

基于 IntServ/RSVP 的 CORBA A/V Stream QoS 实现策略研究

沈 军 董永强

(东南大学计算机科学与工程系 南京 210096)

摘 要: 首先简要介绍基于资源预留的集成服务模型(IntServ/RSVP)、以及 CORBA A/V Stream 规范中的 QoS 相关概念。在此基础上给出一个集成 QoS 模型, 论述 CORBA A/V Stream QoS 与 IntServ/RSVP 之间的映射模型及绑定策略。

关键词: CORBA, QoS, RSVP, IntServ, A/V 流, 流服务

中图分类号: TP393 **文献标识码:** A **文章编号:** 1009-5896(2005)10-1657-05

Research on the Implementation Strategies of IntServ/RSVP-Based CORBA A/V Stream QoS

Shen Jun Dong Yong-qiang

(Dept. of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract Based on the introduction of the QoS conception in CORBA A/V stream specification and the RSVP-based integrated service model, a integrated QoS model is presented. Then the mapping model and the binding strategies between CORBA A/V Stream QoS and IntServ/RSVP are discussed in detail. A prototype system has been developed to verify the feasibility of the model.

Key words CORBA, QoS, RSVP, IntServ, A/V stream, Stream service

1 引言

随着Internet的迅猛发展, 基于Internet的各种新型应用层出不穷, 特别是网络多媒体应用发展前景十分广阔^[1]。然而, 随着高带宽应用的发展, 服务质量(Quality of Service, QoS)正成为发展瓶颈之一, 导致网络及系统的服务质量成为一个研究的热点。另一方面, 随着应用的不断复杂和规模的不断扩大, 应用的快速开发、维护以及集成显得越来越迫切, 特别是异构环境的集成、传统应用的集成问题, 更显得突出。基于分布对象计算的中间件技术的发展, 正好满足了这种需求。因此, 如何将两者结合, 通过中间件技术屏蔽复杂的下层网络的协议和传输控制机制、提供一个高伸缩性、具有集成QoS保证的通用分布多媒体应用开发平台, 具有重要的意义和应用价值

本文结合相关课题的研究和实现, 基于OMG组织的CORBA A/V Stream服务规范^[2], 以及IETF的IntServ/RSVP服务模型(Integrated Service/resource ReSerVation Protocol), 论述集成QoS模型及其描述、协商、映射与确保等实现策略。其目标是建立一个基于CORBA^[3]及QoS使能的高性能分布

多媒体应用开发支撑平台, 以实现分布多媒体应用的快速开发和灵活部署。

本文首先简要介绍网络 QoS 的相关概念、OMG 组织的 CORBA A/V Stream 服务规范及其 QoS 描述, 然后给出一个集成 QoS 模型, 并论述 CORBA A/V(Audio/Video) Stream QoS 与 IntServ/RSVP 之间的映射模型及相关绑定实现策略。

2 IntServ/RSVP 服务模型

从网络通讯和网络服务的角度, 现有的关于提高网络 QoS 的服务模型及相关协议包括: 基于资源预留的集成服务模型(IntServ/RSVP)^[4]; 基于优先权机制的区分服务模型(DiffServ)^[5]; 从流量工程角度提出的多协议标记交换(MPLS)^[6]; 在IEEE802网络上提供分类和优先权功能的子网带宽管理(SBM)^[7]。

RSVP 是集成服务模型建议使用的的一个资源预留建立协议, 它通过 Flowspec 描述应用所需的 QoS, 并用于配置路由器的排队和调度能力。Flowspec 被资源预留建立协议携带, 经过接纳控制的检查, 最终由分组调度机制实现。RSVP 可以被主机用来为特定的应用数据流请求相应的服务质量, 也可以被中继结点(路由器)用以向数据流经过的所有结点发送

服务质量请求。本文所述分别在端系统利用 RSVP 的 API(Application Programming Interface)接口,在中继结点利用 RSVP 与分组调度机制相结合,实现端到端的资源预留和 QoS 确保。RSVP 运行于 IPv4/IPv6 之上,但其不是数据传输协议,而是 Internet 控制协议(信令协议)。

RSVP 的特点是:预留请求是由接收方发起,由接收方根据自身的资源情况和接收愿望指定不同的资源预留,原则上支持系统的异构性;既支持单播的资源预留,也支持多点间的群组通信资源预留,并且它的过滤机制允许预留的资源可以被多个发送者共享或对同一个发送者的预留进行合并;资源预留的建立是在转发数据之前完成,其资源预留是单向的;用软状态指示预留的存在状态,并周期性地地进行刷新。从而支持动态的成员和路由变化;支持多种预留模式,以适应各种应用环境,同时能够保持对不支持 RSVP 的路由器透明。

