ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.2, February 2009, pp.394−402
doi: 10.3724/SP.J.1001.2009.03176

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

面向大规模实时流媒体的应用层组播方案*

崔 勇+, 徐 恪, 吴建平, 宋林健

(清华大学 计算机科学与技术系,北京 100084)

Application Layer Multicast Mechanism for Large-Scale Real-Time Streaming Media

CUI Yong+, XU Ke, WU Jian-Ping, SONG Lin-Jian

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: E-mail: cy@csnet1.cs.tsinghua.edu.cn

Cui Y, Xu K, Wu JP, Song LJ. Application layer multicast mechanism for large-scale real-time streaming media. *Journal of Software*, 2009,20(2):394−402. http://www.jos.org.cn/1000-9825/3176.htm

Abstract: An application layer multicast mechanism, CD-Media, is proposed based on the combination of central-control and distributed self-organization. This mechanism with different hierarchies is stable and suitable for the quasi-real-time application. Special servers with high performance is deployed in the higher levels to organize the star topology, and they are responsible for the construction and maintenance of Mesh network and multicast tree in the lower levels. In the lower levels a distributed protocol for self-organization is used and Mesh-first strategy is adopted in the Cluster. Multicast technologies both in application layer and network layer are combined to make use of their own advantages to achieve higher performance. An experiment is delicately designed to show the advantage of CD-Media, and it is convinced that it will be a promising mechanism in overly multicast.
Key words: application layer multicast; streaming media; real-time; hierarchy; stability

摘 要: 提出一种基于集中控制与分布式自组织相结合应用层组播方案:CD-Media.这种多层次的、稳定的组播方案在上层由能力较强且稳定的专用服务器组成星形结构,并由它们集中控制下层 Mesh 结构和组播树的构造与维护.下层应用层拓扑采用分布式的自组织协议.该方案分超节点、Cluster 和组播岛 3 个层次,它们共同组成一棵组播树.这种“分层”、“分群”的思路提高了可扩展性,最大化提高网络可支持的用户数量,降低了成本,非常适合网络电视这种单源准实时应用.此外,为解决网络电视对稳定性要求高和应用层网络动态变化大之间的矛盾,Cluster 内的应用层组播成员间采用全连通的 Mesh 结构,在此基础上由超节点集中计算组播树.CD-Media 还将应用层组播与网络层组播相结合,充分发挥两者的优势.实验评价了 CD-Media 的性能,并与已有算法进行比较,结果显示该方案具有明显的优点.
关键词: 应用层组播;流媒体;实时;层次化;稳定
中图法分类号: TP393 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60403035, 90604024 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z205 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant Nos.2009CB320501, 2009CB320503, 2007CB307105 (国家重点基础研究发展计划(973))

Received 2006-04-24; Accepted 2007-07-10

近年来,网络的普及和带宽的提升为通过网络提供视频、音频这种高带宽需求的多媒体应用提供了可能.常用的商业方案是以大量的、稳定的专用服务器为中心,为每位用户提供一条单独的信息传输通道.这种单播方式极大地限制了用户群的规模.要为更多用户提供服务,就不得不增加专用服务器数量.为了降低成本,为更多的用户提供服务,近年来,国际上提出了应用层组播的思想.早期的组播协议工作在网络层,由于网络层组播实现复杂,且商业运作上存在种种问题[1−3],基本上被认为是不可实施的.但是网络层组播在很多方面仍然非常诱人,尤其是组播能够大量地节省服务器成本和网络带宽.随着应用层技术的逐渐成熟(最典型的包括 P2P 网络[4]),不少研究机构和公司希望将网络层组播的思想借用到应用层实现.将组播的功能由路由器转移到终端,在终端间用原来的单播进行传输,这样不必改变原来的网络基础设施,易于部署应用.

对于应用层组播,端系统稳定性和性能不如路由器[5],这就造成了应用层组播在稳定性和效率上不如网络层组播.而应用层组播的不稳定则会导致转发数据的不连续,这种不连续的数据对实时的流媒体应用是非常致命的.本文针对以上问题设计了一种基于集中控制与自组织相结合的应用层组播方案:CD-Media,特别适用于单源准实时应用.所谓单源准实时是指对延迟有一定要求,但不是特别严格,如网络电视或网络广播等.针对这种应用该方案需要达到的目标是:提高单台服务器支持的用户数,最大限度地降低成本;能够适应组成员和网络的动态变化,提高系统稳定性;随着网络条件的变化,能够动态优化组播树;用户加入延迟小,用户收看过程中不会出现明显的停顿.

1 相关工作

应用层组播的协议有很多.从不同角度可以进行不同的分类.

