

实时流覆盖组播服务网络研究^{*}

赵焯辉^{1,2} 安玉艳² 王洪君³ 高远¹

(东北大学秦皇岛分校 秦皇岛 066004)¹

(秦皇岛外国语职业学院 秦皇岛 066004)² (吉林师范大学 四平 136000)³

摘要 针对 ISP 在提供实时多媒体服务时对灵活性的需求,设计了一个基于覆盖组播的实时流网络模型,利用面向服务的方法传送语音、视频等流数据,改善了实时流应用中的 QoS,较好地解决了 ISP 部署多媒体增值服务业务的灵活性问题。并结合 NUQ 应用实例,从服务控制和资源管理的角度,研究组合实时流服务和构件来实现组播通信服务,并分析了满足 QoS 和控制资源的方法。

关键词 实时流,覆盖组播,服务组合,面向服务的体系结构,服务控制

Live Streaming Overlay Multicast Service Network

ZHAO Yu-Hui^{1,2} AN Yu-Yan² WANG Hong-Jun³ GAO Yuan¹

(Northeastern University at Qinhuangdao, Qinhuangdao 066004)¹

(Qinhuangdao Foreign Language Professional College, Qinhuangdao 066311)² (Jilin Normal University, Siping 136000)³

Abstract To improve the flexibility in multimedia service for Internet Service Provider(ISP), a live streaming service composition model is addressed. This is an overlay multicast network based on service-oriented methods for transmitting the continuous media data, like audio, video etc. It can achieve the QoS-aware live media applications for ISP's value-added business. Then, an instance called NUQ, is designed and applied according to the model. In NUQ, the composition of the services and the components is discussed in order to improve the service control and resource management, at last the QoS-satisfied and resource-controlled methods is studied.

Keywords Live streaming, Overlay multicast, Service composition, Service-oriented architecture(SOA), Service control

1 引言

随着多用途多媒体通信设备的普及,及用户接入互联网带宽的持续改善,实时流应用呈现不断增长的趋势。为提高灵活性,在多方通信中采用软件体系结构来定义交付和执行服务的网络环境被广泛研究。随着面向服务的体系结构(SOA, Service-Oriented Architecture)在 Web 服务中的成功应用,如何利用 SOA 在实时流通信中提高服务部署的灵活性和可靠性成为热点问题。面对用户对多用途、高性能、安全可靠的服务传输需求,Internet 提供商(ISP)必须重新考虑如何为数据、语音、视频等实时多媒体增值服务业务提供一个健壮且灵活的网络环境。

为解决上述问题,我们定义和部署一个由服务代理(Service Broker, SvB)节点组成的、采用 SOA 软件结构的分层覆盖组播网络(Layered Overlay Multicast Network, LOMN)。LOMN 是一个应用层网络,在 SvB 节点的 IP 层上运行覆盖组播中间件和其他多媒体、服务控制、用户接口等构件,SvB 节点根据作用和位置构成不同的集群,当接收到实时流通信请求时,根据 QoS 需求,多个 SvB 的构件进行组合,这样 SvB 节点间就形成了一个覆盖多播服务树,可以提供用户满意的多方实时流通信服务。LOMN 一方面采用覆盖组播来提高网络应用效率,另一方面采用建立在构件之上的服务组合软件结构实现有服务质量(QoS)保证的语音、视频和实

时数据的通用软件环境,这样已有的集成的、单一应用方式将发展为灵活的多媒体通信服务解决方案。

2 相关工作

随着软件技术的发展,面向服务的软件体系结构 Internet 上得到了应用,服务组合技术可以快速、经济、高效地创建新的网络服务。一些研究工作支持端到端的服务质量服务组合应用,如 Gu^[1] and Xu^[2]采用应用层虚拟链路控制模型来组织服务构件为一个链状系统结构,Jin 等^[3,4]设计了一个分布式的分层服务组合机构,文^[15]给出了嵌入式系统应用中基于 QoS 的服务构件组合方法。

覆盖组播的研究一类是适应特定应用的覆盖网络研究,主要是为广泛使用的网络,如主机组播^[5,6]、内容分发网络^[7]及对等网文件共享^[8]。另一类是适合各种应用的通用覆盖网络研究,可适用于各种应用层的实时流服务。同本文相关的其它一些研究也属于此类。Yoid^[9]是一通用的覆盖体系结构,可支持各种覆盖应用,如,网络新闻、流广播和大量的邮件分布。X-Bone^[10]是一个能自动设置覆盖网络的系统,它工作在 IP 层并基于 IP 隧道技术。OverQoS^[11]是一使用覆盖网络提供 Internet QoS 的体系结构,第三方提供者能使用可控丢失虚拟链路技术为客户提供 QoS 服务,可控丢失虚拟链路技术能确保只要聚合率不超过特定值则聚合丢失率就很小。服务覆盖网络(SON)^[12]使用覆盖技术提供 Internet 增值服务,

