能是基于应用层组播树进行组播数据转发。组播树被定义为组播成员节点间父子关系的集合,因此单从数据转发

① Francis P. Yoid: Extending the multicast Internet architecture. <http://www.aciri.org/yoid/>

② PPStream. <http://www.ppstream.com/>

③ Sopcast. <http://www.sopcast.com/>

平面而言,为了完成组播数据的接收和发送,组播树的内部节点至少具有父节点和子节点的地址信息,而叶节点至少具有其父节点的地址信息.

## 2.2 组播树构造算法

### 2.2.1 集中式算法

集中式算法引入集中控制点 RP (Rendezvous Point), RP 负责收集成员节点的网络测量信息并以此为依据计算应用层组播树,并把组播树结构的全局或部分信息分发给各个成员节点. 应用层组播树的计算和信息分发都是周期性的. 同时,当节点加入、退出或失效时,RP 负责重新计算组播树并把组播树的结构信息分发给成员节点. 集中式算法的优点是易于维持组播树的一致性和效率,其缺点是扩展性差,而且存在 RP 负载过重和单点失效问题. 集中式算法比较适合于小规模的应用场景,所以目前采用集中式算法的协议较少,典型的协议包括 ALMI<sup>[5]</sup> 和 HBM<sup>[6]</sup> 等.

### 2.2.2 分布式算法

分布式算法中不存在 RP 的概念,组成员管理和组播树的构造与维护都是依据各个节点维护的局部信息进行的,并且不存在重新计算整个组播树的问题. 虽然分布式算法较为复杂,但是通常具有优良的扩展性. 因此,大多数应用层组播协议都采用分布式算法.

通常,应用层组播路由控制平面的拓扑结构被称为路由控制拓扑,而数据转发平面的组播树结构被称为数据转发拓扑. 在路由控制拓扑中,成员节点周期性地交互控制消息(如心跳消息、更新消息和探测消息等)以检测节点失效并维护数据转发拓扑的连通性. 可见,路由控制拓扑是节点间邻居关系的集合,反映的是应用相关性. 数据转发拓扑是节点间父子关系的集合,反映的是数据传输关系. 根据路由控制拓扑和数据转发拓扑建立的先后顺序,分布式算法分类为路由控制拓扑优先(即网优先)、数据转发拓扑优先(即树优先)和路由控制拓扑隐含数据转发拓扑等(即隐含式).

#### (1) 网优先算法

网优先算法首先将节点间分布式地自组织成 Mesh, 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它成员节点信息,并及时修复网络分割. 然后通过运行标准的组播路由协议,如 DVMRP 等,在 Mesh 之上构建应用层组播树. 因此,组播树的建立依赖于 Mesh 拓扑结构和具体的路由算法. 网优先算法的优点是方便基于 Mesh 设计抽象的组管理功能和负载均衡机制,数

据传输效率较高. 同时,由于路由协议具有回路避免和检测内在机制,因此依靠标准的路由协议能够简化组播树的构造和维护过程. 缺点是复杂度较高、Mesh 维护开销较大和扩展性受限. 因此,网优先算法主要适合于中小规模的组播应用. 目前已经提出的采用网优先算法的协议主要包括 Narada<sup>[4]</sup>、Scattercast<sup>[7]</sup>、Kudos<sup>[8]</sup>、Bullet<sup>[9]</sup>、CoopNet<sup>[10]</sup>、Ipcast<sup>[11]</sup>、DONet<sup>[12]</sup>、PRO<sup>[13]</sup> 和 Prime<sup>[14]</sup> 等.

#### (2) 树优先算法

树优先算法的组成员首先直接在组内选择父节点,从而构造分布式组播树. 然后,每个组成员从组播树中主动发现一些非邻居节点,并根据特定算法建立并维护到这些节点的控制连接,组播树结构与这些额外连接构成路由控制拓扑. 树优先算法的优点是用户节点对组播树结构具有更多的直接控制能力,例如可以控制节点的最大度限制和选择合适的父节点等. 缺点主要是需要单独进行组播树的回路避免和检测处理,且组播树的性能优化比较困难. Yoid<sup>①</sup>、HMTp<sup>[15]</sup>、TBCP<sup>[16]</sup>、BTP<sup>[17]</sup>、Overcast<sup>[18]</sup>、TAG<sup>[19]</sup>、Hostcast<sup>[20]</sup> 和 PROMISE<sup>[21]</sup> 等协议采用此算法.

#### (3) 隐含式算法

隐含式算法对构造路由控制拓扑和数据转发拓扑没有严格的先后顺序,成员节点之间也不需要额外交互信息. 路由控制拓扑通常满足特定的属性要求,路由控制拓扑中隐含着数据转发拓扑,并且它们是同时由覆盖网络路由机制建立的. 根据路由控制拓扑的特征,隐含式算法又可分为两类:一类是层次型,如 NICE<sup>[22]</sup>、ZIGZAG<sup>[23]</sup> 和 Deaunay Triangulations<sup>[24]</sup>;另一类是 DHT 路由型,如 Bayeux<sup>[25]</sup>、Scribe<sup>[26]</sup>、CAN-Multicast<sup>[27]</sup> 和 SplitStream<sup>[28]</sup>.

## 2.3 协议性能标准

应用层组播协议一般都是针对具体应用场景设计的,所以其协议评价标准显现多样性特点. 评价具体协议的性能主要是通过与 IP 组播或设计的同等应用场景中的其它协议进行相关性能比较,通常比较数据路径质量和协议控制质量等.

### 2.3.1 数据路径质量

由于应用层组播树构建于应用层覆盖网络之上,所以不可避免地会出现覆盖网络拓扑结构和底层物理网络拓扑结构不一致的情况,从而使得覆盖网络中的通信给底层物理网络带来额外的负载,造成数据路径质量下降. 评估应用层组播数据路径质量通常以 IP 组播为基准,常用的性能测度包括链路强度(Stress)、链路伸展度(Stretch)和相对延迟开

销 RDP(Relative Delay Penalty)等<sup>[4]</sup>. 链路强度定义为同一物理链路传输相同数据报文的次数;链路伸展度指的是每个节点在应用层组播树中从源节点到该节点的路径长度与对应的单播路径长度的比值;相对延迟开销指的是应用层组播延迟与 IP 组播环境中相应延迟的比值.

### 2.3.2 协议控制质量

应用层组播协议控制功能主要体现于:(1)节点间交互心跳报文以保持覆盖网络连接;(2)节点间发送探测报文以发现成员节点及其连接的状态变化;(3)根据节点和连接关系的变化情况优化应用层组播树结构.因此,协议控制负载主要包括心跳报文和探测报文,协议控制开销分为通常情况下的控制负载开销和节点失效情况下所产生的额外控制负载开销.控制开销情况与协议的规模可扩展性有着密切关系,优良的协议要在满足应用需求的情况下具有较低的控制开销.除此之外,协议控制对组成员动态变化的适应性以及协议控制的鲁棒性等也是衡量协议控制质量的重要标准.

## 3 影响应用层组播稳定性因素的分析

大多数应用层组播协议都是以构造满足应用需求的高质量应用层组播树为目的.组播树的节点是参与应用层组播的主机,其边为网络层单播连接.整个组播树是在覆盖网络之上构造,因此也称为覆盖网组播树.

节点退出指节点主动地离开应用层组播会话.在集中式算法中,将要退出的节点会向 RP 发送通告离开的消息,然后 RP 立即根据维护的全局节点信息重新计算组播树并将组播树结构信息发布给成员节点.在分布式算法中,将要退出的节点通常向其子节点或子孙节点发送通告离开的消息,然后收到离开通告消息的节点选择将要重新加入的父节点并启动节点加入过程.

节点失效指在没通知任何其它节点的情况下节点离开应用层组播会话.应用层组播节点间连接关系的维护通常是依靠周期性地交互“心跳”报文实现的,因此这种连接被称为“软”连接.如果一个节点长时间内没有收到某个已知节点发送的心跳报文,那么可以有理由认为该节点已经失效,这是目前常用的基于超时机制的节点失效检测手段.一旦发现了节点失效,那么后续的处理过程就和节点退出的处理过程相同.

通常,节点退出和失效统称为节点离开.对于应

用层组播树而言,如果离开节点是内部节点,那么该节点的下游子孙节点接收组播数据过程将被中断,因而其子孙节点需要重新加入到应用层组播树以继续接收组播数据,从而引起应用层组播稳定性问题,该问题严重影响用户接收组播数据的连续性.为了研究由节点离开行为引起的稳定性问题对组播连续性的影响程度,需要定义应用层组播稳定性度量标准——“应用层组播稳定度”.