按照数据转发树的计算是集中于一点还是由各个节点分布计算,可以分为集中式和分布式两种.集中式的优点是设计简单,收敛速度快,但形成了单一故障点,可以通过冗余来提高容错性.这种方式的代表是 ALMI[2].ALMI 适用于小规模组播应用,组播由控制节点和成员节点组成.控制节点集中管理成员的加入、离开和组播树的计算.而组播树的计算有赖于各个成员节点将感知的网络拓扑周期高速控制节点.而分布式的代表有 Narada[5],NICE[6],Yoid[1]等.如 Narada,它得到若干个成员节点后,形成到它们的连接,周期性地检测它们以删除无用连接.周期性地探测一些新节点,以增加新连接.形成 Mesh 网络后通过节点之间运行路由协议生成组播树.

按照是否完全对等可以分为代理式和对等式.代理式是指将一些服务器安置在某些策略相关的位置,一般一个域中有 1 个或多个代理节点,基于它们形成组播树,其优点是稳定性好,性能高,但灵活性降低,而且容易成为系统瓶颈.该方案的代表有 Overcast[7],ScatterCast[8].Overcast 设计用来提供带宽敏感的组播应用,并且提高带宽利用率.ScatterCast 则主要针对如何为大规模、异质的用户提供组播服务.

按照是否引入分层和分群来提高可扩展性可以分为层次化和非层次化.层次化的方案有 NICE[6],ZIGZAG[9].大部分组成员位于分层结构的底层,只与少量固定数目的节点存在联系,这样就大大降低了大部分组播成员的处理开销.NICE 和 Zigzag 的主要区别是:在一个 cluster 内部,在 ZigZag 中 Cluster 管理和数据分发是由不同节点完成的,而在 NICE 中这两个功能统一在一个节点上.

按照先建树还是先建 Mesh 网可以分为树优先和 Mesh 优先.树优先是指先构建数据转发树,并为了容错的考虑再与不是父节点的节点保持控制连接,形成控制结构.基于这种方式的有 Yoid[1],Overcast[7],SwitchTree[10]等.Mesh 优先是先形成 Mesh 网并在此基础上形成数据转发树,如 Narada[5].一般树优先的方案延迟要大于 Mesh 优先的方案,并且 Mesh 优先方案在容错性上要强于树优先方案,但 Mesh 结构的可扩展性不高.

本文提出的方案是用于如 IPTV 等单源准实时应用,所以设计思想如下:采用层次化的管理策略,结合多种组播技术达到稳定和准实时的要求,具体分为以下 4 点:(1) 采用集中式与分布式相结合,选取超节点集中负责下层组播树的建造和维护,下层应用层拓扑采用分布式的自组织协议以提高组播树的能力及稳定性;(2) 采用层次和分群的思想以提高用户数量,降低成本;(3) 采用 Mesh 优先方案,先构造 Mesh 网,在此基础上生成数据转发树以满足准实时应用;(4) 将应用层组播和网络层组播结合在一起,尽可能地发挥网络层组播优势.与以往的

工作相比,本文的主要创新点在于集中式和分布式相结合,以及通过混合应用层组播和网络层组播的技术来改进性能的思想.

2 设计方案概述

2.1 网络层次结构

在 CD-Media 中,网络拓扑被划分为 3 个层次:超节点网络、Cluster 网络、组播岛.超节点网络是超节点之间通过点对点的单播方式形成星形网络,距离发布源最近的超节点是这个网络的中心节点.Cluster 网络是 10~20 个组播岛节点(定义见后)自组织形成一个应用层网络,称为 Cluster.这些用户之间构建一棵应用层组播转发树.一个 Cluster 选举出一个代表节点(cluster designated member,简称 CDM),CDM 从超节点获取数据,再转发给 Cluster 内的其他用户.CDM 就是 Cluster 内应用层组播转发树的树根.IP 组播岛(IP multicast island,简称 IMI)是支持 IP 组播的任意大小的网络,它可以是一台主机、一个以太网、一个校园网等等.假设现在的主机都支持 IP 组播(对 Windows 操作系统都是成立的).在一个 IP 组播岛中,主机使用 IP 组播接收/发送数据.在一次 CD-Media 直播过程中,每个 IP 组播岛会挑选一个用户成为组播岛代表节点(island designated member,简称 IDM).

CD-Media 的网络结构图如图 1 所示.从图 1 可以看出,CD-Media 是一种层次化的、基于树的应用层组播协议.它建立一棵以节目源为树根的应用层转发树,转发树的第 1 层是超节点网络,超节点与节目源采用单播连接;转发树的第 2 层是 Cluster 网络,Cluster 由 10~20 个组播岛组成,每个 Cluster 选举出一个代表节点 CDM,CDM 从某台超节点获取数据,转发给 Cluster 内的其他组播岛.组播岛的结构如图 2 所示.不同的组播岛之间通过组播岛代表节点 IDM 之间的 UDP 通道连接.