^{*} Supported by the National Natural Science Foundation of China under Grant No. 60273078(国家自然科学基金)。赵焯辉 博士生,主要研究领域为面向服务的体系结构、覆盖网络、流媒体;安玉艳 硕士,讲师,主要研究领域为应用层组播;王洪君 博士,副教授,主要研究领域为域间路由、分布式系统;高远 教授,博士生导师,主要研究领域为软件容错及高可信计算机系统。

SON 能从 ISP 那里购买特定 QoS 保证的带宽来建立逻辑上的点对点服务交付覆盖。

本项目在上述研究的基础上,基于自己的覆盖组播算法 BSMON^[13],针对实时流协作应用采用有 QoS 保证的服务组合软件结构和方法定义了分层覆盖组播软件结构。并实施和部署了 NUQ 信天通实时流多方协作平台,意义在于将面向服务的体系结构中服务组合方法应用于覆盖组播通信环境中,结合 ISP 应用实际对上述大多数方法进行了补充,着重于多方覆盖组播通信的服务控制、服务质量保证和实时流处理分发服务方法的研究,通过对服务质量的约束,可以在 SvBs 节点间建立服务质量满意的覆盖连接,平衡覆盖链路和 SvBs 的流量,提高系统的可扩展性和提供服务的灵活性。

3 分层覆盖组播网络及其软件结构

ISP 部署的服务代理 SvBs 节点构成了分层覆盖组播网络 LOMN,相邻的若干个 SvBs 形成 SvBs 簇,若干个 SvBs 簇可进一步构成超级 SvBs 簇,这样就形成一个层次化的服务代理集群,提高了可扩展性,可以支持大规模的实时流通信。同时,一个服务代理的若干个相邻近的终端用户通过 Peer-to-Peer 方式可构成终端用户组,终端用户组通过访问邻近的 SvBs 与 LOMN 相连。ISP 可以根据终端用户的需求状况规划其覆盖网络,并通过服务级别合约(Service Level Agreement, SLA)向用户提供实时流服务。SvB 采用三层的软件结构,服务构件根据应用请求生成服务实例,SvB 间可以形成多个服务组合,为多个应用提供并发服务,通过覆盖组播树为终端用户提供数据交付服务。

3.1 服务代理

SvBs 用来为实时流应用提供组播服务支持。SvB 在 IP 协议之上部署组播中间件和其他多媒体、服务控制、用户接口等构件,形成三个构件层,最高层是抽象服务接口层,由管理器和功能代理构成,是应用软件的接口;中间层是服务控制层,由一些功能实体组成,包括认证、注册、资源发现、时间表、日志和音频视频处理等构件库;最底层是覆盖组播层,由建立在 TCP/IP 协议之上的网络状况探测、组播协议和底层协议的接口模型等构件库构成,底层协议可以采用传输层的 SCTP(Stream Control Transmission Protocol)、TCP(Transmission Control Protocol)、或者 UDP(User Datagram Protocol)协议。SvB 间就是用这些构件和通信模块互相组合来建立覆盖组播网络,来完成具有 QoS 保证的实时流的转发和交付。用户通过接入 SvB 连接到 LOMN,激发 SvB 的相关服务,然后,在不同的 SvB 间形成一个构件连接链路,为用户提供有 QoS 保证的实时通信服务。

一个部署上述软件结构的 SvB 节点处理实时流的组播服务过程如图 1 所示。步骤 1 表示该 SvB 节点收到父节点的服务请求;步骤 2 表示将该请求由输入缓存转发给功能构件处理;步骤 3 表示如果满足功能需求,将请求转发给服务控制构件;步骤 4 通过覆盖组播构件,查看数据传输是否满足用户服务质量等需求;步骤 5、6、7、8 将能够提供的服务状况返回给父节点;达成服务协议后,步骤 9 接收父节点发送的实时流数据到输入缓存;如果输入流与子节点的 QoS 需求一致,则步骤 10 直接从覆盖组播构件处理,步骤 11 数据流到不同子节点的输出缓存;否则,步骤 10' 表示输入流经功能构件和服务控制构件处理,转化为多个子流,每个子流与相应子节点 QoS 规格一致,然后经覆盖组播构件到输出缓存,最后实时流