**定义 1.** 在应用层组播的某段时间  $D$  内,假设组播树  $Tr$  中所有节点组成的集合为  $V(Tr) = \{v_1, v_2, \dots, v_{m-1}, v_m\}$ ,任意节点  $v_i (v_i \in V(Tr))$  的总连接中断时间  $T_{\text{discont}}(Tr, D, v_i)$  为各次连接中断时间之和,节点的总连接保持时间  $T_{\text{keep}}(Tr, D, v_i)$  为各次连接保持时间之和,则时间段  $D$  内的应用层组播稳定度  $SD$  (Stability Degree) 定义为

$$SD(Tr, D) = 1 - \frac{\sum_{i=1}^m T_{\text{discont}}(Tr, D, v_i)}{\left[ \sum_{i=1}^m T_{\text{discont}}(Tr, D, v_i) + \sum_{i=1}^m T_{\text{keep}}(Tr, D, v_i) \right]} \quad (1)$$

可见,应用层组播的稳定性与各个成员节点的连接中断时间占组播会话总参与时间的比值密切相关,该比值科学地反映了连接中断程度.文献[33]提出两种衡量应用层组播稳定性的标准,一种是发生节点离开事件的平均时间间隔,另一种是节点离开的受影响节点数目,它没有考虑节点离开后组播恢复快慢对稳定性的影响.与文献[33]中的标准相比,式(1)能更全面地反映应用层组播的稳定性.

**定理 1.** 在应用层组播的某段时间  $D$  内,假设所有参与组播树  $Tr$  中的节点组成的集合为  $V(Tr) = \{v_1, v_2, \dots, v_{m-1}, v_m\}$ ,任意节点  $v_i (v_i \in V(Tr))$  一直参与该组播会话,即满足  $T_{\text{discont}}(Tr, D, v_i) + T_{\text{keep}}(Tr, D, v_i) = D$ ;节点离开事件的发生频率为  $F(Tr, D)$ ;节点各次离开事件的受影响节点数目的均值为  $\overline{\Delta n}(Tr, D)$ ;节点各次离开事件后组播树重构时间的均值为  $\overline{\Delta T}(Tr, D)$ ,则应用层组播在该段时间  $D$  内的稳定度  $SD$  为

$$SD(Tr, D) = 1 - [F(Tr, D) \cdot \overline{\Delta n}(Tr, D) \cdot \overline{\Delta T}(Tr, D)] / m \quad (2)$$

证明. 假设在组播时间  $D$  内,共发生了  $k$  次节点离开事件,第  $j (0 \leq j \leq k)$  次节点离开事件的受影响节点数目表示为  $N_j$ ,将本次节点离开事件中的受影响节点标识为  $N_{j,q}$ ,其中  $q$  为从 1 到  $N_j$  的正整数,节点  $N_{j,q}$  在本次节点离开事件中的连接中断时

间表示为  $\Delta T_{j,q}$ , 则由稳定度的定义得

$$\begin{aligned} DSD(Tr, D) &= 1 - \sum_{j=1}^k \sum_{q=1}^{N_j} \Delta T_{j,q} / (m \cdot D) \\ &= 1 - \left[ \frac{k}{D} \cdot \left( \sum_{j=1}^k N_j / k \right) \cdot \left( \sum_{j=1}^k \sum_{q=1}^{N_j} \Delta T_{j,q} / \sum_{j=1}^k N_j \right) \right] / m \\ &= 1 - [F(Tr, D) \cdot \overline{\Delta n}(Tr, D) \cdot \overline{\Delta T}(Tr, D)] / m \end{aligned} \quad (3)$$

所以, 该定理得证.

证毕.

由上述定理可知, 应用层组播的稳定度取决于 3 个方面的因素: (1) 节点离开事件发生的频率, 即单位时间内发生节点离开事件的次数, 它反映节点的动态性; (2) 节点离开事件发生时受影响的节点的数目, 它反映节点离开事件的影响范围; (3) 节点离开事件发生后受影响节点重新加入到应用层组播树所用的时间, 它反映了组播连接中断恢复的快慢.

稳定度概念是衡量应用层组播的总体稳定性的科学指标, 但实际中为便于研究影响稳定度的某方面因素, 可分解为如下子指标: (1) 反映节点的动态性的指标, 例如发生节点离开事件的频率和平均时间间隔<sup>[33]</sup>、节点失效发生率等; (2) 反映节点离开事件影响范围的指标, 例如受影响节点数目<sup>[34]</sup>、平均受影响节点数目、累积中断次数<sup>[35]</sup>等; (3) 反映组播连接中断恢复的快慢的指标, 例如平均组播树恢复时间、最小组播树恢复时间和最大组播树恢复时间等. 所有这些指标构成了衡量应用层组播稳定性的指标体系.

可见, 提高应用层组播树的稳定性需要采取三个方面的措施, 即降低节点离开事件发生的频率、减小节点离开事件的影响范围、缩短节点离开后组播树恢复时间. 目前已经提出的应用层组播协议中完全针对组播稳定性做出设计的还较少, 但是它们所使用的机制或算法中用来提高应用层组播稳定性的研究却很多, 下面将按照上述分类方法对目前用来提高应用层组播稳定性的技术进行综述.

## 4 降低节点离开事件发生频率的技术

降低节点离开发生频率的典型方法是采用专职的代理型应用层组播策略. 它利用互联网中的代理服务作为应用层组播的基础设施, 用户主机通过访问代理服务器而享受组播服务 (如图 1 所示). 从某种意义而言, 代理型应用层组播是 IP 组播和应用层组播的折衷, 因此它结合了两者的优点. 它也是组播体系结构研究的热点之一, Lao 等人认为它是未来互联网组播的一种长期解决方案<sup>[36]</sup>.

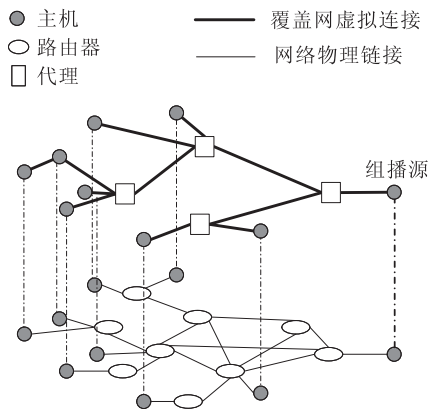


图 1 代理型应用层组播

代理型应用层组播研究的主要思想是提出基于代理的应用层组播体系结构并给出相应的覆盖网拓扑构造算法. 目前, 针对不同的应用需求已提出了多种体系结构, 例如, (1) RMX 体系结构<sup>[37]</sup>. 该体系结构适应于对适时性和可靠性要求较高的组播应用. 为解决异构节点间的可靠数据传输问题, RMX 体系结构提出语义可靠性概念代替传统数据可靠性概念, 从而允许使用应用级信息, 动态改变数据内容以适应传输速率的动态变化; (2) Overcast 体系结构<sup>[18]</sup>使用简单的组播树构造协议, 将内部节点组织成以数据源节点为根的组播树, 每个组播组提供数据复制并存放在各个内部节点. 该体系结构提供可靠的单源组播服务, 具有高扩展性, 特别适合于视频点播或直播应用; (3) Scattercast 体系结构<sup>[7]</sup>提供的组播服务由 Scattercast 代理 SCX 协作完成, 每个 SCX 同时为多个成员提供服务, 成员使用组播机制或单播与 SCX 通信. Scattercast 根据一定的语义对组成员分组, 并对组播传输内容作自适应调整以解决组播会话的异构性问题; (4) OMNI 体系结构<sup>[38]</sup>则适应于大规模流媒体应用, 它由多个分布在网络中的 MSN 设备提供数据分发服务, 端系统从某个 MSN 订阅服务, 从 MSN 到它的用户节点间的数据传输路径独立于 MSN 之间的骨干覆盖网络, 它可以采用 IP 组播、应用层组播或顺序直接单播等方式, 类似于层次式组播; (5) 在 TOMA 体系结构<sup>[39]</sup>中, 由 ISP 部署在网络中的服务代理构成组播服务覆盖网络 MSON 的服务域, MSON 的设计依赖于 MSON 提供者、网络服务提供者和组播组创建者之间的商业关系. MSON 采用聚合组播方法使得多个组播组共享一棵组播树.

由于代理节点通常是由 ISP 专门部署的高性能节点, 与普通用户节点作为转发节点相比, 其健壮性大大增强, 所以发生失效的可能性较低. 对于采用直



连访问的应用层组播(如图 2 所示)而言,用户节点的离开对其它用户不会造成影响.对于采用两层结构的应用层组播(如图 3 所示)而言,用户节点由于位于组播树的外层,其离开事件的影响范围也大为缩小.代理型应用层组播策略通过代理节点既降低节点发生失效的频率,也减小节点离开事件的影响范围,从而提高了应用层组播的稳定性.代理型应用层组播的稳定代理节点由于更接近于用户节点,因而能够提供稳定优质的组播服务.但是由于代理服务器的部署需要资金投入和 ISP 之间的相互协调,因此其可部署性比应用层组播略差.

