

## 支持多点交互同步的应用层组播路由算法\*

姜山<sup>1,2,3+</sup>, 郑庆华<sup>1,2</sup>, 刘均<sup>1,2</sup>, 杜海鹏<sup>1,2</sup>

<sup>1</sup>(西安交通大学 计算机科学与技术系, 陕西 西安 710049)

<sup>2</sup>(智能网络与网络安全教育部重点实验室, 陕西 西安 710049)

<sup>3</sup>(国家计算机网络应急技术处理协调中心, 北京 100029)

### Application-Level Multicast Routing Algorithms for Supporting Multipoint Interaction Synchronization

JIANG Shan<sup>1,2,3+</sup>, ZHENG Qing-Hua<sup>1,2</sup>, LIU Jun<sup>1,2</sup>, DU Hai-Peng<sup>1,2</sup>

<sup>1</sup>(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

<sup>2</sup>(Key Laboratory for Intelligent Networks and Network Security of the Ministry of Education, Xi'an 710049, China)

<sup>3</sup>(National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

+ Corresponding author: E-mail: shaanjiang@gmail.com, http://www.xjtu.edu.cn

**Jiang S, Zheng QH, Liu J, Du HP. Application-Level multicast routing algorithms for supporting multipoint interaction synchronization. Journal of Software, 2011, 22(5): 972-985. http://www.jos.org.cn/1000-9825/3760.htm**

**Abstract:** This paper introduces an idea that solves the synchronization problem in the process of application-level multicast routing without using the dedicated synchronizing controller. Furthermore, the low-delay multicast communication is achieved on base of an idea. The contributions of this paper include: (1) The process of multipoint synchronizing interaction is modeled, and the theorem of application-level multicast routing algorithms supporting multipoint interaction synchronization is proved; (2) An effective low-delay multicast routing algorithm that supports multipoint interaction synchronization is proposed; (3) Performance analysis of this algorithm and the existing similar algorithm is carried out by mathematic methods. Both simulation experiments and practical applications demonstrate that the algorithm is correct and effective.

**Key words:** multipoint video conferencing; low delay; multipoint interaction synchronization; application-level multicast routing; performance analysis

**摘要:** 提出了在多源组播路由过程中解决交互同步问题而无须使用同步控制器的思想,在这种思想的基础上进一步实现低延迟组播。主要贡献包括:(1) 建立面向多点交互过程的同步模型及证明支持多点交互同步的组播路由定理;(2) 提出一种有效的、低延迟的、支持多点交互同步的应用层组播路由算法;(3) 采用数学方法对新算法和现有相关算法进行性能分析和对比。最后,通过仿真实验和实际应用表明,新算法是正确和有效的。

**关键词:** 多点视频会议;低延迟;多点交互同步;应用层组播路由;性能分析

\* 基金项目: 国家自然科学基金(60825202, 60921003, 60633020, 60803079); 国家高技术研究发展计划(863)(2008AA01Z131); 国家科技支撑计划(2006BAJ07B06, 2006BAK11B02); 新世纪优秀人才支持计划(NECT-08-0433)

收稿时间: 2009-05-22; 定稿时间: 2009-10-22

中图法分类号: TP393

文献标识码: A

随着传输带宽和多媒体技术的不断提高和发展,面向 Internet 的视频会议系统有着越来越广泛的应用前景,例如,远程教育、协同工作和网上娱乐等.多点视频会议(multipoint video conferencing,简称 MVC)在同一时间内允许存在多个交互视音频源(简称交互节点),任意参与节点可以接收所有交互节点的视音频数据<sup>[1]</sup>.它是一种最复杂、交互性最强的视频会议形式.本文重点研究多点视频会议的数据分发方法.

应用层组播是一种非常有效的实时数据分发方法<sup>[2]</sup>,提高应用层组播性能的关键是数据流在端节点之间如何选择有效的转发路径——应用层组播路由算法(组播路由).组播路由算法需要满足 MVC 的 3 项应用性能要求:低延迟、低丢包率和多点交互同步<sup>[3,4]</sup>.在 MVC 应用中,多个交互节点的视音频数据存在交互性,即指不同源的多路数据流在呈现时间上有先后顺序.如何使得节点在接收到不同源的多路数据流后,按照合理的时间顺序呈现给用户,是 MVC 的一项技术难题——多点交互同步问题.现有面向 MVC 的组播路由算法忽略了交互同步问题,它们把同步问题留给客户端一个专门的同步控制器来解决.在呈现多路媒体之前,同步控制器需要缓冲至少 2 秒~3 秒的媒体数据用于同步媒体呈现时间<sup>[5]</sup>,从而延缓了交互响应过程,且同步控制器还需在客户端进行复杂的时钟同步操作<sup>[6]</sup>.与现有方法不同,本文研究在数据分发过程中如何有效地选择分发路径,以保证不同源的多路数据流到达每个节点的时间顺序与从数据源发送的时间顺序一致的方法,称为支持多点交互同步的应用层组播路由(简称同步组播路由).同步组播路由的优点是:(1) 无须在客户端使用同步控制器;(2) 数据帧到达节点后可立即播放,无需大量的接收缓冲和额外的同步处理,减少数据帧的播放缓冲时间,加快交互响应过程;(3) 无需节点的时钟同步操作,降低了系统复杂性.因此,与传统方法相比,同步组播路由更适合 MVC 应用的需求.

实现同步组播路由的难点在于:(1) 为了同步多路数据流到达任意节点的时间顺序,路由过程必须考虑多路数据流在端到端延迟上的相互关系.这种相互关系使得任意一路数据流在选择分发路径时会受到其他数据流路由过程的制约,造成数据流不能任意选择上行目标节点.这与现有应用层组播路由算法在路径选择时主要考虑传输延迟和可用带宽两点因素是有很大的区别的.同时,在 MVC 应用中,任意节点都可以成为交互节点.这要求同步组播路由对不同的数据源还要具有很好的普遍性和适应性;(2) 同步组播的路由过程是受到上述约束的,这在一定程度上增加了实现低延迟通信的难度;(3) 在多路数据分发过程中,中间节点需要转发多路数据流,而且节点的可用带宽是有限的,从而容易增大丢包率和造成网络拥塞<sup>[7]</sup>,因此需要进行流量和丢包率控制.

针对以上问题,本文首先建立多点交互过程的同步模型,并证明支持多点交互同步的组播路由定理;其次,本文提出一种有效的、低延迟的同步组播路由算法.该算法基于共享树实现多源组播,从而使得多路数据流的分发路径具有较强的重叠和关联,为实现同步组播路由奠定了基础.相对于现有基于共享树的算法,新算法的特点是在每个交互节点与共享树中最长路径的中点之间建立一条辅助传输链接,从而缩短交互节点到其他节点的传输距离.新算法还进行有效的数据分流,均衡网络流量负载,从而降低丢包率.同时,本文以最小生成树作为共享树,可以利用其低延迟等特性提升数据的分发速度,并且还能有效控制网络丢包率.

本文第 1 节给出支持多点交互同步的组播路由定理的证明和组播路由算法的延迟评价指标.第 2 节引出最小生成树的相关概念.第 3 节采用数学方法对 3 种组播路由算法在多点交互同步和延迟等方面进行深入分析和对比.第 4 节进行实验与分析.第 5 节进行实际性能测试.第 6 节进行相关工作分析与比较.第 7 节总结全文.

## 1 性能分析模型

MVC 允许同时存在多个交互节点,是一种面向 Internet 的实时性高、交互性强的多媒体应用.本文重点分析面向 MVC 的应用层组播路由算法在多点交互同步和组播延迟两方面的性能.

用  $P = \{p_1, p_2, \dots, p_N\}$  表示所有参与节点的集合,  $N$  为节点总数;用  $S = \{s_1, s_2, \dots, s_C\}$  表示所有交互节点的集合,  $C$  为交互节点总数,且  $S \subseteq P$ ;用  $R$  表示组播路由算法,为分发  $s_i$  的数据,  $R$  把所有节点组织成一棵以  $s_i$  为根的有向分发树(简称有向树)  $T_R^{s_i}$ ;用  $T_R = \{T_R^{s_1}, T_R^{s_2}, \dots, T_R^{s_C}\}$  表示  $R$  创建的所有有向树集合;用二元组  $(P, S)$  表示一个多点视

视频会议应用实例.