将转交给各个子节点。

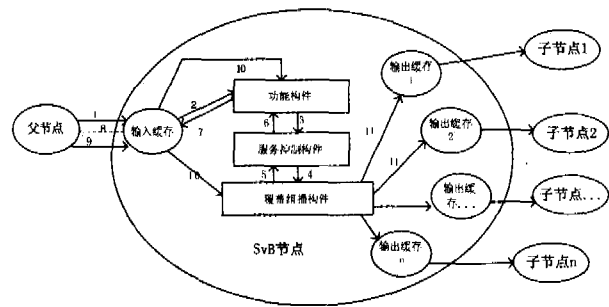


图 1 SvB 节点的组播服务

3.2 LOMN 网络

假如将物理网络的拓扑结构抽象成一个有向无环图 $G=(V, E)$, V 和 E 分别表示网络中的节点集合与物理链路集合,那么覆盖多播网络规划问题就可表示为:已知一个物理网络拓扑结构 G , 给定 QoS 需求和组成员分布的组集合 $\{M_i | i \in N\}$, 在 G 中找到一个虚拟的拓扑结构 G' 使其包含所有的组,并在满足 SLA 的情况下,保持 G' 的代价最小。图 G' 即为 LOMN 网络。

在 LOMN 中,使用 D_x 表示任意自治系统 AS_x , 集合 $\{D_x | x \in N\}$ 表示整个互联网,覆盖节点组成图 $G', G'=(V', E')$, V', E' 分别表示 SvBs 的集合与覆盖边的集合。 $v_i' \in V'$ 表示表示节点 SvB_i, $e_{ij}' \in E'$ 表示 SvB_i 和 SvB_j 间的覆盖链路, $|V'|$ 表示 SvBs 的个数, $|E'|$ 表示 SvB_i 和 SvB_j 间的连接数。

基本链路规则:假如 v_i' 和 $v_j' \in D_x$, 它们之间可以至少存在一个覆盖链路,称为域内链路,表示为 $intraE'$ 。

域间链路规则:假如 $v_i' \in D_x, v_j' \in D_y, x \neq y$, 若存在链路连接 D_x 和 D_y , 则该链路称作域间链路,表示为 $interE'$, 因 D_x 和 D_y 间可以超过一条物理链路,存在多个 $interE'$ 。

节点层次规则:假如 v_i' 有至少一个 $interE'$, 则称之为为核心 SvB, 表示为 cV' ; 假如 v_i' 在 D 的边界并且直接与终端代理相连, 则称之为访问 SvB。

覆盖层次规则:如果 G' 的子集仅由 cV' 节点和 $interE'$ 边组成, 则称它作 G' 的核心组播覆盖子图, 表示为 cG' ; 否则它称作 G' 的访问组播覆盖子图, 表示为 $accG'$ 。

按上述规则定义便可建立层次覆盖组播网络,覆盖网络中连接两个或多个 SvBs 的覆盖链路物理上可能跨越了多个并不拥有任何 SvBs 的 AS, 且同一个域中的 SvBs 相互可形成网状连通, 但当连接多个自治系统时, LOMN 不是全网状拓扑结构, 这能有效地减少网络中的流量并能控制 DHT 表的大小。

当访问 SvB 收到用户加入多方会话的请求时, 首先在其 DHT 中查找此组的组播树, 然后在组播树中为其建立一个或多个与核心 SvBs 的连接, 最后通过核心 SvBs 再与其它域的 SvBs 建立相连, 这是建立组播树的基本过程。

3.3 LOMN 软件结构

在一个基于覆盖网络的实时组通信环境中, 不同的服务代理提供的服务通常是有差异的, 我们采用服务组合技术来满足用户个性化特征, 提供灵活的服务和应用, 适应不同的 QoS、个性化用户界面、异构的设备等等的用户需求。为此, 我们在采用覆盖组播算法^[14] 来提高 ISP 网络的效率和服务的可扩展性的同时, 组合已有服务构件满足多方协作中的复杂应用需求。

实时流服务组合的目标是采用标准的规范和接口为尽可能多的应用提供可交互性服务,增强 ISP 提供服务的灵活性,快速对改变的网络服务需求做出反应,并且可以开发部署、测试新的端系统应用和服务,同时,为 ISP 节约网络资源,减少域间流量,降低费用。

3.3.1 多方协作中实时流服务组合软件结构

多方协作中实时流服务组合软件结构的最顶层抽象服务接口层用于管理用户的服务请求。用户可使用功能代理 (FA) 和 QoS 请求矢量 (VQ^R) 来指定服务请求,FA 指定服务请求的功能 $FA_i = [F_1, \dots, F_n]$ 和服务间相互依赖和替代的关系, VQ^R 指定 QoS 规格和模式。