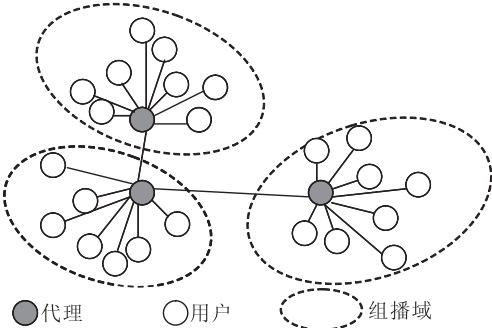


图 2 直连访问的代理型应用层组播

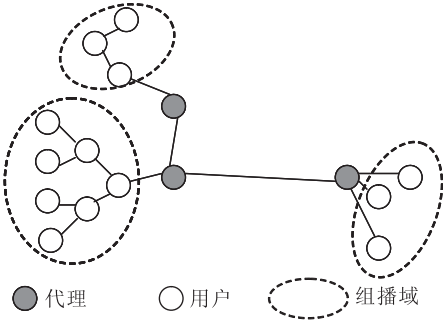


图 3 两层结构的代理型应用层组播

5 缩小节点离开事件影响范围的技术

5.1 利用节点的属性

在应用层组播树中不同节点的子孙节点数目不尽相同,因此不同节点离开的影响范围也不尽相同.整个组播树的平均受影响范围不仅与组播树结构有关,而且与各个节点离开的可能性密切相关.因此,要缩小节点离开的影响范围,需要考虑组播树结构和节点稳定性两方面的因素.

文献[6]提出 HBM 协议,它根据节点性能将参与应用层组播的节点区分为稳定节点和非稳定节点.在构建组播树时,稳定节点作为组播树的内部节点(中转节点),而非稳定节点只能作为组播树的叶

节点.该协议为每个节点分配了一个反映节点稳定性的参数  $node\_stability$ ,新加入节点的稳定程度是未知的,此时为其稳定性参数分配一个默认值,该参数值会随着时间动态变化. HBM 协议为组播树的每个节点赋予能力状态,节点能力状态共有 3 种,即断开状态( $disconnected$ )、叶节点状态( $leaf\_only$ )和中转节点状态( $transit\_possible$ ).节点加入位置由其能力状态决定,而节点能力状态取决于它的能力值  $Ncap$ ,计算节点  $Ncap$  的方法为  $Ncap(node) = f(use\_desires, node\_stability, RP\_param)$ .其中参数  $RP\_param$  表示 RP 给节点赋予的能力状态.当所有节点都希望成为叶节点时,则由 RP 强制某些成员成为中转节点.通过比较节点的  $Ncap$  值和相关阈值确定节点的能力状态:如果  $Ncap(node) \in [0, \alpha)$ ,则该节点将不能加入组播树;如果  $Ncap(node) \in [\alpha, \beta)$ ,则该节点将成为叶节点;如果  $Ncap(node) \in [\beta, 1]$ ,则该节点成为中转节点. HBM 协议的组播树构造机制主要考虑了节点的稳定性.

通过对超过 1000 万条真实视频直播日志进行分析,Luo 等人验证了文献[40]提出的用户在线时间符合对数正态分布的结论,同时发现用户平均剩余在线时间随着用户在线时间的增大而增大,并因此提出了一种适用于视频直播的低中断频率组播树生成算法<sup>[35]</sup>.该算法在组播树构造过程中遵循两个原则:(1)减少在线时间短的节点的子孙数量;(2)让较多子节点的节点靠近根节点,使构建的树尽量矮而宽.模拟实验结果表明,与随机、最小深度和最长持续时间算法相比,该算法能够达到较小的节点平均深度和累积中断次数.

Tan 等人提出的 ROST 算法<sup>[41]</sup>综合考虑了节点度和已在线时间.作者首先分析了单独考虑节点度或节点已在线时间算法<sup>[33]</sup>的缺点,如图 4(b)和图 4(c)所示分别为只考虑节点带宽和节点已在线时间生成的组播树(分别被称为 BO 树和 TO 树).BO 树使得节点离开后受影响的节点平均数目较小,然而维护该树需要周期性调整节点位置.TO 树可以较好地预测成员节点的生命周期,但通常较高,会增大节点失效的相关性.ROST 算法的基本思想是由节点的带宽时间乘积 BTP(Bandwidth-Time Product)决定节点在树中的位置,邻居节点通过比较 BTP 值而决定是否交换彼此的位置.通过将 BTP 值大的节点交换到组播树的高层而得到相对稳定的组播树.与 BO 树和 TO 树算法相比,ROST 算法提高了应用层组播稳定性,但存在的缺点主要为:(1)直接将节点带宽和已在线时间的乘积值作

为节点位置的決定因素,没有考虑带宽和已在线时间对组播稳定性影响的权重是否相同;(2)作者默认高度低的组播树延迟较小,这对于 IP 组播树而

言可能是正确的,然而对于应用层组播树而言可能是错误的。

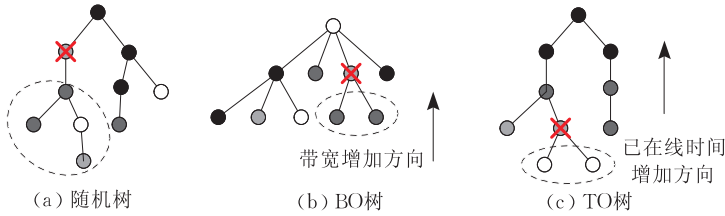


图 4 3 种类型的组播树示例

Bishop 和 Rao 等人研究了异构环境中的组播稳定性问题,设计出了优先级策略并利用真实的应用层组播记录进行评估,同时还研究了各种策略之间的权衡问题<sup>[34]</sup>.其优先级策略分为基于节点度的策略、基于节点已在线时间的策略和综合策略.综合策略将 A 节点对 B 节点 *DegreeRatio* 和 *AgeRatio* 分别定义为  $Degree(A)/Degree(B)$  和  $Age(A)/Age(B)$ ,当两者都大于 1 时,则 A 的优先级高于 B;如果两者不同时大于 1,那么当  $DegreeRatio > AgeRatio^{-p}$  时( $p$  为参数,实验中设置为 0.01),则认为 A 的优先级高于 B.评估结果显示,基于节点度的策略优势明显,而基于节点已在线时间的策略由于改变节点位置比较频繁,其优势不明显.综合策略在祖先节点发生变化时近似于基于节点已在线时间的策略,因此其优势不明显.

针对基于 DHT 的应用层组播,文献[42]提出了 3 种利用节点的已在线时间属性优化组播稳定性的方法:(1)SMO(Senior Member Overlay).该方法将已在线时间大于特定阈值的节点组织为一个专门的覆盖网,同时规定只有此覆盖网中的节点才能充当组播树的内部节点;(2)LNS(Longer-lived Neighbor Selection).该方法作用于某些可以灵活选择邻居节点的 DHT 算法,它使每个节点选择相对稳定的节点作为其邻居节点,以此提高数据传输路径的稳定性;(3)RRS(Reliable Route Selection).该方法作用于可以灵活选择下一跳节点的 DHT 算法,它使节点从多个可选节点中选择较为稳定的下一跳节点.尽管模拟实验表明了该机制的有效性,但由于没有考虑节点度,因此难以将这些方法应用于视频直播等带宽敏感型应用.

缩小节点离开的平均影响范围是提高应用层组播的稳定性的重要措施之一,然而目前的研究还处于起步阶段,存在的不足主要是缺乏应用层组播稳定性问题的理论建模与分析,从而导致大部分相关研究对组播稳定性问题的认识还非常感性,对于什

么是应用层组播稳定性?其衡量标准是什么?怎样的组播树结构稳定性最高?诸如此类问题并没有科学的结论.例如,从节点度属性而言,通常认为短而宽的树稳定性较高,而从节点行为属性而言,通常认为在线时间长的节点处于组播树顶层位置时组播树的稳定性较高.虽然文献[34-35]和文献[41]都已提出了综合考虑节点度和行为属性的高稳定性组播树构造算法,然而哪种综合策略更科学和有效尚无统一的结论.