RSVP 有两类主要的信令消息: PATH 和 RESV-PATH 类消息用于建立路径,它从源端出发,经过路径上的每一个路由器到达会话的接收端。其间,路由器根据本地的 QoS 能力修改通过的 PATH 消息,而接收端根据 PATH 消息中携带的信息来确定网络能够提供的传送能力。可见,PATH 消息存放沿途各结点的“路径传送能力状态”,用以辅助反方向的 RESV 消息逐跳确定路由。RESV 类消息用于预留资源,它从接收端出发,逐跳发往源端,以应答 PATH 消息,标识需要预留资源的会话(与源端)。返回途中,路由器将 RESV 消息(上行消息)与先前经过的相应的 PATH 消息(下行消息)进行对照,以决定每一跳是否能成功地将 RESV 消息中标识的资源数目分配给这些会话。RESV 消息与一个特殊的数据流相关联,并被封装在 IP 或 UDP User Datagram Protocol 数据报中。预留的建立过程如图 1 所示。其中,Adspec 对象用以描述本地流量控制的信息,Tspect 对象用以描述源端业务流的特性,ST(Sender Template)用于描述发送者将会产生的数据分组格式,RC 表示预留确认,Rspec 表示接收端业务流特性。

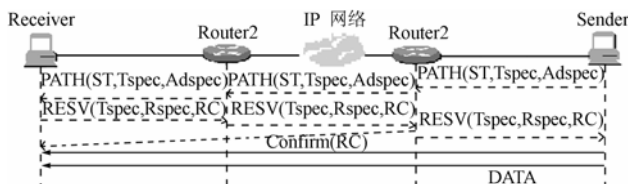


图 1 RSVP 建立过程

3 CORBA A/V Stream 规范及其 QoS 描述

传统的分布对象计算中间件,如 CORBA、DCOM 及 Java RMI 等,都建立在请求/响应语义模型上,不适用于不断增长的连续型媒体数据流的应用^[1]。为此,对象管理组(OMG)

于 1998 年 7 月发布了基于 CORBA 的 A/V Stream 规范,用于 A/V Stream 的管理和控制传输^[2],为实时分布多媒体应用提供一个开放、分布多媒体流框架,使其一方面能够利用传统分布对象计算开放的优点(如伸缩性、可重用性和可移植性等),同时又能获得所需的传输实时多媒体流的服务质量。OMG A/V Stream 服务体系模型如图 2 所示。它对传统的 CORBA 体系进行了扩展,一方面,用传统的 IDL (Interface Definition Language) 接口和 IIOP (Internet Inter-ORB Protocol) 协议管理和控制多媒体流;另一方面,将实时多媒体流的传输从 ORB (Object Request Broker) 中剥离出来,从而减少了 IIOP 协议中的频繁数据拷贝和存储空间分配,避免了 CDR (Common Data Representation) 的编解码过程,大大提高了多媒体流的传输吞吐量,减少了延迟和抖动。CORBA A/V Stream 规范还提出了一个简单流协议 SFP (Simple Flow Protocol),用于实现基于不同传输协议的多媒体流服务,通过 SFP,流服务可以在不同传输网络和协议上映射(如 UDP、ATM/AAL5) (Asynchronous Transfer Model/ATM Adaptation Layer 5)。因此, CORBA A/V Stream 服务可以屏蔽复杂的下层传输网络的控制和协议细节,使应用开发者可以在分布对象中间件层次上设计、开发和部署各种分布多媒体应用,极大地简化了开发过程,提高了维护及部署的伸缩性。

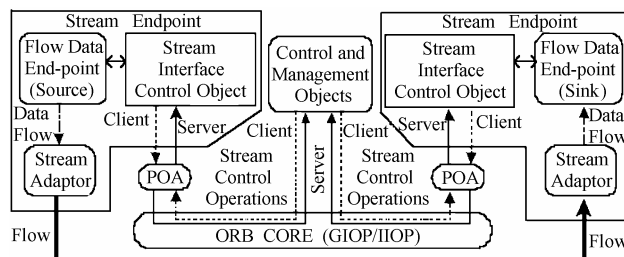


图 2 OMG A/V Stream 体系

CORBA 流服务规范定义了一种基于属性服务 (Property Server)^[8,9] 的实时多媒体流服务质量参数描述方法,用属性表示一个 QoS 参数,用属性序列表示一个 QoS 参数集。相应的 IDL 定义:

```
typedef string PropertyName;
struct Property {
    PropertyName    propert_name;
    Any property_value;
};
typedef sequence<Property> properties;
struct QoS {
    string QoSType;    //聚集多个服务质量参
    // 数并确定其属于哪一个会话流的标志。
    Properties QoSParams;
```

```
};
```

```
typedef sequence<QoS> streamQoS;
```