用户或应用程序向功能代理发送请求之后,消息进入服务控制层,其由已存在的服务构件动态组成的分布式实时流服务所构成。一个实时流服务构件 (Com) 是一提供特定功能的自约束的多媒体应用单元。每个服务构件分别拥有若干个用于接收输入信息和发送输出信息的输入、输出端口。每个输入端口与服务构件间进行异步通信的消息队列相关。服务构件的输入质量请求,如媒体格式、帧速率,表示为 $VQ^i = [q_1^i, \dots, q_n^i]$,服务构件的输出质量规格用 $VQ^o = [q_1^o, \dots, q_n^o]$ 表示。这里引入有向无环图,即服务控制流表示多个服务构件聚合为一个分布式的、有质量保证的服务流。服务流中的节点表示服务构件,构件间的连接被称作服务链路的应用级连接,每条服务链路都经覆盖组播层映射为一条覆盖路径。覆盖组播层用于组织覆盖节点建立有服务质量保证的组播服务树。这样 LOMN 在面向服务的软件结构上就形成了一个服务构件化层次化体系结构。

3.3.2 实时流服务组合控制

多方实时流服务组合的服务质量控制过程主要涉及如下几方面的问题,包括请求服务规格声明、服务发现、服务组合初始化、服务分发等,同时为保证 QoS,在提供服务的过程中,还需要对服务状况进行监控,包括服务质量诊断、服务错误修复等。

如图 2 所示,服务组合分服务建立阶段和服务运行两个阶段进行。服务建立阶段主要步骤如下:1. 用户或应用程序进行服务规格声明,规范的声明所需服务的功能和服务质量、资源需求状况,如服务时间、数据延迟、抖动、图像大小、帧速率和数据丢包率等等,根据 QoS 规格,系统实例化一个功能代理 FA_i ; 2. 服务查找, FA_i 在多个 SvB 节点间进行查找,找到满足要求的服务控制构件和覆盖组播通信构件,在服务控制层形成一个服务控制流图,在覆盖组播层形成一个组播构件图; 3. 服务组合的初始化, LOMN 将根据不同用户或应用程序的服务质量需求和资源状态,对实时流在不同的 SvB 节点位置选择不同的服务控制构件进行初始化处理,对构件的输入输出状况进行服务质量的一致性检验,生成一个服务控制树,通过控制 SvB 节点和链路状况进行组播优化,建立一个多用户的实时流服务组播分发树; 4. 服务分发, LOMN 按照上述组播树分发实时流数据到各个用户或应用程序的访问 SvB 节点,由各个访问 SvB 节点将服务转发给用户。

服务运行阶段主要任务是进行如下两步:1. 服务会话和服务过程监控, LOMN 按照上述步骤建立起了多方实时流通信的服务会话,监控处于活动状态的服务会话的服务质量、资源使用状况、新加入用户等等,对发生的错误进行诊断,通过新的服务构件组合,构建新的服务组播树,满足容错需求; 2. 结束服务会话,组通信结束时,释放所有服务构件和网络资

源,等待下一个服务组合请求。

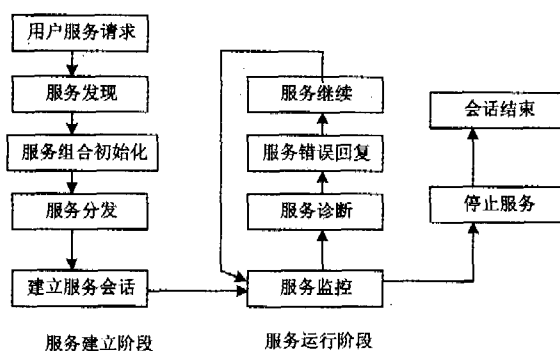


图 2 实时流服务组合控制方法

4 实时流多方协作平台实例

因为视频通信的复杂性,通常用户或应用程序的多方会话实时流服务请求属于复合服务请求,请求包括两部分,服务功能配置要求和服务质量要求。为满足用户服务质量要求,并提高系统的灵活性和可扩展性,我们基于 LOMN 网络及其服务组合软件结构设计和实施了东大信天通实时流多方协作平台 (NUQ),该平台采用了 LMSG^[14] 提供的覆盖组播算法,还需要考虑的问题包括服务初始化、服务控制和实时流服务实施中资源的优化方法。