5.2 基于多树的组播

多树结构降低了节点离开的影响程度,因而将基于多树的应用层组播归类为缩小影响范围的方法.与大多数应用层组播协议不同,Castro 等人提出的 SplitStream<sup>[28]</sup>协议是一种基于多树的协议.SplitStream 协议的基本思想是在所有节点间构造多棵组播树,各个组播树的根节点都是数据源节点,它们共同构成组播的数据转发拓扑.利用多描述编码 MDC(Multiple Description Coding)技术<sup>[43]</sup>将组播数据内容编码为多个数据带(data stripe),每个数据带在不同的组播树上传输,参加组播的节点加入到尽量多的组播树中以获得更高质量的组播数据.同一节点在各个组播树中具有不同的位置,某棵树的内部节点可能成为另一棵树的叶节点.图 5 所示为基于多树的应用层组播数据转发拓扑结构,其中节点 1 为源节点,带 1 的组播树的内部节点集合为 {2, 3, 4, 5, 6, 7, 8},其叶节点集合为 {9, 10, 11, 12, 13, 14, 15, 16}.带 2 的组播树的内部节点集合为

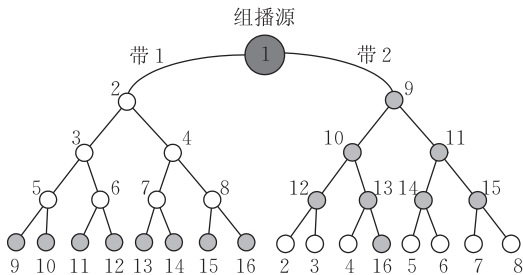


图 5 基于多树的应用层组播数据转发拓扑示例

{9, 10, 11, 12, 13, 14, 15}, 其叶节点集合为 {2, 3, 4, 5, 6, 7, 8, 16}. 基于多树的应用层组播比较适合流媒体数据的组播传输. Bullet<sup>[9]</sup> 和 CoopNet<sup>[10]</sup> 等协议也采用了基于多树的应用层组播思想.

采用多树结构为组播传输的数据转发拓扑提高了应用层组播的稳定性, 其原因是: (1) 由于特定节点在某些树中是内部节点, 而在其它树中是叶节点, 因此节点的离开只对部分树中的部分节点造成影响. (2) 由于单个节点从多个父节点接收不同数据带中的组播数据, 某个父节点离开不影响节点从其它父节点接收数据, 因而只是减少了该节点接收到的数据带数, 造成流媒体播放质量的暂时下降, 而不会中断该节点的流媒体播放过程. 基于多树的应用层组播的缺点是维护多棵组播树的开销较大, 如何实现多源同步也是难题.

表 1 总结了上述通过缩小节点变动影响范围而提高应用层组播稳定性的各种技术. 从表 1 可见, 在

表 1 缩小节点变动影响范围提高应用层组播稳定性的算法/机制

算法/机制	提出时间	主要优缺点
HBM 协议中的机制 <sup>[6]</sup>	2001	能够阻止短生命周期节点成为组播树的内部节点, 但准确评估节点能力值较困难, 节点能力状态划分过于简单
文献 <sup>[35]</sup> 的算法	2006	能够达到较小的节点平均深度和累积中断次数, 但忽略了祖先节点不稳定性的累积问题
ROST 算法 <sup>[41]</sup>	2007	性能优于带宽优先算法和在线时间优先算法, 但没有充分考虑组播树延迟, 因此组播树延迟可能比较大
文献 <sup>[34]</sup> 的算法	2006	基于节点度的策略优势明显, 基于节点已在线时间及其综合策略的优势并不明显. 综合策略的综合方法缺乏科学性
文献 <sup>[42]</sup> 的算法	2007	能够显著提高应用层组播的稳定性, 但没有考虑节点度, 因此难以将其应用于视频直播等应用
基于多树的组播 <sup>[28]</sup>	2003	能够有效提高面向流媒体的应用层组播对节点异构性的适应能力, 但必需结合 MDC 使用, 维护多树结构和多源同步的控制开销较大

上述各种算法和机制中, 基于多树的组播紧密结合流媒体应用的特点, 有效地提高了应用层组播的稳定性. 由于应用层组播是直接面向应用的组播传输技术, 因此如何在满足应用需求的同时, 充分结合应用特点而提高组播传输的稳定性值得深入研究.

## 6 缩短节点离开后组播树恢复时间的技术

在应用层组播过程中, 组播树内部节点的离开

带来的问题就是如何使受影响节点快速地重新加入应用层组播树中, 从而恢复接收组播数据. 缩短恢复时间是提高应用层组播稳定性的重要措施之一. 恢复应用层组播树分为两个步骤: 第 1 步, 发现节点的离开行为. 由于需要退出的节点会主动通告离开消息, 因此节点的离开行为会迅速触发受影响节点重新加入组播树过程. 失效节点不会通告任何消息, 因此需要失效检测机制. 第 2 步是重构组播树. 组播树的重构策略可以分为两种, 即前向式 (Proactive) 和后向式 (Reactive). 前向式重构策略未雨绸缪, 在节点离开之前已经计算出合理的组播树重构方案. 后向式树重构策略在节点离开后, 通常受影响节点需要联系多个仍然在组播树中的节点从而选择合适的父节点. 可见, 缩短组播树恢复时间需要快速失效检测机制和高效的树重构策略. 而树重构策略又分为前向式树重构和后向式树重构.

### 6.1 节点失效检测

目前, 针对缩短节点离开后组播树恢复时间的研究主要集中于树重构机制, 而对失效检测机制的研究较少. 根据失效检测过程中节点间消息类型的不同, 节点失效检测分为心跳机制和探测机制. 在心跳机制中, 各个节点周期性地向邻居节点发送心跳消息; 而在探测方法中, 节点向邻居节点发送探测消息, 然后邻居节点向该节点发送应答消息表明自己的存在. 大部分的应用层组播协议和系统都采用基本心跳机制, 即如果节点在长时间内没有接收到某个已知邻居节点的心跳报文, 则认为该节点失效. 在网优先的应用层组播协议中, 节点的失效可能会导致 Mesh 出现分割, 为此在检测到节点失效后, 需要进一步判断是否出现 Mesh 分割. 例如在 Narada 协议<sup>[4]</sup>中, 每个节点维护的超时节点队列中存放经过某一时限内没有收到其心跳报文的节点, 如果 Mesh 被分割, 那么被分割的节点都会出现在超时节点队列中. 节点周期性地从队列头部取出节点对其进行探测以判断该节点是否失效, 或者出现 Mesh 分割. 如果发现 Mesh 分割, 则通过给两个出现在不同分割子网中的节点之间, 增加一条虚拟链路以修复被分割的 Mesh. 设计高效的失效检测机制需要对检测时间、误检概率和控制开销进行权衡.

在基本心跳机制中, 每个节点独立地对其父节点和各个子节点进行周期性探测, 并根据接收到的应答消息情况独立地判断目标节点是否失效. 为了提高失效检测的效率, 文献<sup>[44]</sup>对基本的心跳机制进行改进, 提出了一种合作式检测 (Cooperative Detection) 机制. 在该机制中, 同一节点的多个检测



节点(其父节点和各个子节点)通过共享心跳消息丢失信息而进行合作式的失效判断,从而缩短检测时间,该机制的缺点是会增加误检概率.在实际应用层组播环境中,根据需求,合理配置相关参数.

文献[45]深入研究了具有不同的信息共享力度、信息交互过程和节点状态维护方法的各种失效检测机制的性能.尽管该文献针对的是基于探测机制的覆盖网络,但它对于设计应用层组播的失效检测机制具有重要的启发意义.

6.2 前向式树重构

6.2.1 PRM 算法

Banerjee 等人提出的 PRM(Probabilistic Resilient Multicast)算法<sup>[46]</sup>的基本思想是:每个节点除了正常参与基于应用层组播树的数据转发外,还随

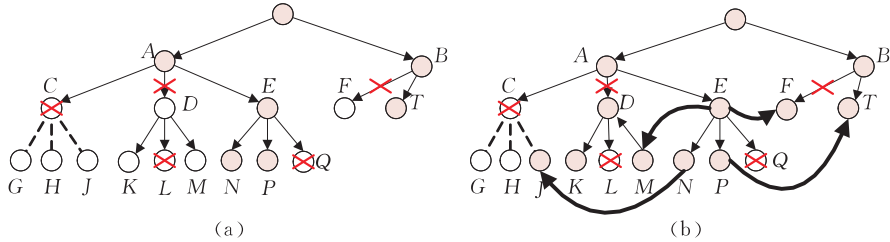


图 6 PRM 算法基本思想

PRM 算法是一种基于数据转发平面的组播树重构策略,对组播树结构进行扩展和改进.该算法的优点是在节点频繁失效的情况下可以保证较高转发率,从而提高了组播稳定性.其缺点是需要为节点增加重复报文检测机制,同时造成一定程度的带宽浪费.

6.2.2 Yang 算法

针对 PRM 算法的缺点,Yang 等人提出了一种作用于控制平面的前向式树重构方法<sup>[47]</sup>,本文称其

机选择少量节点(如 1~3)并以小概率(如 0.01~0.03)额外地为这些节点转发数据.额外增加的数据转发路径为组播树的快速重构提供保障.如图 6 所示,当应用层组播树(如图 6(a)所示)的连接 $\langle A,D \rangle$ 和 $\langle B,F \rangle$ 以及节点 C、L 和 Q 失效时,则集合 $\{D,F,G,H,J,K,M\}$ 中的节点被迫中断组播数据接收.PRМ 算法对基于树结构的组播转发方法作了两点补充(如图 6(b)所示):(1)当接收到报文的首个拷贝时,节点沿着所有其它边(除接收数据的边)转发报文;(2)节点选择少量节点并概率地向其转发数据报文,如图中黑粗线所示.该组播转发策略导致节点可能收到多个相同报文(例如节点 T 收到 P 和 B 发送的相同报文),所以该算法为每个节点增加了重复报文检测机制.