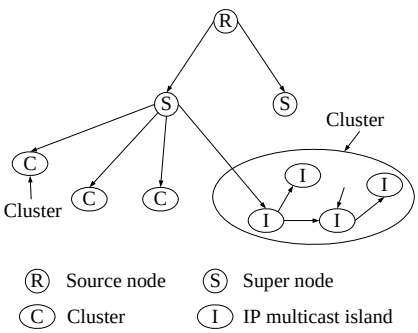

Fig.1 CD-Meida network structure

图 1 CD-Media 网络结构图

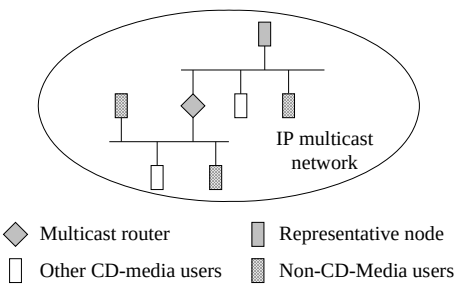

Fig.2 IP multicast island in CD-Media

图 2 CD-Media 中的 IP 组播网络

2.2 CD-Media协议框架

在 CD-Media 的系统框架中,发布源与超节点、超节点与 Cluster 代表节点之间都使用传统的 C/S 模型,因此,在每个 Cluster 内部选举出一个好的代表节点,并构建一棵小型的、高效的应用层组播网络成为成功的关键.每个 Cluster 有一个控制器,集中管理 Cluster 成员的加入、离开和 Cluster 拓扑的维护,并计算以 CDM 为根的应用层组播树.每个 Cluster 成员加入、离开都要通知控制器.每个 Cluster 成员之间全连通地构成 Mesh 结构,周期性地将到达其他节点的链路状态告知控制器.

节点的加入遵循如下步骤:如果该节点所在组播岛已加入 CD-Media,则直接加入该组播岛;否则,该节点作为组播岛代表节点加入 CD-Media.它先向公开节点发请求,得到超节点列表,选择超节点,再选择该超节点下连接的若干个 Cluster,加入 Cluster.加入 Cluster 的过程遵循 Cluster 协议.也就是说,先加入该 Cluster 的 Mesh 网,在通过控制器计算出父节点以后,加入组播树.超时限制内如果没有选出父节点,则直接选择超节点作为父节点,则此时该节点成为 Cluster 代表节点.节点离开的过程是,首先离开组播树,再离开 Mesh 网络.节点的失效是指控制器长期没有收到来自节点的 REFRESH 消息.

3 Cluster协议

3.1 Cluster协议介绍

每个 Cluster 有两个组成部分:1 个 Cluster 控制器和多个 Cluster 组成员.Cluster 控制器是一个程序实体,它运行在所有成员都能访问到的位置.在当前的设计版本中,它运行在 Cluster 所属的超节点上.控制器首先在组成员之间构建一个全连通的应用层网络——Mesh 网络,Mesh 网络中的一条链路代表两个成员之间的一条单播连接,链路的权重为其两端节点之间单播路径的度量(如延迟、可用带宽等),为使设计简单,目前只使用延迟作为度量方式.

Cluster 控制器维护加入 Cluster 成员之间的全局 Mesh 网信息,并负责以 Cluster 代表节点为根,在 Mesh 网络的成员之间构造一棵最短路径树(SPT).使用 Mesh 网能够非常方便地进行最短树的构造,同时提高组播树的可靠性,使组播树性能优化成为可能.另外,这种构建最短组播树的方式能够较好地满足广播应用延时较低的需求.会话数据沿着组播树进行分发,而控制信息则通过控制器和各个成员之间的单播连接进行传输.这种集中式控制与分布式转发融合的方案更有利于系统的稳定性.

3.2 Mesh网络的构造和维护

由于基于源的最短路径树完全在 Mesh 网的基础上构造,所以需要构造一个高效的 Mesh 网.CD-Media 与 Narada 的 Mesh 网设计不同,它为每个 Cluster 维护一个全连通的 Mesh 网来实现组播功能.CD-Media 中的 Cluster 的节点数很少,一般在 10~20 个,这使得构造一棵全连通的 Mesh 网络成为可能,在每个成员中都保存组中其他所有成员的信息也不会带来太大的开销.另外,CD-Media 在超节点存储当前 Mesh 中的所有成员的标识(如 IP 地址).超节点的 IP 地址是众所周知的,用于新加入组播组的成员得到目前组播组成员的信息.用户可以通过离线方式得到超节点的信息(例如通过网页链接).CD-Media 对每个 Cluster 使用集中式控制策略来维护组播树的连续性和效率.这是出于多方面的考虑:提高系统可靠性和降低复杂性(由组成员变化和节点失效恢复引起).另一方面,超节点仅仅在控制平面上进行操作,不会影响到节点之间的高速率数据传输.

