

支持 IP QoS 服务质量机制的 CORBA^{*}

IP QoS-Driven CORBA

郭乐深 刘锦德 张乃靖

(电子科技大学计算机学院微机所 成都610054)

Abstract With the development and deployment of QoS technology, continuous media and real-time application, such as audio and video, are quickly becoming an integral part of distributed computing environments. Increasing demand to build a QoS-Driven CORBA, which is a widely accepted distributed object platform, becomes urgent. Firstly, this paper discusses a proposed IP QoS architecture and protocols supporting the Internet integrated services. Then a QoS-Driven CORBA is presented.

Keywords CORBA, Internet Integrated Model, Multicast, RSVP protocol, QoS Guarantee, Real Time

0 引言

作为分布式计算的一个重要规范——CORBA^[1],其主要目标是解决面向对象的异构应用之间的互操作问题,并提供了分布式计算所需的多项服务。ORB是CORBA平台的核心,用于屏蔽与底层平台有关的细节,使开发者可以集中精力去解决与应用相关的问题,而不必自己为创建分布式计算基础平台而操心。但是随着CORBA应用于分布式多媒体环境(media-rich)和分布式实时系统(mission-critical),端到端的QoS保证成为关键,服务中的带宽(Rate)、差错率(Loss)、延迟(Delay)和延迟抖动(Jitter)必须加以考虑,但是由于OMG在制订CORBA规范时,主要考虑的是互操作性,用以解决异种平台的协作问题,对于服务质量QoS的考虑较少,使得很难基于CORBA建立一个可实用的分布式多媒体环境和分布式实时系统。

在CORBA规范的研究与制定过程中,人们把大量的精力集中在CORBA组件的封闭性、继承性、多态性、可重用(reusable)、可插拔(plug-and-play)等软件特性上,而忽略了实际多媒体和实时服务的应用特性——支持服务质量保证的具体要求。

CORBA的IIOP运行在IP之上,而现在的IP网络已经发生巨大变化,目前,Internet上的应用并不能进行资源预约,于是IETF提出了Internet综合服务体系结构(ISA^[1,3-5,7]),后者定义了实现应用QoS保证的结构。ISA为上层应用提供其所要求的端到端的QoS保证,并根据其服务数据包的QoS要求选择服务类型。ISA包括的QoS控制技术有:服务类型分类,先

进的阻塞管理技术,网络带宽消耗的排队算法,资源预留协议(RSVP),限制延迟和延迟抖动等。

要建立支持服务质量机制的CORBA系统,通过它来开发分布式多媒体和分布式实时服务还有许多问题需要解决。本文首先分析当前的IP QoS技术,根据CORBA系统的特点,着重讨论了CORBA为支持QoS而应作的扩展,同时对服务质量驱动的CORBA的可行性进行了验证和分析。

1. IP 服务质量机制

1.1 综合业务体系结构——ISA

当前IETF提出了Internet综合服务体系结构(ISA),在ISA中定义了三种服务类型:QoS保证服务类型(GS, guaranteed service)、受控负载服务类型(CLS, controlled load service)和尽力服务类型(BES, best effort service)。其中保证服务类型(RFC2212):对带宽、时延、分组丢失率提供定量的保证;受控负载服务类型(RFC2211):给用户提供一种类似在网络欠载情况下的服务,它是一种定性的质量保证;尽力服务类型:类似于目前Internet网上提供的服务,是一种尽力而为的工作方式,基本上无任何质量保证。

同时ISA中定义资源预约协议(RSVP, RFC2205)为其QoS信令。通过RSVP,用户可以给每个业务流申请资源预留,要预留的资源可能包括缓冲区及带宽的大小。这种预留需要在路径上的每一跳都进行,这样才能提供端到端的QoS保证。RSVP是单向的预留,适用于点到点以及一点到多点的通信环境。

由于Internet网络是面向无连接的分布式网络,

*)本文得到电子科学研究院预研项目资助。郭乐深 博士生,主要研究领域为开放系统技术和多媒体技术。刘锦德 教授,博士生导师,主要研究领域为开放系统技术、多媒体技术和虚拟现实应用。张乃靖 硕士生,主要研究领域为安全技术。

对其发生网络阻塞的可能性较大,在ISA中提供了加权随机早期探测(Weighted Random Early Detection, WERD)型的阻塞控制机制,其原理是数据服务对于数据包丢失率十分敏感,如果业务的数据包丢失率较高的话,系统的发送速率就降低;可以使得路由器通过丢弃数据包来降低业务流的速率,即利用随机丢弃数据包以保持较小的平均队列长度,从而避免网络阻塞。

1.2 ISA的RSVP信令格式

RSVP信令由五部分所组成:即协议对象、源地址、源端口、接收端地址和接收端口,其中,协议对象具有如图1所示的共同信令头或对象头,图1所示的信令头为3个长32比特的字。

	0	1	2	3(byte)
0	Vers	Flags	Msg Type	RSVP Checksum
1	Send.TTL	(Reserved)	RSVPlength	
2	Length(bytes)		Class_Num	C.Type