4.1 服务初始化

在 NUQ 中,服务是一个数据和通信功能的结点,表示为一个多元关系 $(Name, Time, FA, NCom, res)$,它由服务名称 (Name), 服务时间 (Time), 功能代理 (FA)、网络构件 (NCom) 和服务实例占用的系统资源 (res) 组成。功能代理是服务与用户或应用程序交互的接口,指定请求服务的功能 $(F_i | i \in N)$ 和服务间相互依赖和替代的关系,用户可通过扩展标记语言直接请求集成服务。网络构件是一组虚拟的构件集合,是分布在 SvB 节点上的计算单元,通过服务组合来调用网络构件的接口进行计算。服务实例占用的系统资源用来衡量服务 s_i 中占用的系统资源情况,如 CPU 时间、网络带宽等,LOMN 可以根据 SLA 和网络状况决定通过什么样的服务组合为用户提供服务。

用户的服务请求 S^{req} 包括功能请求 F_i 和 QoS 请求矢量 (VQ^R),记作 (F_i, VQ^R) 。服务的初始化从 FA 收到服务请求 S_i^{req} 开始,在 LOMN 网络中映射为功能构件集合 $(F_i | i \in N)$,按照用户服务质量的请求, F_i 启动服务控制和覆盖组播等网络构件,形成分布于多个 SvB 节点的分布式服务组合。

4.2 服务控制

服务控制用来规范服务的 QoS 特征,由服务实例的 QoS 规格及其组成部分描述。QoS 规格给出了构件实例的 QoS 约束值,来检查服务组合过程中服务之间的 QoS 一致性,保证服务质量。下面定义有关服务控制的 QoS 特征。

定义 1 (QoS 一致性) 组合服务的 QoS 一致性 $QoS^{coherence} = (VQ^i, res, VQ^o)$,表示对于一个服务实例及该实例调用的构件,输入的 QoS 请求矢量 VQ^i ,经过服务实例的 res 处理,输出的 QoS 请求矢量 VQ^o ,满足 QoS 约束,并与 VQ^i 规格一致。

定义 2 (QoS 约束) 一个 QoS 约束 $QoS^{val} = (qua, op, Val)$ 给出 QoS 属性 qua 的取值约束, $op \in \{=, \leq\}$ 为约束操作, Val 称为 QoS 属性 qua 的约束值。当对 qua 约束为单值

时, $qua = Val$; 当 qua 约束范围为集合时, $qua \subseteq Val$ 。

定义 3 (QoS 规格) 一个 QoS 规格 $QoS^{Prof} = (Com, MList, TList)$ 给出服务实例需要的构件类型 Com 、服务及其构件 QoS 模式列表 $MList$ 、及 QoS 模式转换列表 $TList$, 其中 $MList = \{ \langle Mod_i \rangle \mid i = 1, 2, \dots, n \}$, 其中 Mod_i 为 QoS 模式, 描述了每个 QoS 属性值的约束定义, n 为 QoS 规格存在的模式数; $TList = \{ Transition_i \mid i = 1, 2, \dots, m \}$, $Transition_i = \langle Mod_i, Mod_j, fun_k \rangle$ 指定当构件实例 QoS 运行模式从 Mod_i 变化到 Mod_j 时, 构件框架调用的回调函数 (callback) fun_k , m 为 QoS 规格存在的转换数。

定义 4 (QoS 模式) 一个 QoS 模式 $QoS^{Mod} = (PList, UList, ResList)$ 表现为服务或者服务调用的构件实例的服务质量模式。

(1) $PList = \{ Provide_i \mid i = 1, 2, \dots, n \}$ 为服务调用提供的 QoS 属性约束定义, $Provide_i$ 给出了构件实例 QoS^{out} 中某一 QoS 属性值的约束定义。

(2) $UList = \{ Use_i \mid i = 1, 2, \dots, m \}$ 为服务实例需求的 QoS 属性约束定义, Use_i 定义了 VQ^m 中 QoS 属性值的约束。

(3) $ResList = \{ res_i \mid i = 1, 2, \dots, h \}$, res_i 描述服务实例 res 中 QoS 属性值的约束。

定义 5 (QoS 满足关系) 构件实例 $ComI_i$ 所需 VQ^m 中的 QoS 属性值能够为相邻构件提供的 QoS 属性值所满足, 则 $\forall q, q \in ComI_j, VQ^m, \exists ComI_j, ComI_i$ 为 $ComI_i$ 的相邻构件, 且 $q \in ComI_j, VQ^{out}$, 如果 $ComI_i$ 和 $ComI_j$ 所选 QoS 模式分别为 Mod_n 和 Mod_m , 在两模式中 q 定义的约束值为 Val_i 和 Val_j , 则若 Val_i 为单值, 则 $Val_i = Val_j$; 若 Val_i 为集合值, 则 $Val_i \cap Val_j \neq \emptyset$ 。