3.2.1 Cluster 控制器及 Cluster 成员的操作

Cluster 控制器所在超节点的 IP 地址是众所周知的,它负责整个 Mesh 网拓扑信息的收集、维护,同时为每个 Cluster 节点维护一棵组播树.它在 Mesh 网中的主要作用有以下几点:

1) 处理用户的加入 Mesh 网请求.新加入 Mesh 网的第 n 个节点 I_n 向控制器发送 *JOIN_MESH* 消息,控制器通过 *JOIN_MESH_ACK* 消息向 I_n 返回 Mesh 网中已存在的节点的 IP 地址列表(I_1,…,I_{n-1}).获得该 IP 地址列表后,新加入的节点 I_n 测量到其他所有节点 I_1,…,I_{n-1} 的单播延迟,并将测量结果封装在 *REFRESH* 消息中发送给控制器.控制器收到 *REFRESH* 消息后,更新本地维护的 Mesh 网拓扑信息,向 Mesh 网中加入节点 I_n 及 I_n 到其他节点的链路信息.
2) 周期性更新 Mesh 网的信息.Mesh 网中的每个成员节点 I_k 周期性地测量到其他节点 I_i 的延迟,即在 Mesh 网中两点之间链路 L_{ki} 的权重,并将测量结果封装在 *REFRESH* 消息中发送给 Cluster 控制器.控制器在收到 *REFRESH* 消息后,更新本地维护的 Mesh 网拓扑信息.
3) 处理用户的离开 Mesh 网请求.当某个成员节点 I_k 主动离开 Mesh 网时,向控制器发送 *LEAVE_MESH* 消息.控制器收到 I_k 发送的 *LEAVE_MESH* 消息后,更新本地所维护的 Mesh 网拓扑信息,删除 A 对应的节点以及 A 所连接的链路,并向 Mesh 网除 I_k 以外的所有成员节点发送更新后的用户列表信息.同时向 A 发送应答消息 *LEAVE_MESH_ACK*.
4) 处理用户的失效.如果控制器在一段时间 T_m 内没有收到某个节点 I_k 发送的 *REFRESH* 消息,则认为 I_k 可能失效.于是控制器向 I_k 连续发送一定数量的探测(probe)消息.如果这些 *PROBE* 消息均没有回复,则控制器断定 I_k 失效.此时,控制器将维护的 I_k 的信息标志为失效,但不能删除此成员节点的信息,以防在收到关于此成员的过时的 *REFRESH* 信息后,误将此成员认为是新加入组播组的成员.控制器更新本地所维护的 Mesh 网拓扑信息,删除 I_k 对应的节点以及 I_k 所连接的链路,并将 I_k 失效的信息通知给 Mesh 网中

与I_k有组播树的节点(I_k在组播树中的父节点和子节点).一段时间后,失效成员的信息I_k就可以被控制器删除了.

从上面的说明中可以看出,在 4 种情况下需要控制器更新自己所维护的 Mesh 网拓扑信息,并向 Mesh 网的每个节点发送通知:控制器收到某个节点的请求加入 Mesh 网的 *JOIN_MESH* 消息;控制器收到某个成员节点发送来的 *REFRESH* 消息;收到某个成员节点请求离开 Mesh 网的 *LEAVE_MESH* 消息;在一段时间内没有收到某成员的 *REFRESH* 消息,控制器认为该成员失效.

其中后两种情况对组播树的建立和数据转发的正确性有直接影响,因此,控制器在这两种情况下需要马上将相关的信息通知给Mesh网中的所有成员节点.对于第2种情况,由于只涉及链路质量的变化,只会影响数据转发的效率.为了降低这种情况下由于向节点发送更新信息而给网络带来的额外开销以及组播树的拓扑切换开销,本文采用延时发送的方式:控制器在收到某个节点发送来的 *REFRESH* 消息后,更新本地维护的 Mesh 网拓扑信息,但在更新过程中统计链路度量的相对变化量之和,如果此次更新过程链路度量的相对变化量之和没有超过更新阈值,则不会向 Mesh 网中的节点发送更新后的信息,直到收到某些节点发送来的 *REFRESH* 消息,使得链路度量的相对变化量之和超过了更新阈值,才会触发更新的拓扑信息的发送.

Cluster 成员执行的任务包括:定位控制器、数据转发、监控其他节点、树形切换.控制器在组播树中相邻的节点之间建立起一种父-子关系,组成员利用这种关系来监控组播树的性能和连接.例如,当子节点检测到父节点的连接失败时,它会切换到其他节点.树形切换的原因有两种:控制器计算出更优化的组播树;组成员发现父节点失败或离开组播树.在这两种情况下,控制器都会为该节点找到一个新的父节点.

3.2.2 Mesh 网的维护

控制器维护整个 Mesh 网的拓扑信息(包括所有成员的信息),当一个新成员加入或者一个已存在的老节点离开时都会触发 Mesh 网的更新.Mesh 网中的每个节点周期性地测量到 Mesh 中其他节点的应用层单播延迟,并将这些信息封装在 *REFRESH* 消息中发送给控制器.控制器在获得由 Mesh 网成员发送来的 Mesh 网拓扑信息后,更新本地所维护的 Mesh 网拓扑信息.Mesh 网中每一个节点 *i* 周期性地产生 *REFRESH* 消息,此消息由单增的序号(sequence number)标识.控制器从成员 *i* 接收到的 *REFRESH* 信息包括以下几项内容:*i* 的 IP 地址;*REFRESH* 消息的序列号 s_i;成员 *i* 到其他节点的路径向量信息列表.