为 Yang 算法.与 PRM 算法不同,Yang 算法作用于路由控制平面,其基本思想是每个非叶节点必须提前计算出它失效之后的组播树重构方案,即为所有子节点计算备用父节点.因此,当非叶节点离开时,它的每个子节点能够迅速地加入到备用父节点,从而实现组播树的快速重建.如图 7 所示,节点 5 为子节点 $\{8,9,10\}$ 计算出的备用父节点分别为 $\{9,2,16\}$ .

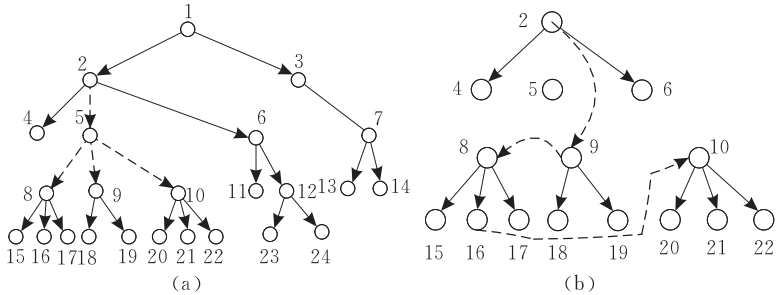


图 7 Yang 算法基本思想

Yang 算法要解决的问题可以形式化为:假设节点  $a_i$  有  $n$  个子节点 $\{c_0,c_1,\cdots,c_{n-1}\}$ ,从组播树根节点到  $a_i$  节点的路径为 $a_0,a_1,\cdots,a_{i-1}$ .问题为寻找由节点  $a_{i-1}$  和以  $c_0,c_1,\cdots,c_{n-1}$  为根的子树构成的树,使该树满足节点度约束且代价最小. Yang 算法首先从  $a_i$  所有子节点中选择一个到  $a_i$  父节点  $a_{i-1}$  开销最

小的节点  $c_i$ ,将  $a_{i-1}$  作为  $c_i$  的备用父节点.然后把  $c_i$  及其子孙节点放入集合 A 中,把  $a_i$  剩余的子节点放入集合 B 中,然后从集合 A 和 B 中选择连接开销最小的一对节点,并在这两个节点间建立备用连接,直到集合 B 为空为止.

由于任意节点的动态行为都会触发 Yang 算法

重新计算备用父节点,所以当节点动态性较高时,Yang 算法的计算量和控制开销都较大.由于 Yang 算法将各个节点的备用父节点选择范围限定在了父节点和兄弟节点范围内,因此不利于节点度的充分利用和组播树延迟性能优化.为此,文献[48]进一步完善了 Yang 算法,将备用父节点的搜索范围扩大到整个组播树,即允许节点选择任意非饱和(具有剩余度)节点作为备用父节点.本文称其为 Yang<sup>+</sup> 算法.同时,Yang<sup>+</sup> 算法还给出了将备用父节点的本地选择策略和全局选择策略有机结合起来的方法.实验结果表明,Yang<sup>+</sup> 算法具有比 Yang 算法更优的组播树延迟方面的性能.

### 6.2.3 Kusumoto 算法

鉴于 Yang 算法计算量和控制开销大的缺点,文献[49]提出了一种为组播树的重构维护备份路由的方法,即 Kusumoto 算法.通过每个节点预留节点度,该方法能够方便地为各个节点在祖父节点和其子节点间建立备份路径.从而有效地降低了选择备用父节点时的计算量和组播树恢复过程中的额外控制开销.但是该算法的缺点也是很明显的,例如它不能高效利用节点的剩余度,从而使得组播树高度较高,组播树延迟较大.

### 6.2.4 RVL 算法

RVL(Redundant Virtual Link)算法<sup>[50]</sup>是由 El-Sayed 等人提出的一种支持应用层组播树快速重构的算法,该算法在组播树结构的基础上增加冗余的虚拟连接作为备用连接.当节点离开后,受影响节点迅速利用备用连接恢复组播数据接收.RVL 算法将应用层组播树节点区分为叶节点和中转节点,备用连接增加策略分为 5 种:策略 I 是在任意节点间增加任意数目的备用连接;策略 II 是的中转节点可以与任意数目的备用连接相连,而叶节点只能与一条备用连接相连;策略 III 是任意两个非叶节点间增加任意数目的备用连接;策略 IV 是仅在中转节点间增加任意数目的备用连接;策略 V 是仅在中转节点和叶节点之间增加备用连接,且每个叶节点只能与一条备用连接相连.

对策略 I 而言,大部分备用连接处于叶节点之间,且某些叶节点与多条备用连接相连.因此,策略 I 适合于所有组成员的处理能力和通信能力相似的情况.与策略 I 相反,策略 IV 只产生了一条备用连接,叶节点没有与备用连接相连.RVL 算法通过在节点之间增加备用连接提高应用层组播的稳定性,但是该算法为集中式算法,由 RP 节点增加冗余连接,因此该算法适用范围有限.

有不少协议采用与 RVL 算法类似的思想,例如 Wang 等人提出的 TMesh 协议<sup>[51]</sup>,在节点之间增加冗余虚拟连接(文献[51]中称之为 Shortcut)的方法.每个节点独立地选择能够减小相对延迟开销或增加链路利用率的虚拟连接增加到组播树中. Shi 等人提出层次型冗余连接增强策略<sup>[52]</sup>,该策略增加的备用连接分为层间冗余连接 INTER-RL (INTER-level Redundant Links)与层内冗余连接 INTRA-RL (INTRA-level Redundant Links).冗余连接存在于节点与父节点的兄弟节点间,层内冗余连接存在于节点与兄弟节点间.

RVL 及其改进算法缩短了节点离开后组播树的恢复时间.这类算法虽然增加了维护备用连接的控制开销,但是在节点离开较为频繁的组播环境中,提高应用层组播稳定性的效果比较明显. RVL 及其改进算法主要不足是还不能回答如何增加备用连接既能解决单点失效问题,又能维持较小的额外控制开销.

### 6.2.5 LER 算法

为了解决应用层组播的可靠传输问题,香港科技大学的 Wong 等人提出了 LER (Lateral Error Recovery)算法<sup>[53]</sup>.他们认为应用层组播树结构利于降低延迟,但不利于丢失数据时的错误恢复.上游节点发生数据错误,那么其下游节点都将受到影响,这种数据错误的相关性使得从上游节点请求数据重传将可能失效,该问题被称为错误相关性问题.在 LER 算法中,各个节点分属于不同的平面,每个平面的节点形成独立的组播树.由于不同平面的节点间的错误相关性较低,因此节点从其它平面的近距离节点恢复数据将更有效. LER 算法的错误恢复方法如图 8 所示,源节点(origin)向各个平面的源节点发送组播数据,初始时节点 B 选择其它平面的节点 A 和 C 作为恢复邻居(recovery neighbors).当发生错误时,节点 B 从恢复邻居中选择任意节点.

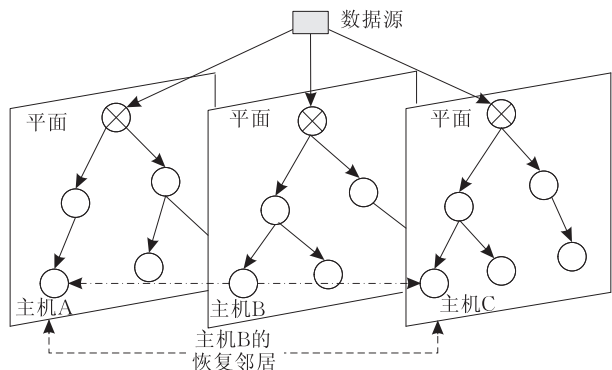


图 8 LER 算法的错误恢复

虽然 LER 算法主要针对丢失数据时的错误恢复问题,但它也是一种支持快速重组组播树的算法,因此可以提高应用层组播的稳定性.模拟结果表明,LER 算法比 PRM 算法具有更短的平均恢复时间和更低的控制开销.

6.3 后向式树重构

Bawa 等人全面地研究了后向式树重构策略<sup>[54-55]</sup>.作者假设每个节点都拥有父节点和根节点的地址信息.当节点离开时,其子节点或子孙节点为了重新加入组播树而联系的节点被称为目标节点.根据需要重新加入节点的范围和目标节点的不同,后向式树重构策略分为 4 种.假设  $v$  表示离开的节点,则此 4 种策略描述为:(1) RTA(RooT-All).节点  $v$  的所有子孙节点通过联系根节点而加入到组播树;(2) GFA(GrandFather-All).节点  $v$  的所有子孙节点通过联系  $v$  的父节点而加入到组播树;(3) RT(RooT).节点  $v$  的子节点通过联系根节点而加入到组播树,同时以  $v$  的子节点为根的子树也就加入到了组播树;(4) GF(GrandFather).节点  $v$  的子节点通过联系  $v$  的父节点而加入到组播树,同时以  $v$  的子节点为根的子树也就加入到了组播树.