1.1 多点交互同步

在MVC应用中,同时有多个节点进行交互.通过应用层组播,每个节点(听众或交互节点)接收所有交互节点的视音频数据并在本地回放,从而“看到”和“听到”整个多点交互过程.在真实的交互过程中,存在“先问后答”的客观规律.例如,A和B进行对话,C浏览整个对话过程.A提出一个问题,B回复A的问题.对于C,应先听到A的提问,后听到B的回答.基于此规律,我们定义MVC的多点交互同步如下:

定义1. 在MVC应用中,对于任意两个交互节点A和B,设 $f_A$ 为A产生的任意一个数据帧, $f_{B \rightarrow A}$ 为B在回放 $f_A$ 时产生的第1个数据帧.如果在任意节点都有 $f_A$ 的播放时刻小于 $f_{B \rightarrow A}$ 的播放时刻,即公式(1)成立,那么我们认为,MVC的交互过程符合“先问后答”的客观规律,称为多点交互同步(简称交互同步);否则不同步.

$$PTI(f_A) < PTI(f_{B \rightarrow A}) \tag{1}$$

其中, $PTI(\cdot)$ 表示节点播放某个数据帧的时刻(playback time instant).

本文研究在组播路由过程中实现交互同步.基于交互同步定义,我们提出支持交互同步的组播路由定理.

定理1. 在应用实例 $\langle P, S \rangle$ 中, $s_i$ 和 $s_j$ 为任意两个交互节点.如果组播路由算法R使得沿着 $s_i$ 的有向树从 $s_i$ 到任意节点 $p_k$ 的延迟小于沿着 $s_j$ 的有向树从 $s_j$ 到 $s_i$ 的延迟与沿着 $s_j$ 的有向树从 $s_j$ 到 $p_k$ 的延迟之和,即公式(2)成立,那么交互过程一定是交互同步的,从而称R支持交互同步;否则,R不支持交互同步.

$$Delay(T_R^{s_i}, s_i, p_k) < Delay(T_R^{s_j}, s_j, s_i) + Delay(T_R^{s_j}, s_j, p_k), \forall k \in \{1, 2, \dots, N\} \tag{2}$$

其中, $Delay(T_R^{s_i}, s_i, p)$ 表示数据沿着有向树 $T_R^{s_i}$ 从 $s_i$ 到 $p$ 的延迟.

证明:设 $s_i=A, s_j=B; A$ 生成数据帧 $f_A$ 的时刻为 $t_0$ .这里,我们考虑节点收到数据帧后就立即播放,所以数据帧播放缓冲延迟忽略不计.在 $p_k$ 点,  $PTI(f_A) = t_0 + Delay(T_R^A, A, p_k); PTI(f_{B \rightarrow A}) = t_0 + Delay(T_R^A, A, B) + Delay(T_R^B, B, p_k)$ . 因为公式(2)成立,所以公式(1)也成立.反之亦然.定理得证. □

这里举例说明公式(1)和公式(2)的实际意义.如图1(a)所示,有一个视频会议对话场景:3个交互节点,分别为其创建3棵有向树(如图1(b)~图1(d)所示),数字表示通信延迟.设定每个节点的转发处理时间为1.假定A是提问者,节点B和C是回答者;B和C在接收到A的问题后即作出回答.此时,D听到这3个节点的对答过程.设A提问的时刻为0,D听到A的问题的时刻为 $5+10+10+2 \times 1=27$ ,D听到B的回答的时刻为 $5+10+10+5+5+3 \times 1=38$ ,表明D先听到A的问题,后听到B的回答,这是合逻辑的.D听到C的回答的时刻为 $5+15+1=21$ ,说明D先听到C的回答,后听到A的提问,这是不符合逻辑的.因此,公式(2)可以用来判断R是否支持交互同步.

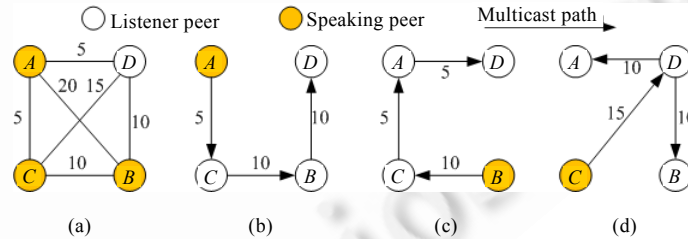


Fig.1 Delivery path for three multicast sources  
图1 3个组播源的分发路径

1.2 组播延迟

MVC可等价看作多组“一对多”通信.决定“一对多”组播延迟的关键度量是最大组播延迟,见公式(3).此外,多个交互节点最大组播延迟的差别也可以很好地反映多点交互质量.因此,对于实例 $\langle P, S \rangle$ ,我们以多组最大组播延迟的平均值 $L(P, S, T_R)$ 和多组最大组播延迟的方差 $D(P, S, L_R)$ 来评估R的组播延迟性能.

$$L(P, S, T_R) = \text{Max}\{l \mid l = Delay(T_R^{s_i}, s_i, p); p \in P\} \tag{3}$$

$$L(P, S, T_R) = \sum_{i=1}^C L(P, S, T_R^{S_i}) / C \quad (4)$$

$$D(P, S, L_R) = \sum_{i=1}^C (L(P, S, T_R^{S_i}) - L(P, S, T_R))^2 / C \quad (5)$$

## 2 最小生成树

基于共享树的组播路由算法为每个交互节点创建的有向树都是以同一个树状覆盖网络(共享树)为基础的.本文以最小生成树作为共享树,其目的是利用最小生成树低延迟等特性来提高 MVC 组播路由算法的性能.

### 2.1 基本概念

最小生成树(minimum spanning tree,简称 MST)引自文献[8],一个 MVC 会话包含一个会话控制中心和一组成员节点,所有成员节点通过节点间的逻辑链接形成一个树状的覆盖网络(组播树).组播树中的逻辑链接(简称链接)表示两个节点之间的 IP 单播连接,每条链接可以双向传输.媒体数据从数据源出发,沿着组播树的链接分发给所有成员节点.另外,每个节点与控制中心建立一条单播链接,用于控制信息传递.除了沿着组播树分发媒体数据之外,每个节点还定期向一部分其他节点发送探测包,测量其链接的性能参数——往返时间  $RTT$  和丢包率  $\lambda$ ,测量结果定期向控制中心报告.控制中心以  $e=k(\lambda) \times RTT$  作为链接的代价权值,计算出一个总体代价最小的树状覆盖网,称为最小生成树. $k(\lambda)$ 为丢包率约束因子,表示丢包率对延迟代价的影响大小.

### 2.2 扩展概念

为了分析基于 MST 的组播路由算法,我们提出了 3 个 MST 的新概念.同时,基于这些新概念,我们提出新的组播路由算法.这些概念如下:

**MST 的最长路径(maximal path).**在 MST 中,数据从任意一个节点出发,可以沿着若干条 MST 的边到达另一个节点.其中,中间节点只能经过 1 次.这样,任意两个节点之间所经过边的有序组合就形成一条路径(路径具有唯一性).在所有的两点间路径中,存在一条路径使得该路径的延迟大于其他路径的延迟,称其为最长路径.并设定  $p_s$  表示最长路径的起点, $p_e$  表示终点,其距离为  $L_{MST}$ .

**MST 的中心节点(center peer).**中心节点是指最长路径上的中点,用 CP 或  $p_c$  表示(注:很难保证在 MST 找到一个真正的节点就是最长路径中点.为了数学分析的需要,我们在最长路径中点虚拟出一个中心节点.但在实际算法实现过程中,我们选定一个在最长路径上距中点最近的节点为 CP).

**节点层级(peer layer).**在 MST 中,CP 到节点  $p$  的路径所包含 MST 边的个数定义为节点  $p$  的层级,用  $Layer(p)$  表示.我们规定中心节点的层级为 0,也就是  $Layer(p)=0$ .

图 2 为最小生成树示意图,包含 14 个节点,分别表示为  $p_1, p_2, \dots, p_{14}$ .其中,  $p_1$  为中心节点,  $p_9$  和  $p_{10}$  为交互节点,节点间的链接是可双向传输的,所有节点被划分为 4 层.