图1 RSVP共同信令头

其中Vers:4比特,表示RSVP版本号,Flags:4比特,未定义,Msg Type:8比特,表示RSVP报文类型;1=PATH类信令,2=RESV类信令,3=PATHERR(路径错误)类信令,4=RESVERR(预约错误)类信令,5=PATHTEAR(路径拆除)类信令,6=RESVTEAR(预约拆除)类信令,7=RESVCONF(预约确认)类信令,RSVP Checksum:16比特,校验值,Send-TTL:8比特,被发送信令的IP TTL值,RSVP length:16比特,以字节形式表示RSVP信令,Length:16比特,总长,以字节形式表示对象的长度,该长度必须是4的倍数,Class_Num:8比特,标识对象类型,共定义19类信令,包括:0=Null(空信令),1=Session(会话信令),2=Session-group(会话组信令),3=RSVP-HOP(具有RSVP能力的节点),4=Integrity(加密信令),5=Time-value(信令报文的刷新周期),6=Error-spec(错误类型),7=Scope(指定范围),8=Style(预约风格),9=FLOWSPEC(QoS描述信令,后定义),10=FILTER-SPEC(具有期望QoS的数据包子集),11=SENDER-TEMPLATE(识别发送端信令),12=SENDER-TSPEC(发送端流量特性描述),13=AD-SPEC(路由用),14=POLICY-DATA(管理用),15=CONFIRM(要求证实接收者信令),30=DIAGNOSTIC(诊断信令),31=ROUTE(记录路径),32=DIAG-RESPONSE(诊断响应),C-Type:与Class-Num一起指明具体的信令,例如,信令SENDER-TEMPLATE的C-Type=1时,表示该信令的发送端地址为IPv4地址;C-Type=2时,表示信令的发送端地址为IPv6地址;而C-Type=3时,则表示除了发送端地址为IPv6地址之外,信令中还包含有一个相应的流标识。紧接着RSVP信令头的是RSVP的对象内容,最后是发

送和接收RSVP的有关地址。

1.3 QoS参数信令格式

图2和图3信令字长都为32比特,源端可使用图2的SENDER-TSPEC信令把发送方所具有的流量特性告知接收方,而接收方则可使用图3所示的FLOWSPEC向网络和端系统提出QoS要求。这里,保证式服务FLOWSPEC中所要求的流特性应至少不小于SENDER-TSPEC中所给出的流特性。否则,系统分配给该信息流资源将无法保证接收端所要求的QoS。

	0	1	2	3(byte)
0	信令头			
1				
2				
3	版本号	(未定义)	对象长度(字节)	
4	服务类别	(未定义)	服务数据长度(字节)	
5	参数, ID	标识	参数, 长度(字节)	
6	参数, ID	标识	参数, 长度(字节)	
7	参数, ID	标识	参数, 长度(字节)	
8	分组峰值速率 r_p			
9	分组平均速率 r_i			
10	分组平均周期 i			
11	分组最大突发长度 b			
12	参数, ID	标识	参数, 长度(字节)	
13	延迟 d			
14	抖动 j			
15	丢失率 l			
16	允许过度延迟概率 p			
17	参数, ID	标识	参数, 长度(字节)	
18	会话, 目的地址 (IP, 4)			
19	会话, 协议 ID		会话, 目的端口	
20	会话, 目的地址 (IP, 4)			
21	会话, 协议 ID		会话, 目的端口	
22	会话, 与会话, 同步程度			

图2 SENDER-TSPEC信令格式

	0	1	2	3(byte)
0	信令头			
1				
2				
3	版本号	(未定义)	对象长度(字节)	
4	服务类别	(未定义)	服务数据长度(字节)	
5	参数, ID	标识	参数长度(字节)	
6	分组峰值速率 r_p			
7	分组平均速率 r_i			
8	分组平均周期 i			
9	分组最大突发长度 b			

图3 FLOWSPEC信令格式

1.4 支持服务质量的应用程序的交互过程

下面具体说明支持服务质量程序的交互过程(现在支持服务质量机制的环境包括 WINDOWS2000、UNIX 系统或 JAVA 环境)如下:

1) 用户通过 QoS 协商界面提出多媒体应用的 QoS 要求,要求建立连接,服务质量参数包括延迟时间、丢失率、速率等等。

2) 用户启动多媒体应用程序,该程序通过 sock 机制(包括了 Winsock、UNIX 的 Socket 和 JAVA 环境中 Socket)与接收方建立路由,操作系统通过 TCP/IP 核心调用 RSVP.exe 资源预留——即 RSVP.exe 通过 IP 网络,发出 PATH 数据包送向目的地址,对经过的交换机和路由器进行资源预留。

3) 由接收方启动 RSVP 协议向路由器传送 SENDER-TSPEC 信令和 FLOWSPEC 信令等,进行资源预约。

4) 交换机和路由器接到 RESV 包文,进行相应资源预留;而接受方的主机上的 RSVP.exe 接到 RESV 包文后,先传给接纳控制,如果资源许可则接纳控制成功;同时还要进行策略接纳控制(Policy admission control),即该用户进程是否有权力预留这么多的资源;如果成功,则向发送进程发出 RESV 数据包,表示预留成功。