QoS 规格的作用在于描述在 S_i^{req} 请求下生成的组合服务的一致性检测 (VQ^m, res, VQ^{out}) 中每个 QoS 属性值的约束定义, 给出服务构件对每个 QoS 属性的期望。同时为使服务的构件实例具有适应性, 通过定义 QoS 模式使得每种 QoS 属性具有多种可选择约束定义。对于一个 QoS 属性, 通过 QoS 属性约束定义方式, 每个 QoS 模式定义了其不同的取值范围。同一 QoS 规格中, 不同 QoS 模式所定义的 QoS 属性相同, 不同在于其 QoS 约束定义, 即取值范围不同。QoS 模式中 res_i 的 QoS 约束直接或者间接转换为资源占用量, 即构件实例所需资源占用量, 不同于 $Provide_i$ 和 Use_i, res_i 中的 QoS 约束的约束操作只能取 '=', res_i 中的 QoS 属性值代表此模式下构件实例所占用的资源量, 不同服务实例和服务构件实例输出的 F_i 必须遵守 QoS 满足关系。

4.3 实时流服务实施方法

在 NUQ 中, 将构件服务组合问题描述为: (1) 针对复合服务请求的功能集和服务质量请求矢量, 映射到相关构件实例; (2) 根据服务的 QoS 请求 VQ^{req} 和系统资源状态, 选择构件实例的特定 QoS 模式, 使得 Q^{req} 得到满足, 并且在相邻构件实例间使 QoS 一致性关系能够成立, 以及所有构件实例所需资源得到满足。具体实施中, 将实时流处理过程拆分为两个阶段: 聚合阶段, 将处于活动态的多个实时流经由组播树混合为一个实时流; 分发阶段, 将混合的实时流再经一个分发树分发给所有接入 SvB。这样, NUQ 系统可以动态的调整不同 SvB 的负载, 采用持续优化策略, 来自适应的提高实时流服务的质量, 同时可以优化系统资源。

4.3.1 实时流混合

为了动态地构建和调节适应多用户服务会话的服务控制

树, NUQ 的基本想法是在多个选择的 SvBs 间自适应地分发多个实时流混合负载, 并按照用户请求的功能集和 VQ^{req} 分配到不同 FA, 混合负载由动态监控改变的处于激活状态实时流数量决定, 并且动态的调整组播树来适应改变的流量负载。

NUQ 服务会话建立之初, 服务控制树在 SvB 节点中创建一个实时流混合处理器, 被称作根混合器 MP_0 , 系统运行一个选举算法, 促使所有的参与者被初始化为 MP_0 的子节点, 同时, 保证 M_0 是一棵最好的覆盖多播树的源节点, 这个多播树满足从该源节点到其他节点之间的最坏情况的延迟是最低的。组播树的自适应性基本思想在于如果混合处理器超载了就分离一部分负载出去, 如果它的负载量不足就与它的兄弟混合处理器合并, 同时, 混合器也通过在不同的对等节点之间动态的移动来提高服务质量。

4.3.2 混合器分离

组播树中的每一个流混合器 MP_i 监控其输入端口并发到达实时流的数量。由于用户或应用程序可以处于抑制状态, 即此时没有发送数据, 组播树的叶子节点只有在其激活时才产生一个实时流。如果一个组播树的内部节点的输入端口收到了一个输入流, 它将生成一个输出的实时流。假设混合器 MP_i 有 n 个输入端口, 表示为 I_1, I_2, \dots, I_n 。我们采用一个时间轴 A_k 来描述在输入端口 I_k 到达的数据模式, 这里 $1 \leq k \leq n$ 。 A_k 由一系列的有时间戳标记的数字构成, 表示为 $a_k \in A_k$ 。在时刻 t , 如果在输入端口 I_k 有数据到达, 我们设置 $a_k = 1$, 否则, 我们设置 $a_k = 0$ 。我们用 $Total_i$ 表示同时到达混合器 M_i 实时流的总数, 因此, 在时刻 t 同时到达 M_i 实时流的总数可以统计为