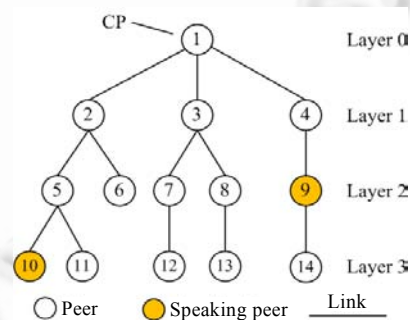


Fig.2 An example of minimum spanning tree

图 2 最小生成树示意

### 3 算法描述及性能分析

本文分析 3 种组播路由算法:邻接式、中心式和并行式.其中,中心式和并行式算法是本文提出的新算法.3 种算法都是以 MST 作为共享树实现多个数据源的并发组播.邻接式算法采用分散式路由,特点为流量分布均衡,延迟较大和支持交互同步;中心式算法采用集中式路由,特点为流量分布集中,延迟小和支持交互同步;并行式算法是邻接式和中心式算法的结合,采用半集中半分散式分发,特点为流量分布均衡,延迟较小和支持交互同步.接下来将逐一介绍这 3 种算法思想,并对它们在交互同步、延迟和流量分布方面进行深入分析和比较.

本文假设在真实的 Internet 中,两个端节点间的正向和反向传输延迟是有一定差别的.本文视正反向延迟相同,并设其值为两个节点往返延迟的一半,从而尽可能地减小分析方法与真实环境的差异,提高分析的真实性.

#### 3.1 邻接式组播路由算法

基于 MST 的邻接式组播路由(adjoining-based multicast routing over MST,简称 AMR)是一种经典的面向 MVC 的组播路由算法<sup>[8]</sup>,其路由方法是媒体数据从每个交互节点开始,沿着 MST 的链接分发给邻居节点;邻居节点复制数据并将其向邻居节点继续分发,分发过程中禁止数据返回分发.递归此过程,直至所有节点接收到数据.图 3(a)显示出一个组播源的 AMR 路由过程.需要强调,在文献[8]中仅提出了 AMR 的思想和实现过程,没有涉及 AMR 在交互同步和延迟方面的分析.本文将用数学方法对 AMR 进行分析,并与我们的算法进行对比.

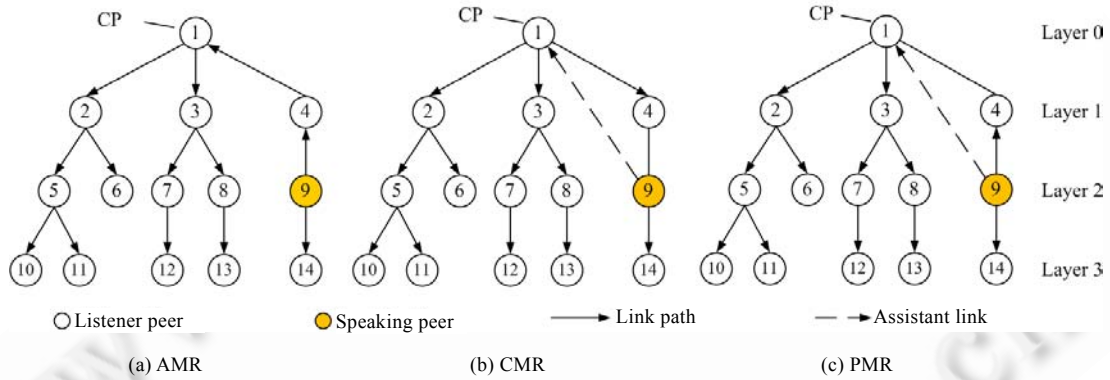


Fig.3 Three multicast routing algorithms based on MST for MVC  
图 3 3 种基于最小生成树的多点视频会议组播路由算法

交互同步.首先提出相关定义和引理,最后证明 AMR 是交互同步的.

**定义 2.** 在 MST 拓扑图中,用  $r(p_i, p_i)$  表示数据从节点  $p_i$  到节点  $p_i$  的路径.对于节点  $k$ ,如果  $r(p_i, k)$  和  $r(p_i, k)$  只相交于  $k$ ,就认为路径  $r(p_i, p_i)$  在  $k$  可分,即  $r(p_i, p_i) = r(p_i, k) + r(k, p_i)$ ;如果  $r(p_i, k)$  和  $r(p_i, k)$  除了  $k$  以外还相交于其他节点,就认为路径  $r(p_i, p_i)$  在  $k$  不可分.

**性质 1.** 在 MST 拓扑图中,如果路径  $r(p_i, p_i)$  在  $k$  点不可分,那么  $r(p_i, k)$  和  $r(k, p_i)$  两条路径所经过边的总和一定完全包含路径  $r(p_i, p_i)$  所经过的边.

**引理 1.** 在 MST 拓扑图中,如果从节点  $s_i$  出发的一条路径与从节点  $s_i$  出发的一条路径第 1 次相交于节点  $k$ ,那么从  $s_i$  到  $s_i$  的路径必定经过  $k$ .

证明:在树形拓扑图中,从  $s_i$  到  $k$  的路径与其逆路径从  $k$  到  $s_i$ ,所经历节点完全一致.且因为从  $s_i$  到  $k$  与从  $s_i$  到  $k$  只相交于  $k$ ,所以从  $s_i$  到  $k$  与从  $k$  到  $s_i$  的也只相交于  $k$ .因此,从  $s_i$  到  $k$ ,再从  $k$  到  $s_i$ ,必定是  $s_i$  到  $s_i$  的路径.又因为两点间的路径是惟一的,所以从  $s_i$  到  $s_i$  的路径必定经过  $k$ .引理得证. □

**引理 2.** 在 MST 树形拓扑图中,如果两个节点  $s_i$  和  $s_i$  分别到节点  $p$  的两条路径第 1 次相交于节点  $k$ ,那么这两条路径在  $k$  之后必定重叠.

证明:反证法.假设两条路径第 1 次相交于  $k$  后不重叠.因为这两条路径最终又相交于  $p$ ,所以根据假设,确定

从  $k$  到  $p$  至少存在两条不同的路径.但是在 MST 树形图中,两个节点之间的路径是惟一的.得出矛盾,假设不成立.引理得证.  $\square$

**定理 2.** AMR 是交互同步的.

证明:用  $R(s,A,B)_{AMR}$  表示数据从节点  $A$  沿着有向树  $T_{AMR}^s$  到节点  $B$  的路径(注: $R(s,A,B)_{AMR}$  与  $r(A,B)$  相同).在 AMR 路由中,对于任意两个源  $s_i$  和  $s_t$ ,设它们到任意一节点  $p$  的两条路径第 1 次相交于  $k$ ,那么  $R(s_i,s_t,p)_{AMR} = R(s_i,s_i,k)_{AMR} + R(s_i,k,p)_{AMR}$ ,  $R(s_t,s_t,p)_{AMR} = R(s_t,s_t,k)_{AMR} + R(s_t,k,p)_{AMR}$ .根据引理 1,  $R(s_i,s_t,p)_{AMR} = R(s_i,s_i,k)_{AMR} + R(s_t,k,p)_{AMR}$ ;根据引理 2,  $R(s_i,k,p)_{AMR}$  与  $R(s_t,k,p)_{AMR}$  是重叠的.所以,  $R(s_i,s_t,p)_{AMR}$  和  $R(s_t,s_t,p)_{AMR}$  两条路径的线段之和完全包含  $R(s_i,s_i,p)_{AMR}$  的线段,满足公式(2),证明完毕.  $\square$

**组播延迟.**根据公式(3)和公式(4)计算 AMR 的组播延迟.首先我们提出相关定理.

**定理 3.** 在 MST 树形拓扑图中,  $p_c$  到  $p_e$  或  $p_s$  的路径是  $p_c$  到其他节点路径中最长的一条.

证明:反证法.假设存在一个节点  $k(k \neq p_e, k \neq p_s)$  使得  $p_c$  到  $k$  的路径是  $p_c$  到其他节点路径中最长的一条,很明显,  $r(p_c,k)$  至少与  $r(p_c,p_s)$  或  $r(p_c,p_e)$  的一条路径在  $p_c$  之外不再相交,我们假定与  $r(p_c,p_s)$  不再相交.根据引理 1,所以  $r(p_s,k) = r(p_s,p_c) + r(p_c,k)$ .而  $r(p_s,p_e)$  也可分为两段  $r(p_s,p_c)$  和  $r(p_c,p_e)$ ,因为  $r(p_c,k)$  的距离大于  $r(p_c,p_e)$ ,所以  $r(p_s,k)$  大于最长路径  $r(p_s,p_e)$ .得出矛盾.当  $r(p_c,k)$  与  $r(p_c,p_e)$  在  $p_c$  之外不相交时,同样得出矛盾.假设不成立.定理得证.  $\square$