这里,本地到其他成员的路径向量信息通过距离向量算法计算得到.首先,Mesh 网中的每个成员周期性地得到它在 Mesh 网中所连接的各条边的度量(如对于边的延迟的测量,可以通过向邻居节点发送 ping 包来实现);然后,通过与邻居节点交互这些信息并通过距离向量算法计算得到路径向量信息.路径向量信息除了包括本地到其他所有成员的路径的度量之外,还包括这些路径所经过的成员链表.

控制器为每个节点 *i* 维护一个表项,包括以下内容:*i* 的 IP 地址;最近一次从 *i* 收到的 *REFRESH* 消息的序列号 s_i;上一次收到序列号为 s_i 的 *REFRESH* 消息的本地时间;成员 *i* 到其他节点 *k* 的路径向量信息列表.当控制器收到节点 *i* 发来的 *REFRESH* 消息时,遍历本地维护的节点哈希表项,并进行如下处理:如果 *i* 为新加入的成员时,在本地所维护的 Mesh 网信息中加入此成员节点的信息.否则,如果 *REFRESH* 消息中的序列号大于本地为 *i* 维护的序列号 s_i,则更新本地信息.否则,控制器认为收到的 *REFRESH* 消息已经过时,不作任何处理.如果在规定时间内(T_m)控制器没有从节点 *i* 收到新的 *REFRESH* 消息,则控制器认为节点 *i* 失效.

3.2.3 节点加入 Mesh 网

当新节点 *A* 想加入 Mesh 网时,按照如下的步骤完成加入过程:*A* 向控制器发出 *JOIN_MESH* 消息,请求加入控制器维护的 Mesh 网;控制器通过 *JOIN_MESH_ACK* 返回当前 Mesh 网中所有节点的 IP 地址列表,并向其他节点发送组成员更新消息 *UPDATE*.*A* 根据获得的 IP 地址列表,测量到其他节点的单播延迟,并将测量结果封装在 *REFRESH* 消息中发送给控制器.控制器从收到的 *REFRESH* 消息中解析出 *A* 到其他节点的延迟信息,更新本地所维护的 Mesh 网拓扑信息,在 Mesh 网成员列表中加入 *A*,同时还加入 *A* 到其他节点的链路状态信息,更新后的 Mesh 信息以单播的形式发送给 Mesh 网中的其他节点.此时 *A* 已经成功地加入了 Mesh 网.如果 *A* 希望接收数据,则只需加入以 Cluster 代表节点为根的组播树即可.加入组播树的过程详见后文的描述.

3.2.4 节点离开 Mesh 网

如果某个节点 *A* 希望离开 Mesh 网,则它按照以下的步骤完成离开流程:首先,*A* 向控制器发送离开消息 *LEAVE_TREE*,离开组播树的过程详见下节描述.在此阶段,*A* 继续转发数据,减少由于 *A* 离开而造成的子节点的数据分组的丢失.在 *A* 完成离开组播树的操作以后,*A* 不再接收其他节点的数据,也不再向其他节点发送数据.此时,*A* 向控制器发送 *LEAVE_MESH* 消息通知,并启动一个计时器.控制器收到 *A* 发送的 *LEAVE_MESH* 消息以后,更新本地所维护的 Mesh 网拓扑信息,删除 *A* 对应的节点以及 *A* 连接的链路,并将 *A* 离开的消息广播给 Mesh 网的其他节点.同时向 *A* 发送确认消息 *LEAVE_MESH_ACK*.当 *A* 收到 *LEAVE_MESH_ACK* 消息时,表示成功离开 Mesh 网.如果计时器超时以后还没有收到 *LEAVE_MESH_ACK* 消息,则重复向控制器发送 *LEAVE_MESH* 消息,直到收到 *LEAVE_MESH_ACK* 消息为止.

3.2.5 Mesh 网中的节点失效

Mesh 网中的节点 *A* 突然失效(可能由于节点崩溃或者节点附近的网络失效),*A* 无法主动地通知控制器.这样就需要控制器检测节点的失效.如果控制器在一段时间内没有收到节点 *A* 发送的 *REFRESH* 消息,则控制器认为 *A* 可能失效.于是控制器连续向 *A* 发送一定数量的探测消息.如果这些 *Probe* 消息均没有回复,则控制器认为 *A* 失效.控制器随后更新本地所维护的 Mesh 网拓扑信息,从 Mesh 网络中删除 *A* 对应的节点以及与 *A* 相关的链路,并将 *A* 失效的消息通过 *UPDATE_MESH* 消息发送给 Mesh 网的每个节点.

3.3 组播树的构造和维护

控制器负责计算出一棵连接所有 Cluster 成员的最短路径树,并将计算结果传递给所有组成员.链路开销可以由用户自己选择,包括延时、带宽等度量参数.对于视频直播这种实时应用,延时可能是最敏感的度量参数.

如果某个父节点失效,则其下游节点的数据会发生丢失.如果不及时为子节点选择一条合适的新路径,则失效会蔓延到所有的下游子节点.CD-Media 提供了失败检测机制和链路备份机制,使子节点能够有效地检测到父节点的失败,并迅速切换到备份父节点.