$$Total_i(t) = \sum_{k=1}^n a_k$$

服务代理节点的资源是有限的, 为满足 VQ^{req} , 仅能处理有限数量的服务流。设一个 SvB_{*i*} 节点最多能够处理 C_i 个并发流, 被选为混合器 MP_i 的状况。如果到达的服务请求超过了 SvB_{*i*} 能够处理的上限, $Total_i > C_i$, MP_i 将启动分离过程。如果过载的混合器是根混合器, 如图 3a 所示, $MP_i = MP_0$, 此时, SvB_{*i*} 首先生成一个混合器 M_1 , 并且将 MP_0 的所有子流转交给 MP_1 。新的混合器 MP_1 成为 MP_0 的唯一孩子节点, 并且移植给 SvB_{*i*} 的具有最强的流处理能力的邻近节点, 这样组播树的深度值将加一。假设 M_1 被安置在 SvB_{*i*} 上, 如果 M_1 的作业量仍然超过了 SvB_{*i*} 的容量, 因为它已经不是根节点, MP_1 将采用非根节点的分离操作步骤, 继续执行分离操作。所有产生的混合器都将成为根混合器 M_0 的孩子节点。

如果过载的混合器不是根混合器, 如图 3b 所示, 它把自己分为两个混合器 $MP_{i,1}$ 和 $MP_{i,2}$, 其中的一个 $MP_{i,1}$ 仍然运行在 SvB_{*i*}, 同时处理混合器 M_i 的一部分作业量, 这部分负载不超过 C_i 。剩余的负载被指定给一个新的混合器 $MP_{i,2}$ 。SvB_{*i*} 选择一个一个处理能力最强的邻近节点 SvB_{*j*} 来运行 $MP_{i,2}$ 。如果 $MP_{i,2}$ 仍然超过了 SvB_{*j*} 的处理上限, 则将 $MP_{i,2}$ 继续进行分离操作, 直到每一个混合器的作业量都在它在终端的资源负载能力之内。以 MP_i 为父节点的子节点数目的增加, 将触发上述过程。

为最小化所有输入流的平均作业量, 我们将 M_i 的所有子实时流按照数据到达的时间顺序 A_1, \dots, A_i 分发给每一个新产生的子混合器 $MP_{i,1}, \dots, MP_{i,k}$ 。我们计算任何两个数据到达的时间顺序 A_i 和 A_j 的相关系数, 该系数代表并发数据同时到达数据端口 I_i 和 I_j 的概率。然后, 我们配置具有最

小相关性的输入流到一个相同的混合器来最小化每一个混合器的平均聚合作业量。

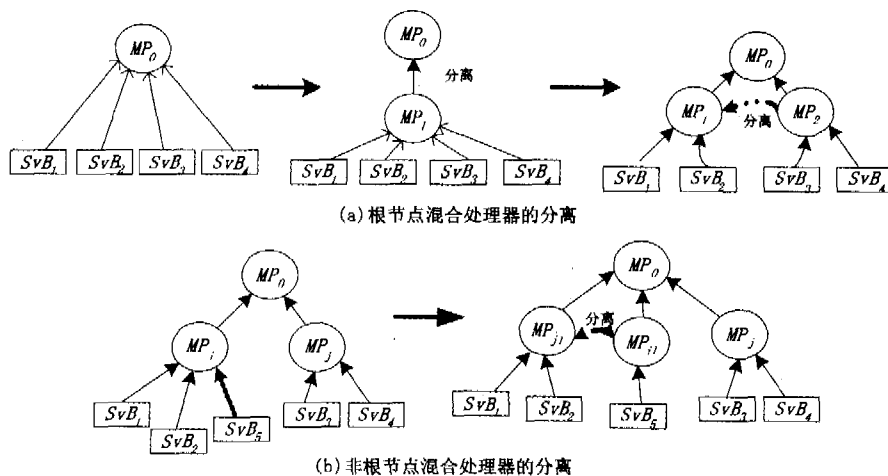


图3 混合处理器的分离操作

4.3.3 混合器聚合

混合器聚合算法可以有效地收缩组播树,从而最小化实时流的转发延迟,来避免额外的音频混合费用,如延迟、丢包率等,同时有效地降低系统代价。与混合器分离的过程相似,每一个混合器 MP_i 监控着并发到达它的输入端口的实时流的数量。如果 $Total_i$ 的所有作业量明显地小于该混合器的处理能力 C_i , 如 $Total_i < |C_i/2|$, 就搜索这棵组播树中随后的姐妹节点。如果 MP_i 和 MP_j 的合并作业量小于一个混合器的处理能力, 如 $Total_i + Total_j < \max(C_i, C_j)$, 就将两个混合器