由于 RTA 策略需要为所有受影响节点重新计算位置,因此易于保持组播树的平衡性. GFA 策略将节点重新加入的位置限制在以  $v$  的父节点为根的子树,因此避免了根节点接收多个节点加入请求时负载过重的问题. RT 和 GF 策略的优点是把节点离开的影响限制在局部范围,因此具有较优的综合性能,是实现组播树快速重构的理想选择.

6.4 基于环的组播

环结构有利于节点离开后组播数据传输的快速恢复,因此也是一种缩短恢复时间的技术. Sobeih 等人提出了一种基于环结构的应用层组播协议 VRing<sup>[56]</sup>,该协议首先将所有成员节点连接成一个主环结构,然后通过增加冗余虚拟连接而将节点组织成备用环结构.如图 9 所示,路径  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0$  为主环(实线所示),而路径  $0 \rightarrow 3 \rightarrow$

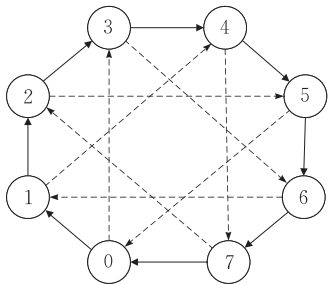


图 9 VRing 结构

$6 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow 5 \rightarrow 0$  构成备用环(虚线所示).在组播数据传输过程中,节点将从主环接收到的数据分别转发给主环和备用环的节点,而从备用环接收到的数据只转发给备用环的节点.

除 VRing 之外,Wang 等人还提出了一种基于虚拟多环的组播方案<sup>[57]</sup>,多环的构造只需要本地信息,而不需要全局知识.每个节点有两条到邻居节点的备用连接,以提高应用层组播传输的稳定性.与树结构相比,环结构具有内在可靠性和容错特征,双环结构还能自动提供冗余备份和避免单点失效问题.环结构的缺点是可扩展性差,传输延迟较大,因此它只适合于较小规模的应用层组播.

6.5 流补丁机制

6.5.1 集中式流补丁

Guo 等人提出的集中式流补丁机制<sup>[58]</sup>通过两种技术保证发生节点离开后组播视频流的连续性:(1)服务器提供一组时间移位(Time-shifted)的流,使受影响节点重新加入时,得到移位的视频流.(2)使用流补丁方法以便受影响节点可以赶上正常流的进度.该机制中的所有节点在流回放(Playback)之前缓冲一定时间的原始视频流数据,因此即使用户节点从组播树中断开后,在重新加入到组播树时可以播放缓冲区的视频数据.服务器维持一条常规流和  $n$  条补丁流.用户节点经过  $t$  时间重新加入原始流后,选择一个延迟时间和  $t$  最接近的并大于  $t$  的补丁流.原始流用来赶上视频直播的进度,补丁流用来修补重新加入期间丢失的数据.

图 10 所示为集中式流补丁过程中应用层组播树的变化情况.最初,所有节点通过原始的组播树(如图 10(a)所示)接收原始流.在时间  $t_0$ ,节点 A 离开,因此以节点 Y 和 Z 为根的子树的组播连接被中断(如图 10(b)所示).因此,节点 Y 和 Z 分别向 S 发送包含所要请求的原始流和补丁流的“Rejoin”消

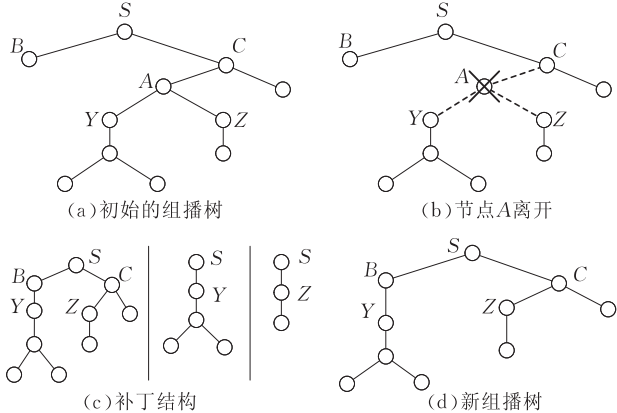


图 10 流补丁过程中应用层组播树的变化

息,然后, $S$ 指定节点 $B$ 和 $C$ 分别为 $Y$ 和 $Z$ 的新父节点以获得原始流,同时分别为 $Y$ 和 $Z$ 分配合适的补丁流(如图10(c)所示).当流补丁过程结束时,补丁流形成的组播树将被释放,此时只需要维护一棵新的应用层组播树(如图10(d)所示).

### 6.5.2 分布式流补丁

集中式流补丁机制在并发用户数目较小的情况下效果良好,但当并发用户数目较大且用户节点加入、退出和失效频繁发生的情况下,服务器将会不堪重负,成为组播系统性能的瓶颈,此时节点的加入或重新加入延迟将会很大.同时,服务器要为原始流和多条补丁流维护不同的组播树,其控制开销和实现难度都较大.

为了缓解集中式流补丁机制给服务器带来的负担,Guo等人进一步提出了分布式流补丁(cooperative patching)机制<sup>[59]</sup>,该机制主要通过用户节点间的协作进行流补丁操作.与集中式流补丁机制相比,分布式流补丁机制也需要用户节点在回放前对数据进行一定时间的缓冲,不同的是回放后的数据也不会立即被丢弃,而是要缓存一定时间.每个用户节点维护着一个备用流补丁节点集合.当受影响节点重新加入时,从备用流补丁节点集合中寻找一个合适节点进行流补丁.分布式流补丁机制扩展性较好,对节点的动态行为适应性强.但是为了保证备用流补丁节点的可用性,节点需要周期性的额外发送探测报文以判断流补丁节点是否可用,因此控制开销较大.同时,用户节点需要保存回放点前后一段时间的数据,因此会占用更多的存储空间.

流补丁机制降低节点离开后造成的组播数据接收损失,变相地增加了受影响节点接收组播数据的连续性,取得了与缩短组播树重构时间相同的效果.当受影响节点重新加入延迟不是很大时,流补丁机制可以完全补偿节点与组播树连接中断期间的数据损失.

### 6.6 小结

表2是缩短节点离开后组播树恢复时间提高应用层组播稳定性的研究的小结.

由表2可见,缩短节点离开后组播树恢复时间的方法,尤其是组播树的快速重构问题得到了不少的研究关注.但从表1和表2所列研究成果发表时间来看,近年来对减小节点离开的影响范围的研究逐步加强.失效检测的效率对平均的组播树恢复时间有着重要的决定作用,但目前对快速失效检测机制的研究重视程度不够.

表 2 缩短组播树恢复时间提高应用层组播稳定性的算法/机制

算法/机制	提出时间	主要优缺点
合作式失效检测 <sup>[44]</sup>	2005	缩短了失效检测时间,但略微提高了误检概率,增加了控制开销
PRM 算法 <sup>[46]</sup>	2003	在高节点失效率时,保证较高转发率,但需要重复报文检测机制,且重复报文浪费网络带宽
Yang 算法 <sup>[47]</sup>	2004	能有效地缩短组播树恢复时间,能保证较低的组播树延迟,但计算量和控制开销较大,不能充分利用节点度
Yang <sup>+</sup> 算法 <sup>[48]</sup>	2005	解决了 Yang 算法不能充分利用节点度的缺点
Kusumoto 算法 <sup>[49]</sup>	2005	能有效地缩短组播树恢复时间,计算量和控制开销较 Yang 算法小,但对节点度的利用率较低
RVL 算法 <sup>[50-51]</sup>	2001	支持应用层组播树的快速重构,但增加了维护备用连接的控制开销
LER 算法 <sup>[53]</sup>	2004	与 PRM 算法相比,具有更短的平均恢复时间和更低的控制开销
后向式树重构策略 <sup>[54-55]</sup>	2001	RTA 和 GFA 策略相比,RT 和 GF 策略的综合性性能(平均恢复时间和控制开销等)较优
基于环的组播 <sup>[56-57]</sup>	2004	能够自动提供冗余备份和避免单点失效问题,但可扩展性较差,传输延迟较大
流补丁机制 <sup>[58-59]</sup>	2004	有效地降低了节点退出或失效造成的数据接收损失,但控制开销和实现难度较大,只适合于流媒体.