CD-Media 在 Mesh 上运行距离向量路由协议.为了避免出现距离向量路由协议中的计数到无穷问题,采用类似于 BGP 协议[RFC 1771]的策略.每个成员不仅维护到其他成员的路由的代价,而且需要维护相应的路径.另外,邻居节点之间进行路由更新时不仅包括到目的地址的路由的代价,而且包括相应的路径.构造数据转发路径时采用了 DVMRP 协议[RFC 1812]中采用的反向最短路径机制.也就是说,只有当 *N* 是从 *M* 到 *S* 的最短路径中的 *M* 的下一跳节点时,*M* 才接收从源 *S* 经过 *N* 转发来的分组.而且,*M* 也只把分组转发给到 *S* 的最短路径的下一跳节点的邻居节点.

3.3.1 节点加入组播树

节点 *A* 加入 Cluster 的组播树之前,首先加入 Mesh 网(参考上节),成为 Mesh 网络的节点.然后按照下面的步骤加入以 Cluster 代表节点 CDM 为根的应用层组播树.节点 *A* 向控制器发送加入组播树的请求消息 *JOIN_TREE*,消息中包括 *S* 的 IP 地址.

由于在控制器上维护当前 Mesh 中的每个成员沿组播树到 *S* 的路径及此路径的度量,于是控制器可以计算出一棵以 CDM 为根的、包括节点 *A* 的度受限的最短路径树,除了考虑最短路径以外,控制器在生成组播转发树时还需要考虑尽量减少对原有组播树的改变;只要从 CDM 到任意一个节点的延时不超过一个阈值(默认为 600ms).控制器计算出的 *A* 在组播树中的父节点和子节点,封装在 *JOIN_TREE_ACK* 消息中发送给 *A*,同时树形拓扑的变化通过 *UPDATE_TREE* 消息通知相关节点.

A 的父亲节点收到 *UPDATE_TREE* 消息后,将 *A* 加入到自己的儿子节点列表中.此后,每当收到 CDM 发送来的数据以后,都要向自己儿子列表中的成员(包括 *A*)转发此数据.*A* 的子节点收到 *UPDATE_TREE* 消息后,将 *A* 加入到自己的父亲节点列表中,同时向 *A* 发送 *JOIN_PARENT* 消息.当 *A* 节点收到 *JOIN_PARENT* 消息时,表示自己加入组播树成功(*A* 也要发送 *join_parent* 给其父节点).

3.3.2 组成员离开

当节点 *A* 主动离开某棵组播树时,执行下面的过程:向控制器发送 *LEAVE_TREE* 消息.控制器收到某个成员

发来的 *LEAVE_TREE* 消息后,执行如下操作:根据当前 Mesh 网计算新的组播树,通知因组播树改变而收到影响的节点;通知 *A* 的父亲节点,从其儿子节点列表中删除 *A* 的标识;根节点向 *A* 发送 *LEAVE_TREE_ACK* 消息.

3.3.3 组成员失效与组播树更新

控制器可以通过 Mesh 网获得某个成员节点的失效信息,并通知给所有节点;另外,收到邻居发送来的通知某个成员失效的信息后,控制器将所维护的该成员的信息标志为失效,但不删除该成员的信息,以防控制器在收到关于此成员的过时的 *REFRESH* 信息后,误将此成员认为是新加入组播组的成员.一段时间后,当失效信息通知到了 Mesh 网中的每个成员时,这个失效成员的信息就可以被删除了.

组播树的根以某种顺序(目前采用随机的顺序)遍历此组播组中的所有节点,从加入组播树的第 2 步开始为每个节点重新选择父亲节点.

4 实验模拟与结果分析

4.1 CD-Media模拟

CD-Media 协议模拟的主要目的是验证 CD-Media 协议是否能够完成设计目标.本文通过模拟器 myns[11]在模拟实验中比较 CD-Media 协议和 Narada 协议.实验使用 1 000 个节点的随机图,节点的平均度数为 5.从图中随机地挑选出一部分节点构建应用层组播树,并对 100 张图的模拟结果取平均值,获得如图 3~图 5 所示的结论.

一个好的应用层组播方案应该能够保证数据路径的质量.应用层组播数据路径的质量通常采用如下两个参数来衡量:强度和伸展度.图 3 研究的是 CD-Media 的传输延迟,其中,*X* 轴表示应用层组播组的成员序号.从图中随机地挑选出 500 个节点构建应用层组播树.实验中,超级节点、CDM(cluster)、组播岛数量、组播岛成员数量按照 1:5:25:250 的比例安排.比较了 CD-Media 和 Narada 两种方案的延时.从图 3 可以看出,CD-Media 在节点的传输延时性能上与 Narada 类似.参考 Narada 的结论,CD-Media 也能达到与网络层组播相似的性能.