**定理 4.** 在 MST 中,从任意一个节点  $p$  出发到其他节点的路径中最长的一条路径必定经过  $p_c$ .

证明:反证法.假设节点  $p$  到其他节点的路径中最长的一条路径不经过  $p_c$ ,设其最长路径为  $r(p,k)$ .无论  $r(p,p_c)$  和  $r(k,p_c)$  是否只相交于  $p_c$ ,根据性质 1,  $r(p,k)$  的边肯定不大于  $r(p,p_c)$  和  $r(p_c,k)$  的边之和.又根据定理 3,  $r(p,k)$  的边不但小于  $r(p,p_c)$  和  $r(p_c,p_e)$  的边之和,同时小于  $r(p,p_c)$  和  $r(p_c,p_s)$  的边之和.很显然,  $r(p,p_c)$  至少与  $r(p_c,p_e)$  和  $r(p_c,p_s)$  两条路径中的一条只相交于  $p_c$ ,假定是  $r(p_c,p_e)$ .所以,  $r(p,p_e) = r(p,p_c) + r(p_c,p_e)$ .因此,  $r(p,k)$  的距离小于  $r(p,p_e)$ ,且  $r(p,p_e)$  经过  $p_c$ .与最初假设矛盾.定理得证.  $\square$

由定理 3 和定理 4 求得 AMR 的组播延迟为

$$L(P,S,T_{AMR}^{s_i}) = Delay(T_{AMR}^{s_i}, s_i, p_c) + L_{MST}/2 \quad (6)$$

$$L(P,S,T_{AMR}) = \sum_{i=1}^C (Delay(T_{AMR}^{s_i}, s_i, p_c) + L_{MST}/2) / C \quad (7)$$

**上行流量.**MVC 可以同时存在多个视频源,如此大量的数据流在所有成员节点之间转发成为导致网络拥塞和大量丢包的风险<sup>[3]</sup>.虽然 MST 在构建时实现了链接的流量控制,但其上层的不同组播路由方法也会对链接流量造成不同程度的影响.因此,本文有必要从流量方面来分析组播路由算法的性能.每个视频源采用树状拓扑分发数据,使得节点的下载流量只与并发视频源个数相关,与路由算法无关.本文中,我们重点考虑节点的上行流量.我们视每个视频源具有相同的输出比特率,因此,节点的上行流量等同于节点的上行目标节点数.我们用公式(8)来表示组播路由算法  $R$  的上行流量:

$$B(P,S,T_R, p) = \sum_{i=1}^C b(T_R^{s_i}, p) \quad (8)$$

其中,  $b(T_R^{s_i}, p)$  表示在组播路由算法  $R$  为组播源  $s_i$  创建的组播树中节点  $p$  的出度.

根据公式(8)求得到 AMR 的上行流量分布为公式(9):

$$B(P,S,T_{AMR}, p) = \sum_{i=1}^C b(T_{AMR}^{s_i}, p) = \begin{cases} Links(p) + (C-1) \times (Links(p)-1), & p \in S \\ C \times (Links(p)-1), & p \notin S \end{cases} \quad (9)$$

其中,  $Links(p)$  表示节点  $p$  在 MST 中相邻边的个数.

### 3.2 中心式组播路由算法

对于 AMR 算法,交互节点越远离中心节点,它到其他叶子节点的距离就越远,使得最大组播延迟就越大.因此,考虑搭建一条新的传输路径,缩短交互节点到其他叶子节点的距离,从而减小最大组播延迟.

中心式组播路由(center-based multicast routing over MST,简称 CMR)是本文提出的一种具有参考价值的基

于 MST 的快速组播路由算法,其路由方法是同时沿着两个方向分发数据(如图 3(b)所示):

- (1) 数据流从每个组播源开始,沿着 MST 的链接发送给下层邻居节点(层级高的节点),并逐层往下分发数据.
- (2) 每个组播源与 CP 建立一条新的单播链接,称其为辅助链接(assistant link).通过辅助链接,数据流从组播源发送给 CP,并由 CP 沿着 MST 的链接逐层往下分发数据;同时,禁止数据流返回组播源.

交互同步.CMR 也是交互同步的,证明如下:

**定理 5.** CMR 是交互同步的路由算法.

证明:用  $A \rightarrow B$  表示节点  $A$  到节点  $B$  的单播,设任意两个源节点  $s_i$  和  $s_t$ .对于任意一个节点  $p$ , $R(s_i, s_i, p)_{CMR}$  分为两段  $s_i \rightarrow p_c$  和  $R(s_i, p_c, p)_{CMR}$ ;  $R(s_i, s_i, s_i)_{CMR}$  也分为两段  $s_i \rightarrow p_c$  和  $R(s_i, p_c, s_i)_{CMR}$ ;  $R(s_i, s_i, p)_{CMR}$  也分为两段  $s_i \rightarrow p_c$  和  $R(s_i, p_c, p)_{CMR}$ .对于 CMR,  $R(s_i, p_c, p)_{CMR} = R(s_i, p_c, p)_{CMR}$ ,所以  $R(s_i, s_i, s_i)_{CMR}$  和  $R(s_i, s_i, p)_{CMR}$  两条路径的线段之和完全包含  $R(s_i, s_i, p)_{CMR}$  的线段.所以公式(2)满足,证明完毕.  $\square$

**组播延迟.**根据公式(3)和公式(4)计算 CMR 的组播延迟.由定理 3 可得:

$$L(P, S, T_{CMR}^s) = \text{delay}(s_i, p_c) + L_{MST}/2 \quad (10)$$

其中,  $\text{delay}(s_i, p_c)$  表示节点  $s_i$  到  $p_c$  的单播延迟.

$$L(P, S, T_{CMR}) = \sum_{i=1}^C (\text{delay}(s_i, p_c) + L_{MST}/2) / C \quad (11)$$

$L(P, S, T_{CMR}^s)$  与  $L(P, S, T_{AMR}^s)$  比较,

$$L(P, S, T_{AMR}^s) - L(P, S, T_{CMR}^s) = \text{Delay}(T_{AMR}^s, s_i, p_c) - \text{delay}(s_i, p_c),$$

其中,  $\text{Delay}(T_{AMR}^s, s_i, p_c)$  是包含  $K$  条单播链接的总延迟( $K \geq 2$ ).  $K$  越大,经过的单播链接越多,导致延迟越大.同时,每个中间节点转发数据需要一定的处理时间,  $K$  越大,转发处理总时间就越大.在 MVC 应用中,需要进行流量控制用于防止网络拥塞.因此, MST 中节点的邻居节点数是受到一定限制的.所以对于多数交互节点,  $K$  会比较大;特别在会话规模较大的情况下,  $K$  就更大了.但是,  $\text{delay}(s_i, p_c)$  仅是一条单播延迟,也不存在转发处理延迟.所以  $\text{delay}(s_i, p_c)$  相对  $\text{Delay}(T_{AMR}^s, s_i, p_c)$  较小.根据以上分析得出,  $L(P, S, T_{AMR})$  大于  $L(P, S, T_{CMR})$ .本文第 4 节的实验结果也验证了该分析的合理性及其结论的正确性.

**上行流量.**当交互节点层级为 0 或 1 时, CMR 和 AMR 路由路径是一样的.本文着重比较算法的区别,且这种情况少,因此,本文作性能分析时不考虑交互节点层级为 0 或 1 的情况.根据公式(8)求得 CMR 的上行流量:

$$B(P, S, T_{CMR}, p) = \sum_i^C b(T_{CMR}^s, p) = \begin{cases} \text{Links}(p) + (C-1) \times (\text{Links}(p) - 1), & p \neq p_c, p \in S \\ C \times (\text{Links}(p) - 1), & p \neq p_c, p \notin S \\ C \times \text{Links}(p), & p = p_c \end{cases} \quad (12)$$

$B(P, S, T_{CMR}, p)$  与  $B(P, S, T_{AMR}, p)$  的比较如公式(13):

$$\begin{cases} B(P, S, T_{CMR}, p) - B(P, S, T_{AMR}, p) = C, & p = p_c \\ B(P, S, T_{CMR}, p) = B(P, S, T_{AMR}, p), & p \neq p_c \end{cases} \quad (13)$$

由公式(13)可以看出, CMR 与 AMR 相比,中心节点的网络流量增大了,其增加的流量与并发的交互节点数成正比.在其他节点, CMR 与 AMR 的流量相等.