可以为一个媒体流的每个会话流指定 QoS, 含有两个会话流的媒体流, 一个是 Video, 一个是 Audio, 当创建这种类型的流时, 传送给 bind_dev()操作的 streamQoS 型参数如下:

```
<
{“Video_QoS”
  <
    { “Video_framerate”, 25}
    {“Video_colorDepth”,8}
  >
}
{“Audio_QoS”
  <
    {“Audio_sampleRate”,8000}
    {“Audio_numChannels”,2}
  >
}
>

<
{“Video_QoS”
  <
    { “ServerType”, 0} // Best effort
    {“Bandwidth”,1500000}
    { “Bandwidth_Min”, 1250000}
    {“Delay”,100}
    ...
  >
}
{“Audio_QoS”
  <
    {“ServerType”,1} // Guaranteed
    {“Bandwidth”,8000}
    {“Delay”,100}
    ...
  >
}
```

CORBA 流服务针对标准的多媒体流(如音频、视频媒体流), 定义了一些 QoS 参数, 分为应用层 QoS 参数和网络层 QoS 参数。应用层 QoS 参数主要是指与采集和播放设备相关的参数, 如视频流的分辨率、颜色深度、每秒播放帧数等, 网络层 QoS 参数是指一些与网络传输相关的参数, 如带宽延时和

抖动等。

4 集成 QoS 模型及其实现

4.1 集成 QoS 模型及其协商

集成服务质量模型如图 3(a)所示。其中, 用户层 QoS 是应用层 QoS 的显化, 直接面向用户, 是应用层 QoS 的用户视图。设备层 QoS 面向设备, 用于定义和描述特殊设备的各种 QoS 参数, 是应用层 QoS 的设备视图。应用层 QoS 与不同的媒体流类型相关, 用于定义和描述各种媒体流的 QoS 参数。网络层 QoS 与所传输的媒体流类型无关, 只表示传输媒体数据的质量。

根据 QoS 模型, 为不同层次的服务质量定义一个参数集, 通过指定这些参数集中的部分或全部参数值, 就可以表示不同层次的服务质量。

当定义完应用的 QoS 需求后, 流服务需要在不同的端系统间、流服务与承载网络间进行 QoS 协商, 以确定是否能够满足应用的需求。一般地, 协商有两种类型: 端到端 QoS 协商(peer-to-peer)和层到层(layer-to-layer)QoS 协商。前者是指不同端系统的应用程序之间的协商, 而后者是指同一端系统内上下层之间的协商, 即下层服务提供者能否为上层应用提供足够的服务质量。协商模型如图 3(b)所示

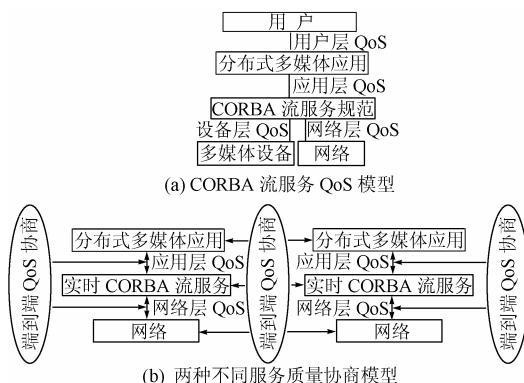


图 3 集成 QoS 模型

4.2 QoS 映射与绑定

由于不同层次的 QoS 需要用不同的参数集表示, 并且某个层次的 QoS 只能用对应于该层的服务质量参数集来表示, 因此, 需要实现不同层次的 QoS 参数的映射。其中, 要实现用户层 QoS 参数到应用层 QoS 参数的映射, 应用层 QoS 参数到网络层 QoS 参数的映射、设备层 QoS 参数到网络层 QoS 参数的映射。由于设备层 QoS 与应用层 QoS 可以使用相同的参数集, 因此, 只要实现用户层 QoS 参数到应用层 QoS 参数的映射, 以及应用层 QoS 参数到网络层 QoS 参数的映射。

用户层到应用层的 QoS 映射不是直接可见的, 分布式多媒体应用程序可以使用调节服务(tuning service)的模式实现该映射。用户可以通过调节服务提供的图形界面输入和设定