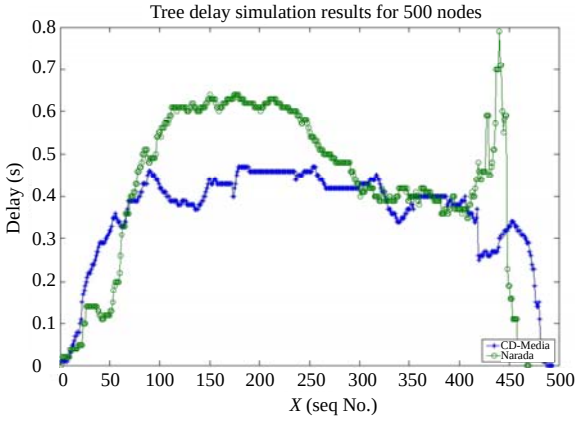

Fig.3 Transmission delay of CD-Media and Narada

图 3 CD-Media 和 Narada 的节点传输延时

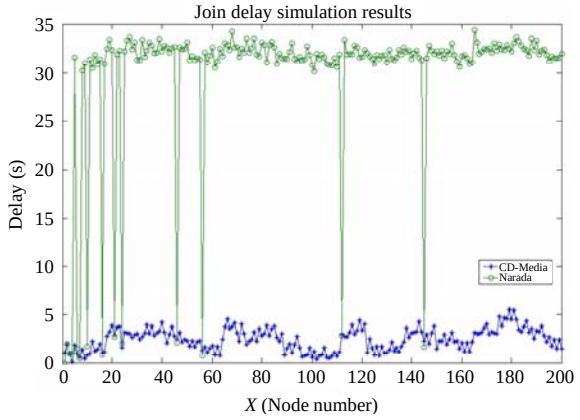

Fig.4 Joining delay of CD-Media and Narada

图 4 CD-Media 和 Narada 的节点加入延时

用户加入和离开 CD-Media 的时间也是评价组播系统的一个重要因素.加入延时定义为从用户加入应用层组播组到获得数据的时间.加入延时是应用层组播设计中的一个重要参数,一种应用层组播在获得广泛的应用之前,其加入延时必须足够低.图 4 表示对 CD-Media 的加入延时的研究,其中,*X* 轴表示应用层组播组的成员序号.从图中随机地挑选出 200 个节点形成应用层组播组,比较了两种方案的加入延时.从图中可以观察到,CD-Media 的加入延时远远低于 Narada.图 5 研究 CD-Media 的离开延时情况,其中,*X* 轴表示应用层组播组的成员序号.从图中随机地挑选出 200 个节点形成应用层组播组,比较了 CD-Media 与 Narada 的离开延时.实验结果表明,CD-Media 的节点离开延时情况与 Narada 类似.

4.2 CD-Media原型实验

在PlanetLab实验网[12]上实现了CD-Media的原型系统.原型系统包括了PlanetLab实验网中几乎所有的活跃节点,主要用来研究CD-Media在真实网络中的稳定性和协议开销.实验环境配置如下:所有的节点在初始100秒内随机加入CD-Media组播组,然后在120分钟之内随机地离开.数据源默认的发送速率为300Kbps.

CD-Media中大部分控制消息是节点间交换的状态信息,这类信息主要发生在Cluster内部,因此Cluster的节点数量可能成为影响协议开销的一个重要因素.图6给出了平均协议开销相对于平均Cluster节点数量的变化函数.基本上,随着Cluster规模的增加,协议开销随之增加,但是相对于数据流量,协议开销是非常低的.

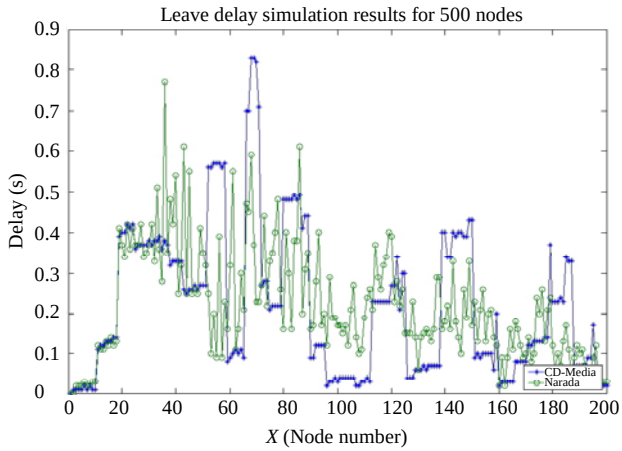

Fig.5 Leaving delay of CD-Media and Narada

图5 CD-Media和Narada的节点离开延时

维持数据稳定性是CD-Media的一个主要目标,为了评价CD-Media的数据稳定性,我们定义一个连续性因子*CI*,*CI*是用户接收到的数据和发送端发送数据的比值.图7给出了节点平均的连续性因子相对于平均Cluster节点数量的变化函数.从图7可以看出,连续性随着Cluster规模的提高而提高,因为每个用户节点有更多的父节点可供选择.