### 3.3 并行式组播路由算法

在 CMR 算法中,虽然辅助链接能够有效地降低组播延迟,但同时在中心节点增加了大量的上行网络流量.因此,我们进一步考虑在 CMR 的基础上对中心节点的上行网络流量进行合理的分流.

并行式组播路由(parallel-based multicast routing over MST,简称 PMR)是本文提出的一种有效的基于 MST 的路由算法. PMR 是 AMR 和 CMR 的结合,其路由方法是按照 AMR 和 CMR 两种方式同时进行数据分发(如图 3(c)所示):

- (1) AMR 方式:数据流从每个交互节点开始,向其邻居节点递归分发,但数据不向 CP 转发.

(2) CMR 方式:每个交互节点与 CP 建立一条辅助链接.通过该链接,数据从交互节点分发给 CP,并由 CP 沿着 MST 的链接逐层往下分发给邻居节点,但数据不向交互节点所在子树转发.

**组播延迟.**因为 PMR 是 AMR 和 CMR 算法的结合,所以我们很容易求得 PMR 的组播延迟,

$$L(P, S, T_{PMR}^{s_i}) = \text{Max}(L(P, S, T_{CMR}^{s_i}), L(P, S, T_{AMR}^{s_i})_{Subtree}) \quad (14)$$

其中,  $L(P, S, T_{AMR}^{s_i})_{Subtree}$  表示交互节点  $s_i$  在其所处的一级子树中,采用 AMR 算法的组播延迟(注:一级子树的根节点层级为 1).

$$L(P, S, T_{PMR}) = \sum_{i=1}^C \text{Max}(L(P, S, T_{CMR}^{s_i}), L(P, S, T_{AMR}^{s_i})_{Subtree}) / C \quad (15)$$

$L(P, S, T_{PMR})$  与  $L(P, S, T_{AMR}), L(P, S, T_{CMR})$  比较,因为  $L(P, S, T_{AMR})_{Subtree}$  是 AMR 在一棵 MST 的子树中的组播延迟,所以  $L(P, S, T_{AMR})_{Subtree} < L(P, S, T_{AMR})$ . 根据公式(7)、公式(11)和公式(15),因此满足

$$L(P, S, T_{CMR}) \leq L(P, S, T_{PMR}) < L(P, S, T_{AMR}).$$

**上行流量.**根据公式(8)求得 PMR 的节点上行带宽分配大小为

$$B(P, S, T, p)_{PMR} = \sum_{i=1}^C b(T_R^{s_i}, p) = \begin{cases} 1 + \text{Links}(p) + (C-1) \times (\text{Links}(p) - 1), & p \neq p_c, p \in S \\ C \times (\text{Links}(p) - 1), & p \neq p_c, p \notin S \\ \text{Links}(p) + (C-1) \times (\text{Links}(p) - 1), & p = p_c, p \in S \\ C \times (\text{Links}(p) - 1), & p = p_c, p \notin S \end{cases} \quad (16)$$

$B(P, S, T_{PMR}, p)$  与  $B(P, S, T_{AMR}, p), B(P, S, T_{CMR}, p)$  比较,根据公式(9)、公式(12)和公式(16)求得公式(17):

$$\begin{cases} B(P, S, T_{CMR}, p) - B(P, S, T_{PMR}, p) = C, & p = CP \\ B(P, S, T_{PMR}, p) - B(P, S, T_{AMR}, p) = 1, & p \in S \\ B(P, S, T_{PMR}, p) = B(P, S, T_{AMR}, p), & p \neq CP, p \notin S \end{cases} \quad (17)$$

由公式(17)可以看出,PMR 是把 CMR 在中心节点增加的  $C$  路流量负载平均分配给每个交互节点,从而提高了网络流量的均衡性.AMR 的网络负载分配最合理、最均衡,CMR 容易造成中心节点网络负载的单元瓶颈.PMR 的网络流量分布与 AMR 非常接近,也具有很好的均衡性.在第 4 节的实验中,将通过丢包率的分析来证明这一结论.

**交互同步.**PMR 是把 AMR 和 CMR 相结合.经过证明,AMR 和 CMR 都严格支持多点交互同步.因此,PMR 也能很好地支持交互同步.本文将通过第 4 节的仿真实验和第 5 节的实际应用来验证这一结论.

## 4 仿真实验及结果分析

仿真实验的目的是进一步对比上述 3 种组播路由算法的性能.实验建立在 NS-2 网络仿真平台上<sup>[9]</sup>.使用 GT-ITM 网络拓扑发生器生成 2 000 个节点组成的 Transit-stub 类型的随机网络<sup>[10]</sup>.随机地从 2 000 个节点中选取一部分与覆盖网络节点(可看作用户节点)绑定,计算出它们的最小生成树.实验将从 3 个方面来评价组播路由算法的性能:组播延迟、丢包率和交互同步.

### 4.1 组播延迟实验

组播延迟的评价指标为平均最大组播延迟及方差,分别参见公式(4)和公式(5).最小生成树的初始规模为 10 个节点,并以 5 的步长递增,直到 80 个节点.并在交互节点个数分别为 1,2,4 的情况下测量延迟.实验重复 10 次,每次实验的交互节点是随机选择的,以 10 次测试结果的平均值为最终结果.

图 4 显示出 3 种算法在交互节点数为 1,2 和 4 时的平均最大组播延迟.结果表明,相同环境下,CMR 具有最好的延迟性能,AMR 的延迟最大.CMR 的延迟比 AMR 的延迟减少了近 1/3;PMR 的延迟比 CMR 要稍微差一点,但与 AMR 相比,取得了更好的延迟性能.在图 4 中,3 种算法的延迟曲线呈现为近似阶梯式形状,大致分为 4 个规模区间:[10,20],[20,40],[40,70],[70,80].当 MST 规模为 20,40 和 70 时,3 种算法的延迟都有较大的增加.这是因为,此时 MST 的深度增加了.这一点很好理解,根据公式(7)、公式(11)和公式(15),MST 的深度决定了  $L_{MST}$ ,从而



也影响了 3 种算法的最大组播延迟.从图 4 的 3 个子图可以看出,在交互节点数不同的情况下,同一种算法的延迟近似相等.这是因为不同交互节点数下的测试是基于相同的 MST.

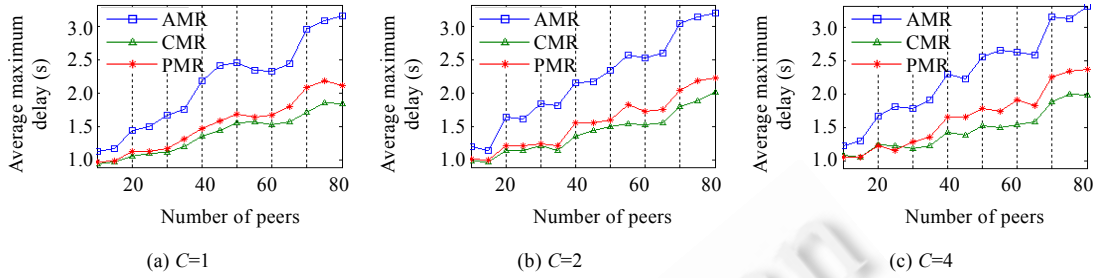


Fig.4 Average maximum multicast delay of three routing algorithms  
图 4 3 种路由算法的平均最大组播延迟

从图 4 中还可以看出,同一种算法的延迟在每个规模区间内并不是随组播规模严格递增的,偶尔会出现略微递减的情形.这是因为  $L(P,S,T_R)$  中的  $Delay(T_R^s,s,p_c)$  或  $delay(s_i,p_c)$  会受到交互节点所处位置的影响,而我们在不同规模下做实验时交互节点是随机选取的.

图 5 分别显示出  $C=2$  和 4 时的最大组播延迟的方差.从图中可以看出:同样是 CMR 的方差最小;AMR 的方差最差;PMR 的方差比 CMR 要稍微差一点,但比 AMR 要好.根据公式(6)、公式(10)和公式(14),决定 AMR 最大组播延迟变化的因素是不同交互节点的  $Delay(T_R^s,s,p_c)$  的变化,对于 CMR 是  $delay(s,p_c)$  的变化.  $Delay(T_R^s,s,p_c)$  包含多条单播路径,因此它受到交互节点位置变化的影响较大; $delay(s,p_c)$  仅为一条单播路径,因此它受到交互节点位置变化的影响较小.PMR 综合了 AMR 和 CMR,因此它的最大组播延迟方差介于 AMR 和 CMR 之间.