除了研究应用层组播稳定性提高技术外,也有不少针对组播稳定性模型理论的研究.例如,Zhang等人对组播树的拥塞瓶颈建立了链路拥塞模型和独立度模型<sup>[60]</sup>.Shi等人用随机模型分析了层次型组播拥塞控制的稳定性问题<sup>[52,61-62]</sup>.Xu等人为应用层组播的稳定性提出了一种随机模型,并分析了组播树结构因素对其稳定性的影响<sup>[63]</sup>.

## 7 总结与展望

组播传输技术是组通信应用的关键支撑技术,然而传统 IP 组播的广泛部署却举步维艰.应用层组播的出现解决了传统 IP 组播的部署性问题,但也面临着新的挑战.与以路由器为组播内部节点的 IP 组播不同的是,应用层组播树的节点全为用户节点.用户节点行为的动态性和自身的脆弱性给组播树的稳定性带来的严重影响.为了提高应用层组播的稳定性,相关研究已经提出了多种算法和机制,它们主要从3个方面着手,即降低节点离开的频率、缩小节点离开的影响范围以及缩短节点离开后组播树恢复时间.



随着对应用层组播研究的深入,其稳定性问题会越来越受到研究者的重视<sup>[64-65]</sup>.我们认为如下方面有待进一步研究:(1)模型与理论方面.文献[63]是目前提出的较为完善的应用层组播稳定性模型,但是它也存在着明显的缺点,例如它只是一个关于应用层组播树结构的静态模型,即它只是对组播树个体节点的退出或失效对其它节点的影响建模,而没有考虑在应用层组播节点动态加入、退出或失效过程中的稳定性问题.(2)算法与机制方面.应用层组播用户行为表现出许多方面的统计学特性,例如节点的组播会话时间长度分布呈现重尾现象<sup>[33,40,66]</sup>等.如何充分利用这些行为特征来提高应用层组播的稳定性,目前这方面的研究还处于起步阶段.(3)协议设计方面.缺乏一个充分考虑组播稳定性问题的应用层组播协议.现有研究提出的协议大多只是运用某种机制或算法对应用层组播某个方面的稳定性进行加强,没有明确提出以构建和维护稳定性高的组播树为目标的协议.为此,设计高稳定性应用层组播协议是研究如何提高应用层组播稳定性的目的所在,也是下一步的研究目标.

## 参 考 文 献

- [1] Deering S. Multicast routing in Internetworks and extended LANs. *ACM Transactions on Computer System*, 1990, 8(2): 85-110
- [2] Diot C, Levine B, Lyles J, Kassem H, Balensiefen D. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 2000, 14(1): 78-88
- [3] Kevin C, Almeroth. The evolution of multicast: From Mbone to inter-domain multicast to Internet2 deployment. *IEEE Network*, 2000, 14(1): 10-20
- [4] Chu Y H, Rao S G, Zhang H. A case for end system multicast. *IEEE Journal on Selected Areas in Communications*, 2002, 20(8): 1456-1471
- [5] Pendarakis D, Shi S, Verma D, Waldvogel M. ALMI: An application level multicast infrastructure//*Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems*. San Francisco, California, 2001: 5-15
- [6] Roca V, El-Sayed A. A host-based multicast (HBM) Solution for group communications//*Proceedings of the 1st International Conference on Networking Colmar*. France, 2001: 610-619
- [7] Yatin C. Scattercast: An architecture for Internet broadcast distribution as an infrastructure service [Ph. D. dissertation]. University of California, Berkeley, USA, 2002
- [8] Sushant J, Ratul M, David W, Gaetano B. Scalable self-organizing overlays. University of Washington, USA: Technology Report UW-CSE 02-06-04, 2002
- [9] Kostic D, Rodriguez A, Albrecht J, Vahdat A. Bullet: High bandwidth data dissemination using an overlay mesh//*Proceedings of the 19th ACM SOSP*. Bolton Landing, NY, USA, 2003: 282-297
- [10] Padmanabhan V N, Wang H J, Chou P A, Sripanidkulchai K. Distributing streaming media content using cooperative networking//*Proceedings of the 12th ACM NOSSDAV*. Miami, Florida, USA, 2002: 177-186
- [11] Demers A J, Greene D H, Hauser C, Irish W, Larson J, Shenker S, Sturgis H E, Swinehart D C, Terry D B. Epidemic algorithms for replicated database maintenance//*Proceedings of the 6th ACM Symposium on Principles of distributed computing*. Vancouver, British Columbia, Canada, 1987: 1-12
- [12] Zhang X, Liu J, Li B, Yum T P. CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming//*Proceedings of the IEEE INFOCOM 2005*. Florida, USA, 2005: 2102-2111
- [13] Rejaie R, Stafford S. A framework for architecting peer-to-peer receiver-driven overlay//*Proceedings of the 14th ACM NOSSDAV*. Cork, Ireland, 2004: 42-47
- [14] Magharei N, Rejaie R. Prime: Peer-to-peer receiver-driven mesh-based streaming//*Proceedings of the IEEE INFOCOM*. Alaska, USA, 2007: 1415-1423
- [15] Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users//*Proceedings of the IEEE INFOCOM*. New York, USA, 2002: 1366-1375
- [16] Mathy, Canonico R, Hutchison D. An overlay tree building protocol//*Proceedings of the Networked Group Communication*. London, UK, 2001: 76-87
- [17] Helder D A, Jamin S. End-host multicast communication using switch-trees protocols//*Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*. Berlin, Germany, 2002: 419-429
- [18] Jannotti J, Gifford D K, Johnson K L, Kaashoek M F, Toole W O. Overcast: Reliable multicast with an overlay network//*Proceedings of the 4th USENIX Symposium on OSDI*. San Diego, CA, 2000: 64-72
- [19] Minseok K, Sonia F. Topology-aware overlay networks for group communication//*Proceedings of the 12th International Workshop on ACM NOSSDAV*. Miami, Florida, USA, 2002: 127-136
- [20] Li Z, Mohapatra P. Hostcast: A new overlay multicasting protocol//*Proceedings of the IEEE International Conference on Communications (ICC)*. Alaska, USA, 2003: 702-706
- [21] Heffeeda M, Habib A, Botev B, Xu D, Bhargava B. PROMISE: Peer-to-peer media streaming using Collect-Cast//*Proceedings of the 11th ACM International Conference on Multimedia*. Berkeley, CA, USA, 2003: 45-54
- [22] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast//*Proceedings of the ACM SIGCOMM*. Pittsburgh, PA, 2002: 205-217
- [23] Tran D A, Hua K A, Do T T. A peer-to-peer architecture for media streaming. *IEEE Journal on Selected Areas in Communications*, 2004, 22(1): 121-133