合并为一个,如果 $C_i \leq C_j$, 算法就删除 MP_i , 并且将 MP_i 的子实时流连接到混合器 MP_j , 否则,算法就删除 MP_j , 并且将 MP_j 处理的子流连接到混合器 MP_i 。注意,上述过程因为可能引起 M_i 和 M_j 的父节点输入流的减少,或许会触发执行它们的聚合操作。如果一个混合器 MP_i 成为其父节点 MP_p 的唯一子节点,可以合并 MP_i 到 MP_p , 来减少组播树的深度。当 MP_p 的所有子节点互相聚合在一起,则将聚合为一个混合器,上述过程如图4所示。

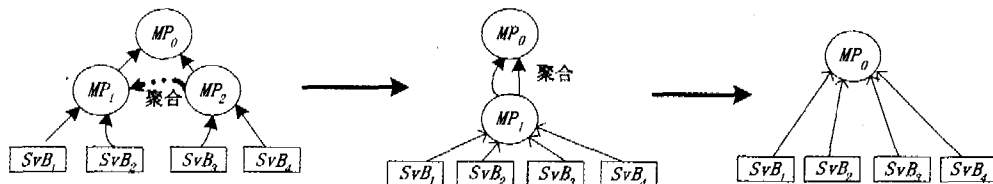


图4 混合处理器节点聚合示意

结论和未来工作 本文采用面向服务软件体系结构为传输语音、视频等流数据建立了一个分层覆盖组播网络环境,可以实现有 QoS 保证的多用户实时协作服务,来满足 ISP 媒体增值服务业务的灵活性需求。并结合 NUQ 软件实例,从服务控制和资源管理的角度,分析和研究了组合服务和构件实现实时流组通信服务的方法,并通过实时流的混合处理控制对构件组合的服务质量约束和提高资源效率的方法进行了探索。

目前,该项目已有一些产品有很好的应用效果,但因为资金和网络环境因素,系统缺乏定量的性能测量。因为面向服务的实时流组播研究近几年刚刚起步,未来工作需要进一步研究面向服务的多方实时流通信机制和方法,研究服务的种类和构件的粒度,在系统仿真的基础上,在真实网络环境中对系统性能进行定量测量。此外,还需要研究 3G 等无线接入用户与 Internet 用户多方协作的实时流调度方法。

参考文献

- 1 Gu X, Nahrstedt K, Chang R, et al. QoS-Assured Service Composition in Managed Service Overlay Networks. In: IEEE ICDCS'2003, May 2003
- 2 Xu D, Nahrstedt K. Finding service paths in an overlay media service proxy network. In: SPIE/ACE MMCN, Jan. 2002
- 3 Jin J, Nahrstedt K. QoS Service Routing in One-to-One and One-to-Many Scenarios in Next-Generation Service-Oriented Net-

- works. In IEEE IPCCC'2004, Apr. 2004
- 4 Jin J, Nahrstedt K. Large-scale service overlay networking with distance-based clustering. In: ACM/IFIP/USENIX International Middleware Conference, Jun. 2003
- 5 Chu Y, Rao S G, Zhang H. A case for end system multicast. In: Proc ACM SIGMETRICS, June 2000. 1~12
- 6 Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users. In: INFOCOM'02, New York, June. 2002
- 7 Krishnamurthy B, Wills C, Zhang Y. On the use and performance of content distribution networks. In: The ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, Nov 2001
- 8 Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. ACM SIGCOMM, Aug. 2001. 149~160
- 9 Francis P. Yoid; Extending the Internet Multicast Architecture [Online]. <http://www.aciri.org/yoid/docs/index.htm>
- 10 XBone [Online]. Available: <http://www.isi.edu/xbone>
- 11 Subramanian L, Stoica I, Balakrishnan H, et al. OverQoS: Offering Internet QoS using overlays. In: The HotNet-I Workshop, Princeton, NJ, Oct. 2002
- 12 Duan Z, Zhang Z, Hou Y T. Bandwidth provisioning for service overlay networks. In: SPIE ITCOM Scalability and Traffic Control in IP Networks (II)'02, Boston, MA, July. 2002
- 13 Zhao Yuhui, An Yuyan, Liu Jiemin, et al. BSMON: Bandwidth-satisfied multicast in overlay network for large-scale live media applications. Lecture Notes in Computer Science, 2005, 3619: 163~172
- 14 Zhao Yuhui, An Yuyan, Wang Cuirong, et al. A QoS-Satisfied Interdomain Overlay Multicast Algorithm for Live Media Service Grid. GCC 2005, Lecture Notes in Computer Science, 2005, 3795: 13~24
- 15 廖渊,唐磊,李明树.一种基于 QoS 的服务构件组合方法.计算机学报,2005,28(4):627~634