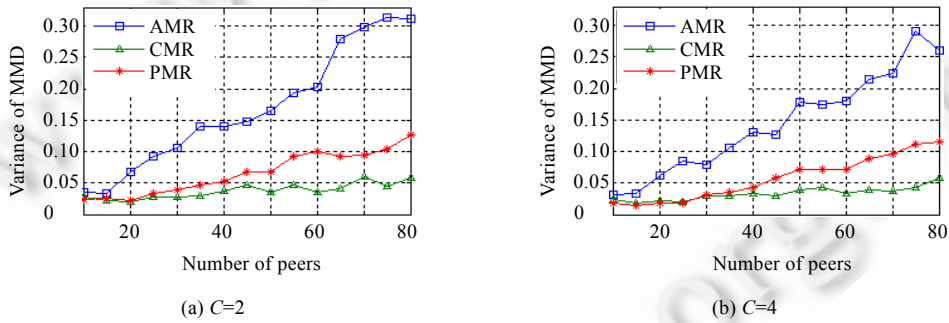


Fig.5 Variation of maximum multicast delay of three routing algorithms  
图 5 3 种路由算法的最大组播延迟方差

4.2 丢包率与上行流量实验

本实验任务是测量和比较 3 种组播路由算法的上行流量和丢包率.丢包率评测指标见公式(18);交互节点的数据输出速率为  $CBR=100kbps$ ,交互节点个数分别为 1,2 和 4 时,根据公式(13)和公式(17),3 种算法的上行流量主要是中心节点和发言节点上有区别,其他节点差别不大.因此,这里主要测量中心节点和发言节点的上行流量和丢包率.同时,也随机测量一个非交互节点(LP),用于进行广泛比较,其实验结果见表 1.

$$PLR(p) = 1 - \frac{\text{节点 } p \text{ 的所有孩子节点接受到 } p \text{ 的数据包个数之和}}{\text{节点 } p \text{ 向所有孩子节点发送数据包个数总和}} \quad (18)$$

表 1 显示出,AMR 算法的总体丢包率是 3 种算法中最低的,这说明 AMR 的上行流量分布是最均衡的.与 AMR 相比,CMR 在 CP 的上行流量增多,从而导致丢包率增加;并且交互节点数目越大,其上行流量的增量就越

大,丢包率也大幅度增加.这说明 CMR 的每条辅助链接使得网络流量集中于 CP,而过多的网络流量造成了丢包率的提升.与 AMR 相比,PMR 在每个交互节点的上行流量增加 100kbps 左右,且其增量与交互节点个数无关.这说明 PMR 把辅助链接在中心节点增加的网络流量平均分流给了交互节点.从表 1 可以看出,AMR 和 PMR 的丢包率是大体接近的.因此,PMR 也实现了很好的流量负载均衡.

**Table 1** Uploading traffic and packet loss rate at key peers of three algorithms

表 1 3 种算法在关键节点的上行流量和丢包率

| C | Peer | Links (p) | Uploading traffic (kbps) |               |              | PLR (%)     |              |             |
|---|------|-----------|--------------------------|---------------|--------------|-------------|--------------|-------------|
|   |      |           | AMR                      | CMR           | PMR          | AMR         | CMR          | PMR         |
| 1 | CP   | 6         | 513.9                    | 615.5         | 516.2        | 0.43        | 1.04         | 0.37        |
|   | SP1  | 6         | 624.7                    | 624.0         | 715.4        | 0.65        | 0.59         | 1.16        |
|   | LP   | 7         | 614.0                    | 607.1         | 607.9        | 0.37        | 0.41         | 0.49        |
| 2 | CP   | 4         | <b>611.3</b>             | <b>817.1</b>  | <b>608.9</b> | <b>0.65</b> | <b>3.51</b>  | <b>0.73</b> |
|   | SP1  | 4         | 721.5                    | 722.6         | 818.7        | 0.77        | 0.82         | 1.48        |
|   | SP2  | 5         | 927.2                    | 927.1         | 1020.6       | 1.37        | 1.24         | 2.06        |
|   | LP   | 5         | 830.8                    | 825.1         | 823.8        | 0.86        | 0.82         | 0.85        |
| 4 | CP   | 3         | <b>817.5</b>             | <b>1233.8</b> | <b>813.5</b> | <b>1.32</b> | <b>17.05</b> | <b>1.28</b> |
|   | SP1  | 2         | 507.1                    | 509.4         | 616.9        | 0.89        | 0.94         | 1.87        |
|   | SP2  | 3         | 919.0                    | 928.4         | 1023.8       | 1.44        | 1.35         | 2.07        |
|   | SP3  | 2         | 516.7                    | 514.6         | 619.4        | 1.01        | 0.96         | 1.91        |
|   | SP4  | 3         | 916.3                    | 916.6         | 1023.8       | 1.27        | 1.40         | 2.26        |
|   | LP   | 3         | 811.6                    | 819.6         | 814.6        | 0.88        | 0.97         | 0.86        |

### 4.3 PMR交互同步实验

本实验的目的是检测 PMR 算法实现交互同步的程度.根据支持交互同步的组播路由定理,我们建立一个会话,会话规模为 80 个节点.随机选择两个交互节点,分别作为提问者和回答者.提问者发送提问数据包,回答者收到提问者数据包后立刻发送其响应数据包.数据分发采用 PMR 算法.在此过程中,其他 78 个节点记录提问数据包和响应数据包的到达时刻.重复整个会话 100 次,每次会话的交互节点是随机的,从而降低实验的偶然性.实验数据统计如图 6 所示,y 轴表示记录个数,x 轴表示一组区间,其中每个区间表示节点收到提问数据包与收到其响应数据包的时间间隔的范围.间隔(单位为秒)为负值表示提问数据包先于响应数据包,正值相反.因此,正值为异步交互记录.从图 6 中可以看出,整个实验记录了  $78 \times 100 = 7800$  条,其中异步记录仅为 8 条,约占总记录的 0.102%.因此从实验结果看出,PMR 算法能够很好地实现交互同步组播路由.

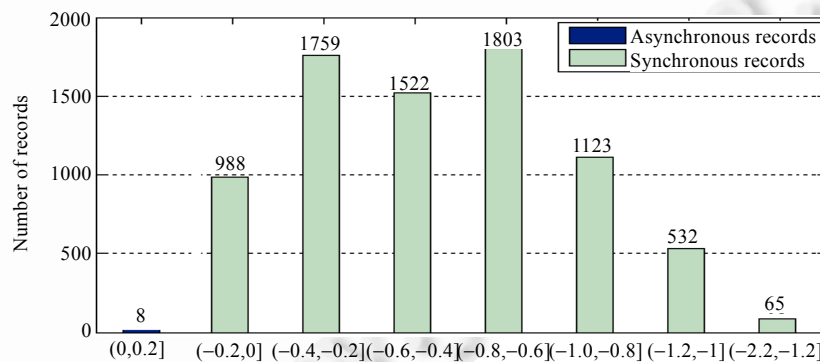


Fig.6 Records of testing multipoint interaction synchronization for PMR algorithm

图 6 PMR 算法的多点交互同步测试记录

## 5 实际应用

本文提出的 PMR 算法已经实现并集成于天地网远程教育系统 Skyclass 3.0 版本.Skyclass 支持视频直播、

视频会议、VOD 和文本聊天室等功能<sup>[11]</sup>,其视频会议采用 PMR 算法,如图 7 所示.Skyclass 可运行在地面 IP 网络之上,系统的组成部分包括教师端、学生端和控制中心,教师端和学生端可分布在广域网中.