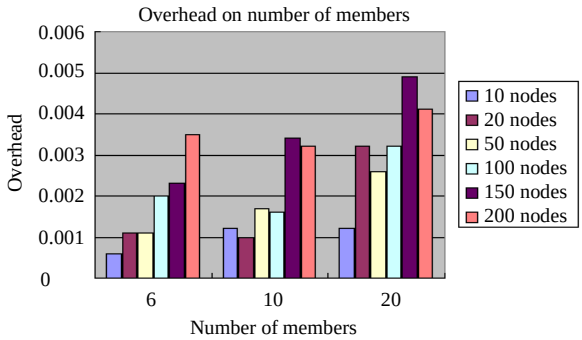

Fig.6 Protocol cost of CD-Media

图6 CD-Media的协议开销

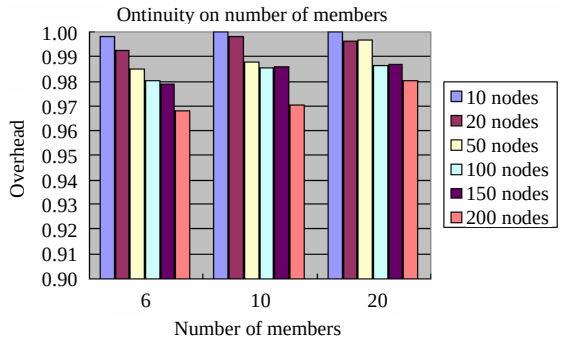

Fig.7 Data continuity of CD-Media

图7 CD-Media的数据连续性

5 结 论

应用层组播中节点的动态变化可能对应用层组播树的稳定性产生很大的影响,而应用层组播的不稳定性则会导致转发数据的不连续,这种不连续的数据对实时的流媒体应用是非常致命的.本文通过专用服务器形成第一层的超节点网络,比单独依靠应用层组播提高了稳定性.另外,系统在Cluster内部采用全连通Mesh结构,进一步提高了稳定性.通过这些途径,降低了稳定性问题对应用层组播的影响.此外,分层和应用层组播的引入以及应用层组播、网络层组播相结合的思想都使可支持的用户数量增加,降低了成本.在CD-Media的部署应用上充分考虑了这种分层和技术融合的思想,解决了分布式流媒体系统中的扩展性问题和服务质量问题.模拟实验表明,CD-Media能够保证用户获得稳定而连续的数据,并获得与Narada相似的性能,而用户加入时间则远远低于Narada用户的加入时间.

References:

[1] Francis P. Yoid: Extending the multicast Internet architecture. White Paper, 1999. http://www.aciri.org/yoid

[2] Pendakaris D, Shi S. ALMI: An application level multicast infrastructure. In: Anderson T, ed. Proc. of the 3rd USENIX Symp. on Internet Technologies and Systems. San Francisco: USENIX Association, 2001. 49−60.

[3] El-Sayed A., Roca V, Mathy L. A survey of proposals for an alternative group communication service. IEEE Network, 2003,17(1): 46−51.

[4] Duan ZH, Zhang ZL, Hou YW. Service overlay networks: SLAs, QoS and bandwidth provisioning. In: Proc. of the 10th IEEE Int'l Conf. on Network Protocols. 2002. 334−343.

[5] Chu YH, Rao SG, Seshan S, Zhang H. A case for end system multicast. ACM SIGMETRICS Performance Evaluation Review, 2000,28(1):1−12.

[6] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast. In: Proc. of the Sigcomm 2002. 2002. 205−217.

[7] Jannotti J, Gifford DK, Johnson KL, Kaasheok MF, Jr O'Toole JW. Overcast: Reliable multicasting with an application layer network. In: Proc. of the 4th USENIX OSDI. 2000. 197−212.

[8] Yatin Chawathe. ScatterCast: An architecture for internet broadcast distribution as an infrastructure service [Ph.D. Thesis]. Berkeley: University of California, 2000.

[9] Tran DA, Hua KA, Do TT. ZIGZAG: An efficient peer-to-peer scheme for media streaming. In: Proc. of the IEEE INFOCOM. San Francisco, 2003.

[10] Helder DA, Jamin S. End-Host multicast communication using switch-trees protocols. In: Proc. of the Workshop on Global and Peer to Peer Computing on Large Scale Distributed System (GP2PC). 2002.

[11] MYNS (p2p) simulator. http://www.cs.umd.edu/~suman/research/myns/

[12] PlanetLab. http://www.planet-lab.org/

崔勇(1976－),男,新疆乌鲁木齐人,博士,副教授,CCF 高级会员,主要研究领域为计算机网络体系结构,服务质量控制,路由算法和性能评价.

徐恪(1974－),男,博士,副教授,CCF 高级会员,主要研究领域为计算机网络体系结构,路由器体系结构,路由算法与协议,组播,服务质量控制.

吴建平(1953－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络体系结构,下一代互联网路由体系结构,形式化方法.

宋林健(1982－),男,博士生,主要研究领域为计算机网络体系结构,网络安全,P2P 性能评价.