应用层服务质量,而调节服务通过设定输入的范围来表示应用层可以提供的服务质量。

应用层 QoS 到网络层 QoS 的映射,需要将表示不同连续媒体设备特征的参数(如视频图像的分辨率)转换成表示该媒体传输特征的参数(如带宽)。CORBA A/V 流服务中,定义流端点对象的 QoS_Mapping 操作实现应用层 QoS 到网络层 QoS 的映射。用户可以重载该操作扩展映射,用于特定的应用。

4.3 基于 IntServ/RSVP 的 QoS 绑定实现

实现模型如图 4 所示。其中 HPNQ(High-Performance-Network-oriented Queuing kits)是一个用于流量整形和分组调度控制的工具集,它基于 FreeBSD 操作系统内核实现,向 RSVP 提供控制接口,实现与信令协议的交互,进而定义、配置和实现 QoS 确保机制。

RSVP Implement 是 RSVP 协议的实现,它作为 FreeBSD 下的一个守护进程(Daemon)实现,完成与其它网络结点 RSVP 模块之间的协议交互,并向上层提供 API 调用接口。

CORBA A/V Stream StreamCtrl Object 实现了 CORBA A/V Stream 的 QoS 封装,并通过下层提供的接口,实现 QoS 的映射与绑定。应用层可以建立在 CORBA 基础之上,从而实现应用层的异构集成,并确保相应的 QoS 需求。

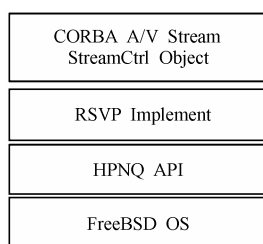


图4 基于 Implement/RSVP 的 QoS 实现模型

对于端到端 QoS 协商,通过 CORBA 流服务中的 Negotiator 对象实现,其 IDL 定义如下:

```
interface Negotiator {
```

```
    Boolean negotiate(in Negotiator remote_negotiator, inout QoS QoSspec);
};
```

当应用程序需要创建一个具有 QoS 需求的流时,发起方首先调用响应方的 Negotiator 对象的 negotiate()操作进行端到端的协商,并将己方的 Negotiator 对象的引用及所需的 QoS 需求 QoSspec 作为参数传递给对方。响应方 Negotiator 对象的 negotiate()操作,检查发起方所需的 QoS 能否满足,如可以满足,返回 TRUE,否则,将自己的可以满足的 QoS 通过 inout 参数 QoSspec 传递给发送方。如果发送方认为该 QoS 能接受,则调用流端点对象的 QoS_negotiate 操作进行流服务与下层网络的服务质量协商,即调用 RSVP 的 API 实现,为该流分配一定的资源并根据实时任务调度方法进行调

度,从而保证多媒体应用要求的 QoS。否则,就要求应用降低服务质量要求,重新进行协商。

5 原型实现及分析

上述集成QoS模型及其实现,已用于“现代远程教育工程”中央财政专项项目:“基于CORBA及QoS使能的网络教育支撑平台研究”之中^[10],原型系统的实现,逻辑上由3个部分组成:会话目录服务器、教师端应用、学生端应用。教师端应用和学生端应用以及中间的路由拓扑构成集成QoS模型的典型应用。

实验环境下对原型系统进行了测试,实验环境拓扑如图5所示,其中仿真路由器基于FreeBSD实现,仿真路由器1和2之间的网络带宽为10Mbps,我们有意设定其为瓶颈链路。会话目录服务器和教师端应用部署在实验网络中,学生端应用则分布在实验网络和校园网上,所有主机都配置FreeBSD、ORBUS¹⁾、RSVPD、摄像头、耳机和麦克风。

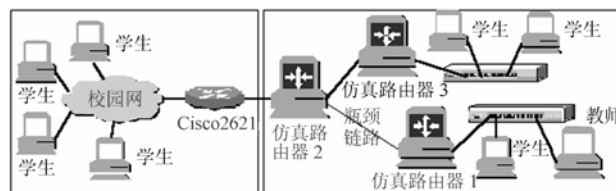


图5 原型系统测试拓扑

原型系统的 QoS 实现体现在两个层面,一是在应用层可以通过调整音视频的编码方案、采样率、帧速率、色深等来调整音视频的质量,二是在网络层通过端到端的资源预留及流量整形与分组调度机制实现网络传输时的 QoS 确保。测试步骤如下:

(1)打开应用,采用默认设置,完成教师和学生之间的交互。

(2)通过 VideoControl 和 AudioControl 调整 QoS 参数,观察到音视频效果有较明显的变化。

(3)在仿真路由器不启用预留的情况下,向瓶颈链路注入突发流量,发现音视频交互的效果显著下降,视频出现马赛克,音频产生明显抖动。

(4)启用仿真路由器的资源预留和 QoS 确保机制,在同样突发流量的情况下,应用系统音视频交互的效果保持平滑稳定。

(5)为进一步测试在不同的突发流量情况下,资源预留和分组调度机制的性能,我们取待测流量为 2Mbps,并首先将仿真路由器分组调度模块的最大队列长度设置得相对小一些(0.3MByte),测试在无预留和有预留情况下,应用数据的丢包率,结果如图 6 所示。

¹⁾实验室自己实现的CORBA2.0 规范C++版

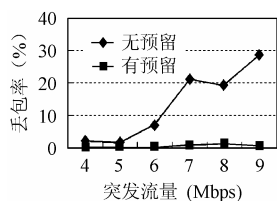


图 6 丢包率性能测试

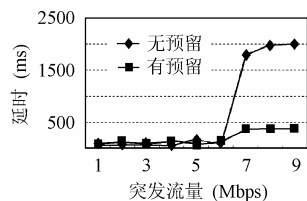


图 7 时延性能测试

(6)将仿真路由器分组调度模块的最大队列长度适当放大(3MByte),测试在无预留和有预留情况下,应用数据的时延变化,结果如图 7 所示。

试表明,在未启用资源预留的情况下,当 10Mbps 瓶颈链路上的突发流量达到 7Mbps 左右时,2Mbps 待测数据流的丢包率和端到端时延均发生明显变化,这是由于瓶颈链路上的数据流量已接近链路带宽,来不及发送的数据包只好在队列中排队较长时间,从而造成端到端时延增大;而当最大队列长度较小、队列缓冲区已满或将满时,就会发生数据包丢弃,从而造成丢包率增大。从图中可以看出,在启用资源预留机制,为待测数据流分配相应带宽的情况下,待测数据流的时延和丢包率在不同突发流量的冲击下保持相对稳定,这正是资源预留和公平队列调度作用的结果。

6 结束语

随着多媒体应用的迅猛发展,服务质量越来越重要,特别是端到端的服务质量确保。本文基于 CORBA A/V Stream 服务规范、以及 Intserv/RSVP 网络服务模型,给出了一个集成 QoS 模型,并论述相关的 QoS 协商、绑定、确保等实现策略,通过一个原型系统验证了模型的可行性,有机地将分布式对象计算技术与基础网络服务结合,建立基于 CORBA A/V Stream 服务规范与 Intserv/RSVP 基础上的 QoS 使能的分布多媒体应用开发支撑平台,以便快速、方便地开发各种多媒体应用,集成已有的应用,提供灵活的部署策略和必要的 QoS 保证。本文的工作只是一个开始,还有更多的技术没有涉及,以后的研究主要包括:其他网络 QoS 服务模型的实现与集成,HPNQ API 的扩充和完善,组播 QoS 的实现,多种传输网络的映射与绑定,各种性能的测试等。

参 考 文 献

- [1] Mungee S, Surendran M, Schmidt D C. The Design and Performance of a CORBA Audio/Video Streaming Service, in Proceedings of the HICSS - 32 International Conference on System Sciences, Minitrack on Multimedia DBMS and the WWW, Hawaii, January 1999.
- [2] Object Management Group, Control and Management of A/V StreamSpecification, OMG Document telecom/97 - 05 - 07 ed., October 1997.
- [3] Object Management Group, The Common Object Request Broker: Architecture and Specification, Revision 2.0, July 1995.
- [4] IETF Integrated Service Working Group[EB/OL], <http://www.ietf.org/html-charters/intserv-charter.html>.
- [5] IETF Differentiated Service Working Group[EB/OL], <http://www.ietf.org/html-charters/diffserv-charter.html>.
- [6] Rose E C, et al.. Multiprotocol Label Switching Architecture, IETF Draft. July 2000, <draft-ietf-mpls-arch-07.txt>.
- [7] 冯径等. 多协议标记交换技术. 北京: 人民邮电出版社, 2002/01, 第 6 章.
- [8] Object Management Group, CORBA Services: Common Object Services Specification, Revised Edition, 97 - 12 - 02 ed., Nov. 1997.
- [9] Object Management Group, Property Service Specification, 1.0 ed., July 1996.
- [10] 沈军等. 基于 CORBA 及 QoS 使能的网络教学支撑平台研究与实现 CSIT'2002 会议论文集南京, 2002/10/27 - 28: 253 - 260.

沈 军: 男, 1963 年生, 教授, 博士生, 研究方向为可定制服务网络服务体系、分布对象计算及网络教育.

董永强: 男, 1973 年生, 博士生, 研究方向为可定制服务网络服务体系、高性能网络.