Fig.7 User interface of multipoint video conferencing in Skyclass  
图 7 Skyclass 的多点视频会议的用户界面

这里,我们以 PMR 算法在北京和西安等 5 地的一个实际应用测试例子来说明 PMR 在交互同步、延迟和丢包率方面具有良好的实际性能.实例场景:控制中心运行在西安交大网络学院,其主要作用为管理所有客户端节点(教师/学生)和维护视频会议会话;教师端程序运行在人大附中,作为视频会议的管理员节点,在视频会议中可以指定或撤消焦点学生(speaking peer);学生节点总共 75 个(分布见表 2),作为成员节点,可以接收所有焦点学生的视音频流,或可被指定为焦点学生发送其视音频流;Skyclass3.0 客户端的视音频发送/接收模块具有 QoS 参数计算功能.作为发送端,在数据帧包头存放数据包编号(128bit)、发送时间戳;作为接收端,通过统计接收数据帧编号计算丢包率,通过对比接收时间和发送时间计算延迟和判断交互同步性.每个客户端都具有通信日志记录功能.实际测试网络环境:主干网络为中国电信和 CERNET 的混合网络.为保证应用测试效果,部分学校在电信网络与 CERNET 之间建立专用代理通道.终端接入带宽 10Mbps 或 100Mbps;视频采用 MPEG4 编码,输出码率为 128kbps.在实测中,分别以焦点学生规模为 2,4,5 进行测试,且焦点学生都不在同一地点,从而保证测试结果的普遍性和真实性.实际应用性能测试见表 3.所有性能参数均在最小生成树稳定情况下测得.

Table 2 Distribution of all participating peers during a practical application  
表 2 一次实际应用中的所有节点分布

| Distribution    | The High School Affiliated to Renmin University of China | The High School Affiliated to Xi'an Jiaotong University | The High School Affiliated to Xinjiang Agricultural University | The Fiftieth High School of Urumqi | The High School of Mianyang | Yongning High School of Ningxia |
|-----------------|--|---|--|------------------------------------|-----------------------------|---------------------------------|
| Number of peers | 20   | 20  | 20   | 5                                  | 5                           | 5                               |
| Access network  | China Telecommunications                                 | CERNET  | China Telecommunications                                       | China Telecommunications           | China Telecommunications    | China Telecommunications        |

Table 3 Performance of PMR during a practical application  
表 3 PMR 的实际应用性能

| Average maximum multicast delay (s) | Maximum multicast delay (s) | Average loss rate (%) | Asynchronous receive rate (%) | Average maximum multicast delay (s) |
|-------------------------------------|-----------------------------|-----------------------|-------------------------------|-------------------------------------|
| C=2                                 | 1.89                        | 2.02                  | 1.25                          | 0                                   |
| C=4                                 | 2.41                        | 2.72                  | 1.87                          | 0                                   |
| C=5                                 | 2.75                        | 2.97                  | 1.85                          | 0                                   |

## 6 相关工作分析与比较

本文主要涉及针对多点视频会议的传输延迟、同步控制和丢包率等 QoS 问题,其相关工作主要包括以下 5 个方面:

(1) 最小代价应用层组播.文献[12]定义度约束下求解最小延迟生成树的问题模型,并考虑节点的转发处理延迟因素,分别通过基于最大出度结合最短路径的路由算法和考虑节点出度的 H-MDM 改进算法实现两种近似最小延迟组播.Hosseini 等人定义面向多源分发的度约束下求解最小代价生成树的问题模型<sup>[1]</sup>.不同于文献[12],该模型考虑多棵生成树分布在同一组节点之间.其算法首先为每棵生成树分配节点出度.在此基础上,采用一种紧凑树(CT)算法的变种方法构建多棵组播数.该方法的特点是,尽可能晚地路由到会使生成树的可用出度变为零的节点,并在路由选择过程中考虑路径的代价因素.基于上述基本思想,构建了面向多源的度约束的近似最小代价生成树.类似研究中还涉及到其他问题模型包括时延约束的最小生成树<sup>[13]</sup>、最小延迟组播问题<sup>[14]</sup>、度约束的最小直径生成树问题<sup>[15]</sup>和度约束的最小平均延迟有向生成树问题<sup>[16]</sup>.它们的区别在于约束条件不同、目标代价函数不同;共同点是寻求一个运算复杂度适中的、求解近似最小代价生成树这个 NP-Hard 问题的方法.这类研究针对各自的应用目标已取得了较好的效果,但是这些算法在选择数据转发路径时所考虑的因素是节点出度、传输延迟、可用带宽、丢包率和算法复杂度,它们都没有考虑多路不同源的数据流到达任一节点的时间相互顺序.为追求最小组播代价,多路不同源数据流到达同一节点的时间顺序被打乱了.因此,客户端节点在呈现多路媒体数据前,必须使用额外的同步方法在接收缓存中同步这些顺序混乱的数据流.此外,这些应用层组播方法没有给出如何有效计算节点最大出度.因此,它们在实际应用中还有待完善.

(2) 同步控制机制.文献[5]刻画用户感官质量(user perceived quality)在应用级别上的量度标准,并提出一种面向实时多对象多媒体应用的同步策略.其策略以用户感官质量为基础,其特点在于采用一种分布式计时模型和两种基于应用自身特点的配置选项用于优化媒体间同步.相关的同步问题研究还有一部分,大致可以分为两类:预防异步发生或降低同步误差影响<sup>[5,17]</sup>.它们的基本思想是,采用接收缓冲和时钟同步等基本技术来均衡网络延迟的变化,从而提高多路媒体的同步精度.但是,这会使得在接收到媒体数据后,需要缓冲 2 秒~3 秒才能呈现给用户<sup>[5]</sup>.为了达到精确的同步呈现质量,这些方法甚至会牺牲通信延迟.以上这些因素会延缓 MVC 的交互响应过程;同时,其时钟同步过程(网络时钟协议<sup>[6]</sup>)会增加系统的复杂性;而且这些同步控制主要解决视音频之间的同步问题,而没有研究视频会议中多点交互同步问题.此外,它们没有涉及到数据如何分发的问题,因此不能单独应用于 MVC 系统.

(3) 文献[8]是与本文相关的研究,它提出一种实现面向 MVC 的应用层组播基础架构,并以共享树为基础实现多源数据分发.但是,文献[8]没有涉及多点同步问题,其组播路由方法的同步性是本文证明的.与之相比,本文的方法在延迟方面具有更好的性能;在流量分布方面,两者都可以有效地均衡网络流量,降低丢包率.

(4) 基于优先级的组播策略.文献[18]构建一个具有流量限制和邻居节点限制的覆盖网络,其优先级策略是根据每个用户对不同视频源的偏好而定.在覆盖网络上为每个视频源计算最优组播路由路径,使得在覆盖网络上传输的视频流的优先级之和是最大的.其目标是在网络资源受限的情况下,让每个用户尽可能多地看到自己喜欢的视频内容.Wu 等人提出了一种应用层服务区分的通信模型<sup>[19]</sup>.不同于文献[18],该模型的优先级策略是以每个通信会话为基础的,考虑不同会话的优先级区分,并根据会话优先级分配带宽、节点计算能力等资源.诸如此类的方法还是较多的,它们的不同点在于不同的优先级策略,共同点在于按照优先级实现在多源数据分发过程中对资源的优化分配.它们考虑了不同数据源之间的重要程度区分,但忽略了不同源数据流之间的时间相关性.

(5) 更广泛的 QoS 问题.Zhang 等人提出一种应用于基于共享树多源组播的拥塞控制方法<sup>[20]</sup>,其特点为构建 PLR(packet loss rate)矩阵,基于 PLR 矩阵准确找到引发网络拥塞的链路,最终通过调整相应数据源的发包率降低网络拥塞.此外,Zhang 等人还提出了一种面向 MVC 系统的混合组播模式<sup>[21]</sup>,其基本思想把所有节点划分为两个域——组播域与单播域,两个域之间的节点按照一定规则进行组播通信.其主要研究工作为通过对组播域中节点进一步分簇和单播域节点与组播域节点的近距离路径选择,用来降低视频会议应用的通信延迟.文献

[22,23]采用分层编码技术来优化在低带宽接入网络环境下多点视频会议的多媒体呈现质量问题,通过考虑节点的重要性和不同用户对变化的敏感度因素,优化多源应用层组播方案.上述这些方法也没有在组播过程中解决多个视频源之间的时序同步问题.

与现有工作相比,本文针对 MVC 的数据分发方法的研究体现了以下新特性:

(1) 分析在 MVC 应用中多路交互数据流到达每个用户节点的时间顺序对交互过程的影响,建立了 MVC 交互过程的同步模型,证明了支持多点交互同步的组播路由定理.其突出特点是,多路不同源的数据流通过同步组播路由到达客户端后,无需专门的同步模块即可解决多源数据流的同步播放问题,并有效减少数据帧的播放缓冲时间,加快 MVC 交互响应.