- [24] Jorg L, Michael N. Application-layer multicast with Delaunay triangulations. *IEEE Journal on Selected Areas in Communications*, 2002, 20(8): 1472-1488
- [25] Zhuang S Q, Zhao B Y, Joseph A D, Katz R H, Kubiatiowicz G D. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination//*Proceedings of the International Workshop on ACM NOSSDAV*. New York, USA, 2001: 11-20
- [26] Castro M, Druschel P, Kermarrec AM, Rowstron A. Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications*, 2002, 20(8): 1489-1499
- [27] Ratnasamy S, Handley M, Karp R, Shenker S. Application-level multicast using content-addressable networks//*Proceedings of the Networked Group Communication*. London, UK, 2001: 14-29
- [28] Castro M, Druschel P, Kermarrec A M, Nandi A, Rowstron A, Singh A. SplitStream: High-bandwidth multicast in cooperative environments//*Proceedings of the 19th ACM SOSP*. Bolton Landing, NY, USA, 2003: 298-313
- [29] Hosseini M, Ahmed D T, Shirmohammadi S, Georganas N D. A survey of application-layer multicast protocols. *IEEE Communications Surveys & Tutorials*, 2007, 9(3): 58-74
- [30] Xie S S, Li B, Keung G Y, Zhang X Y. Coolstreaming: Design, theory and practice. *IEEE Transactions on Multimedia*, 2007, 9(8): 1661-1671
- [31] Li B, Xie S S, Yang Q, Keung G Y, Lin C, Liu J C, Zhang X Y. Inside the new coolstreaming: Principles, measurements and performance implications//*Proceedings of the IEEE INFOCOM*. Phoenix, AZ, USA, 2008: 1031-1039
- [32] Hei X, Liang C, Liang J, Liu Y, Ross K W. A measurement study of a large-scale P2P IPTV system. *IEEE Transactions on Multimedia*, 2007, 8(9): 1672-1687
- [33] Sripanidkulchai K, Ganjam A, Maggs B, Zhang H. The feasibility of supporting large-scale live streaming applications with dynamic application end-points//*Proceedings of the ACM SIGCOMM*. Portland, Oregon, USA, 2004: 107-120
- [34] Bishop M, Rao S, Sripanidkulchai K. Considering priority in overlay multicast protocols under heterogeneous environments//*Proceedings of the IEEE INFOCOM*. Barcelona, SPAIN, 2006: 1-13
- [35] Luo Jian-Guang, Zhao Li, Yang Shi-Qiang. An algorithm of constructing ALM tree based on user behavior analysis. *Journal of Computer Research and Development*, 2006, 43(9): 1557-1563 (in Chinese)  
(罗建光, 赵黎, 杨士强. 基于用户行为分析的应用层组播树生成算法. *计算机研究与发展*, 2006, 43(9): 1557-1563)
- [36] Lao L, Cui J H, Gerla M, Maggiorini D. A comparative study of multicast protocol: Top, bottom, or in the middle? //*Proceedings of the IEEE INFOCOM*. Miami, USA, 2005: 2809-2814
- [37] Chawathe Y, McCanne S, Brewer E. RMX: Reliable multicast in heterogeneous environments//*Proceedings of the IEEE INFOCOM*. Tel-Aviv, Israel, 2000: 795-804
- [38] Banerjee S, Kommareddy C, Kar K, Bhattacharjee B, Khuller S. Construction of an efficient overlay multicast infrastructure for real-time applications//*Proceedings of the IEEE INFOCOM*. San Francisco, USA, 2003: 1521-1531
- [39] Lao L, Cui J H, Gerla M. TOMA: A viable solution for large-scale multicast service support//*Proceedings of the IF-TP NETWORKING*. Waterloo, Canada, 2005: 906-917
- [40] Eveline V, Virgilio A, Wagner M, Azer B, Shudong J. A hierarchical characterization of a live streaming media workload//*Proceedings of the ACM SIGCOMM*. Marseille, France, 2002: 117-130
- [41] Tan G, Jarvis S A. Improving the fault resilience of overlay multicast for media streaming. *IEEE Transactions on Parallel and Distributed Systems*, 2007, 18(6): 721-734
- [42] Tan G, Jarvis S A. Stochastic analysis and improvement of the reliability of DHT-based multicast//*Proceedings of the IEEE INFOCOM*. Anchorage, Alaska, USA, 2007: 1423-1432
- [43] Goyal V K. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 2001, 18(5): 74-93
- [44] Yang M K, Fei Z M. Cooperative failure detection in overlay multicast//*Proceedings of the IFIP NETWORKING*. Waterloo, Canada, 2005: 881-892
- [45] Zhuang S Q, Geels D, Stoica I, Katz R H. On failure detection algorithm in overlay networks//*Proceedings of the IEEE INFOCOM*. Miami, FL, USA, 2005: 2112-2123
- [46] Banerjee S, Lee S, Bhattacharjee B, Srinivasan A. Resilient multicast using overlays. *IEEE/ACM Sigmetrics Performance Evaluation Review*, 2003, 31(1): 102-113
- [47] Yang M K, Fei Z M. A proactive approach to reconstructing overlay multicast trees//*Proceedings of the IEEE INFOCOM*. Hong Kong, China, 2004: 2743-2753
- [48] Fei Z M, Yang M K. Restoring delivery tree from node failure in overlay multicast. *IEICE Transactions on Communication*, 2005, E88-B(5): 2046-2053
- [49] Kusumoto T, Kunichika Y, Katto J, Okubo S. Tree-Based application layer multicast using proactive route maintenance and its implementation//*Proceedings of the ACM P2PMMS*. Singapore, 2005: 49-58
- [50] Ayman E S. Application-level multicast transmission techniques over the Internet [Ph. D. dissertation]. INRIA Rhones-Aples, USA, 2004
- [51] Wang W J, Helder D, Jamin S, Zhang L X. Overlay optimizations for end-host multicast//*Proceedings of the 4th International Workshop on Networked Group Communication (NGC)*. Boston, MA, USA, 2002: 121-129
- [52] Shi F, Wu J, Xu K. Stability of a multicast tree in cumulative layered multicast congestion control//*Proceedings of the IPCCC*. Phoenix, Arizona, 2003: 725-731
- [53] Wong K F S, Gary Chan S H, Wong W C. Lateral error recovery for application-level multicast//*Proceedings of the IEEE INFOCOM*. Hong Kong, China, 2004: 2708-2718
- [54] Bawa M, Deshpande H, Garcia-Molina H. Transience of peers and streaming media. *ACM SIGCOMM Computer Communication Review*, 2003, 33(1): 107-112
- [55] Deshpande H, Bawa M, Garcia-Molina H. Streaming live media over a peer-to-peer network. CS Department Stanford University, USA: Technology Report CS-2001-31, 2001

- [56] Sobeih Ahmed, Yurcik William, Hou J. VRing: A case for building application-layer multicast rings (rather than trees)//Proceedings of the IEEE Computer Society's 12th Annual International Symposium on MASCOTS. Vollen-dam, Netherlands, 2004; 437-446
- [57] Wang J, Yurcik W. A survey and comparison of multi-ring techniques for scalable battlespace group communications//Proceedings of the SPIE. Miami, FL, USA, 2005; 139-151
- [58] Guo M, Ammar M H. Scalable live video streaming to coop-erative client using time shifting and video patching//Pro-ceedings of the IEEE INFOCOM. Hong Kong, China, 2004; 1501-1511
- [59] Guo M, Ammar M H, Zegura E W. Cooperative patching: A client based P2P architecture for supporting continuous live streaming//Proceedings of the ICCCN. Rosemont, USA, 2004; 481-486
- [60] Zhang X, Shin K G. Statistical analysis of feedback-synchro-nization signaling delay for multicast flow control//Proceed-ings of the IEEE INFOCOM, Anchorage, Alaska, 2001; 1152-1161
- [61] Shi F, Wu J, Xu K. Impact of congestion on the stability of a multicast tree in cumulative layered multicast. IEE Com-munications Proceedings, 2003, 150(5): 371-376
- [62] Long B, Sun L, Chen W, Zhong Y. Improving the stability of spanning trees for application-layer multicast//Proceedings of the ISCC. Alexandria, Egypt, 2004; 1071-1076
- [63] Xu K, Liu J C, Fu L Z, Liu C. On the stability of applica-tion-layer multicast tree//Proceedings of the ISCIS, Istan-bul, Turkey, 2006; 401-412
- [64] Liu Y, Guo Y, Liang C. A survey on peer-to-peer video streaming systems. International Journal of Peer-to-Peer Networking and Application, 2008, 1(1): 18-28
- [65] Sripanidkulchai K, Maggs B, Zhang H. An analysis of live streaming workloads on the internet//Proceedings of the ACM SIGCOMM. Taormina, Sicly, Italy, 2004; 41-54
- [66] Popescu A, Constantinescu D, Erman D, Ilie D. A survey of reliable multicast communication//Proceedings of the NGL. Trondheim, Norway, 2007; 111-118



**SU Jin-Shu**, born in 1962, Ph. D. , professor, Ph. D. supervisor. His current research interests include computer network and information security.

**CAO Ji-Jun**, born in 1979. Ph. D. candidate. His current research interests include high performance router archi-tecture, overlay networks and application layer multicast.

**ZHANG Bo-Feng**, born in 1979, Ph. D. , assistant re-searcher. His current research interests include information security, Internet text categorization.

## Background

Multicast provides an efficient way for Group communi-cations. It is the key technique to support the next-generation Internet applications. The traditional IP multicast provides multicast mechanism in IP layer. However, the deployment of IP multicast is hampered by lots of unsolved technical problems. Nowadays, Application Layer Multicast (ALM) has attracted lots of attention and become the most alterna-tive to IP multicast. The migration of multicast functions from routers to hosts has the potential to address most prob-lems associated with IP multicast, but also faces with new challenges. In ALM tree, when a parent node quits or fails, all its descendent nodes must adjust their positions, which causes the interruption of multicast connections. This prob-lem is called stability problem of ALM trees, and it may af-fect the continuity of multicast data transmission and then de-grade user experience seriously. Therefore, it is necessary to improve the quality of ALM service by enhancing the stability.

How to enhance the stability of ALM is an important re-search top that we are concerned about. This paper analyzes the reasons for the stability problem in depth and proposes

the criterion for evaluating the stability of ALM. The influ-encing factors of the ALM stability, which includes the fre-quency of node leave, the influencing scope of node leave and the time used to recovery ALM tree. According to the influ-encing factors, the technologies to enhance the stability of ALM can be classified as reducing the frequency of node leave, reducing the influencing scope of node leave and short-ening the time used to recovery ALM tree after node leaves. Based on the classification, all the technologies of stability enhancement were overviewed and compared. This paper also concludes by indicating some research directions for ALM stability enhancement. We hope this survey will play a fun-damental role in exploring the technicality to enhance the sta-bility of ALM.

The work is supported by the National Natural Science Foundation of China under grant No. 90604006, the National High Technology Research and Development Program (863 Program) of China under grant No. 2008AA01A325 and the National Grand Fundamental Research Program (973 Pro-gram) of China under grant No. 2009CB320503.