(2) 分析基于共享树的组播路由算法的特点,发现其分发路径具有较强的重叠性和关联性,从而为同步多路交互数据流到达每个节点的时间顺序奠定基础,从而能够有效地支持多点交互同步.

(3) 本文还重点考虑应用层组播的延迟性能和流量控制,在保证同步组播路由的基础上,有效地减少组播路由算法的延迟和丢包率,实现了同步路由、低延迟路由和低丢包率路由三者的有效结合.

## 7 总 结

本文主要研究应用层组播路由算法如何同时解决 MVC 应用中的同步、延迟和丢包率问题.首先,我们定义了 MVC 交互过程的同步模型,并证明了支持多点交互同步的组播路由定理.支持多点交互同步的组播路由不但能够有效地解决多源交互式数据流的同步呈现问题,而且可以减少数据帧的播放缓冲时间 2 秒~3 秒,加快交互响应.进一步,本文采用数学方法对 AMR,CMR 和 PMR 算法在交互同步和延迟方面进行分析和对比.通过分析,这 3 种基于共享树的组播路由算法都能很好地支持交互同步.其中,PMR 算法具有综合优势.最后,通过仿真实验和实际应用,进一步证明了 PMR 算法的正确性和有效性.

在今后的研究中,我们将致力于大规模应用层组播的研究.如何有效地构建大规模组播树,不能仅仅依靠优化算法,还需要结合应用特点和需求实施某种策略才能获得较好的实际应用性能.

## References:

- [1] Hosseini M, Georganas ND. End system multicast routing for multi-party videoconferencing applications. *Computer Communications*, 2006,29(11):2046–2065. [doi: 10.1016/j.comcom.2005.12.009]
- [2] Luo JG, Zhang M, Zhao L, Yang SQ. A large-scale live video streaming system based on P2P networks. *Journal of Software*, 2007, 18(2):391–399 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/391.htm> [doi: 10.1360/jos180391]
- [3] Zheng CY, Wang X, Zhao J, Xue XY. P2P video-on-demand content distribution schemes. *Journal of Software*, 2007,18(11): 2942–2954 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/2942.htm> [doi: 10.1360/jos182942]
- [4] Segui FB, Cebollada JCG, Mauri JL. Multimedia group synchronization algorithm based on RTP/RTCP. In: *Proc. of the 8th IEEE Int'l Symp. on Multimedia (ISM 2006)*. IEEE Computer Society, 2006. 754–757. [doi: 10.1109/ISM.2006.110]
- [5] Liu H, Zarki ME. Towards the delay and synchronization control for networked real-time multi-object multimedia applications. In: *Proc. of the 9th IEEE Int'l Workshop on Object-Oriented Real-Time Dependable Systems*. IEEE Press, 2003. 79–86. [doi: 10.1109/WORDS.2003.1267493]
- [6] NTP: The network time protocol. <http://www.ntp.org/>
- [7] Civanlar MR, Ozkasap O, Celebi T. Peer-to-Peer multipoint videoconferencing on the Internet. *Signal Processing: Image Communication*, 2005,20(8):743–754. [doi: 10.1016/j.image.2005.05.003]
- [8] Pendarakis D, Shi S, Verma D, Waldvogel M. ALMI: An application level multicast infrastructure. In: *Proc. of the 3rd USENIX Symp. on Internet Technologies and Systems*. Berkeley: USENIX Association, 2001. 49–60.
- [9] NS, the network simulator. <http://www.isi.edu/nsnam/ns/>
- [10] Zegura EW, Calvert KL. How to model an Internetwork. In: *Proc. of the IEEE INFOCOM'96*. IEEE Press, 1996. 594–602. [doi: 10.1109/INFCOM.1996.493353]
- [11] Skyclass 3.0—Satellite-Terrestrial networks distance education system (in Chinese). <http://www.runzhi.com.cn/product.asp>
- [12] Cao J, Lu SW. A minimum delay spanning tree algorithm for the application-layer multicast. *Journal of Software*, 2005,16(10): 1766–1773 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/1766.htm> [doi: 10.1360/jos161766]

- [13] Tan SW, Waters G, Crawford J. A survey and performance evaluation of scalable tree-based application layer multicast protocol. Technical Report, No.9-03, Kent: University of Kent, 2003. 1–35.
- [14] Broash E, Shavitt Y. Approximation and heuristic algorithms for minimum delay application-layer multicast trees. In: Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2004), Vol.4. IEEE Press, 2004. 2697–2707. [doi: 10.1109/TNET.2007.892840]
- [15] Shi SY, Turner JS. Multicast routing and bandwidth dimensioning in overlay networks. IEEE Journal on Selected Areas in Communications, 2002,20(8):1444–1455. [doi: 10.1109/JSAC.2002.803065]
- [16] Luo C, Wang W, Tang J. A multiparty videoconferencing system over an application-level multicast protocol. IEEE Trans. on Multimedia, 2007,9(8):1621–1632. [doi: 10.1109/TMM.2007.907467]
- [17] Liu CD, Lee MJ, Xie Y, Saadawi TN. A multipoint multimedia teleconference system with adaptive synchronization. In: Proc. of the 15th Annual Military Communications Conf., Vol.2. IEEE Press, 1996. 543–547.
- [18] Nakamura Y, Yamaguchi H, Hiromori A. On designing end-user multicast for multiple video sources. In: Proc. of the 4th Int'l Conf. on Multimedia and Expo, Vol.3. Washington: IEEE Computer Society, 2003. 497–500. [doi: 10.1109/ICME.2003.1221357]
- [19] Wu C, Li BC. Diverse: Application-Layer service differentiation in peer-to-peer communications. IEEE Journal on Selected Areas In Communications, 2007,25(1):222–234. [doi: 10.1109/JSAC.2007.070122]
- [20] Zhang X, Li CR, Li X, Bao CX. Multicast congestion control on many-to-many videoconferencing. In: Proc. of the 2008 2nd Int'l Conf. on Future Generation Communication and Networking, Vol.1. Washington: IEEE Computer Society, 2008. 260–263. [doi: 10.1109/FGCN.2008.136]
- [21] Zhang X, Li CR, Li X. Hybrid multicast scheme for many-to-many videoconferencing. In: Proc. of the 2008 4th Int'l Conf. on Networked Computing and Advanced Information Management, Vol.2. Washington: IEEE Computer Society, 2008. 482–487. [doi: 10.1109/NCM.2008.96]
- [22] Akkas IE, Civanlar MR, Ozkasap O. Peer-to-Peer multipoint video conferencing using layered video. In: Proc. of the 2006 IEEE Int'l Conf. on Image Processing. IEEE Press, 2006. 3053–3056. [doi: 10.1109/SIU.2006.1659865]
- [23] Akkas IE, Ozkasap O, Reha CM. Multi-Objective optimization for peer-to-peer multipoint video conferencing using layered video. In: Proc. of the 16th Int'l Packet Video Workshop. IEEE Press, 2007. 182–190.

#### 附中文参考文献:

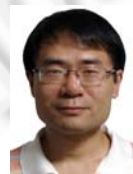
- [2] 罗建光,张萌,赵黎,杨士强.基于 P2P 网络的大规模视频直播系统.软件学报,2007,18(2):391–399. <http://www.jos.org.cn/1000-9825/18/391.htm> [doi: 10.1360/jos180391]
- [3] 郑常熠,王新,赵进,薛向阳.P2P 视频点播内容分发策略.软件学报,2007,18(11):2942–2954. <http://www.jos.org.cn/1000-9825/18/2942.htm> [doi: 10.1360/jos182942]
- [11] Skyclass 3.0——天地网远程教育系统. <http://www.runzhi.com.cn/product.asp>
- [12] 曹佳,鲁士文.应用层组播的最小延迟生成树算法.软件学报,2005,16(10):1767–1773. <http://www.jos.org.cn/1000-9825/16/1766.htm> [doi: 10.1360/jos161766]



姜山(1979—),男,四川邻水人,博士,主要研究领域为多媒体网络,对等网络.



郑庆华(1969—),男,博士,教授,博士生导师,主要研究领域为智能网络教育,网络安全.



刘均(1973—),男,博士,教授,主要研究领域为数据挖掘,计算机支持的协同工作.



杜海鹏(1984—),男,博士生,助理工程师,主要研究领域为多媒体网络,对等网络.