5) 发送方的主机上 RSVP.exe 接到 RESV 数据包后,RSVP.exe 通过业务量控制界面(TC API)向包分类器(Packet Classifier)注册这次交互的类型(即 GS、或 CLS、BES 业务等)。

6) 具体执行双方的网络交互等 socket 操作。

7) 在具体数据交互过程中,IP QoS 中的包调度器(Packet Scheduler)根据这个数据包类型(Class)号对应不同资源(调度优先级,缓存大小,延迟和峰值大小等)进行数据包的发送与接送,以保证实时服务于多媒体服务的 QoS 要求。

2. 基于 IP QoS 服务质量机制的 CORBA 的要求

连续媒体,如音频和视频,要求分布式系统支持媒体 QoS 参数(如带宽、差错率、延迟和延迟抖动)在许可的范围内,若无法保证这一关系而被打破,媒体信息的完整性将受到影响,信息的语义变得不可理解;对于分布式实时服务的要求是端到端的 QoS 参数中的延迟必须确定,否则会导致服务失败,所以无论是分布式多媒体或者分布式实时服务的要求可以归结为对分布式系统的服务质量保证要求。

具有服务质量保证的中间件 CORBA 系统主要针对:存在大量关键性任务和多媒体任务开放式环境中,通过服务质量机制管理全局的资源,以提供任务所需服务质量。针对上述连续媒体和实时任务 QoS 的特殊性质,IP QoS 技术可以较好解决,但是如何把 IP QoS

机制加入 CORBA 系统环境中,使得实现支持服务质量机制 CORBA 系统,提出了以下要求:

·支持服务质量机制 CORBA 系统应该具有显式的服务质量说明界面——即实现支持服务质量的对象及其说明界面,而原来 CORBA 系统是隐式说明。

·支持服务质量机制 CORBA 系统中的 ORB 应该充分地结合当前资源的 IP QoS 技术,使得 CORBA 系统在 IP QoS 的技术上,为上层用户进一步提供抽象、集成、自适应的基于服务质量机制的对象,而不是 CORBA 系统自己实现服务质量技术(Enforcement)。

·建立全局统一的资源管理机制,通过全局统一的资源管理机制可以:监视整个开放式系统中的计算资源和通信资源;按照用户 QoS 要求,有效分配资源满足用户服务质量要求;当前资源不足时,根据任务优先级自适应地重新分配资源。

·为用户层提供 QoS 访问控制界面。由于实现端到端的服务质量保证机制,所以服务质量保证技术对于高层用户是半透明的。下层中间件系统为高层用户提供一个半透明的 QoS 访问控制界面:1)首先对系统资源进行一定抽象,通过高层界面可见地访问控制下层开放系统资源;2)允许高层用户自适应地调节下层资源以满足自己的服务质量要求。

·建立端到端的服务质量保证机制。这是具有服务质量中间件系统的核心,它包括:1)提供高层用户之间进行明确的服务质量协商;2)系统在不同层次(网络层、操作系统层和中间层)上进行服务质量映射;3)建立反馈机制,监视端到端服务质量情况,根据开放系统环境的资源变化自适应地调节以满足用户要求。

3. 基于 IP QoS 服务质量机制的 CORBA 的设计与实现

3.1 CORBA 系统中支持 IP QoS 的服务扩展

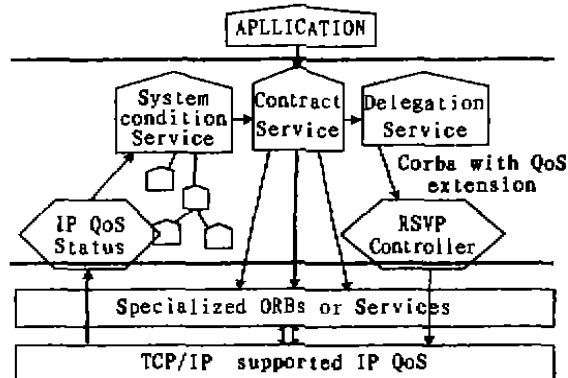


图4 合约服务、系统条件服务和代表服务

支持 IP QoS 的 CORBA 系统要在原来基础上,为加入 IP QoS 机制,我们须实现三个通用本地服务:合约服务(Contract Servicer)、系统条件服务(System

Condition Service)和代表服务(Delegation Service)。三者的相互关系如图4,其中合约服务根据系统条件服务得到的系统状态,以及对象本身的服务质量要求具体调节对象的服务质量参数。例如在图5中,CORBA中对象服务质量说明为当系统的资源大于10,对象之间正常交互;当系统资源在2-10之间,对象可以降低服务质量标准进行交互;当系统资源小于2时,系统无法

满足对象的服务质量要求,交互失败。当端系统中的合约服务得到对象类似上述的服务质量请求,同时结合系统条件服务(System Condition Service)提供的分布式系统资源情况,来选择采用哪种级别的服务质量参数;同时调用代表服务,来执行资源预留,同时改变系统条件服务中系统的资源状态。

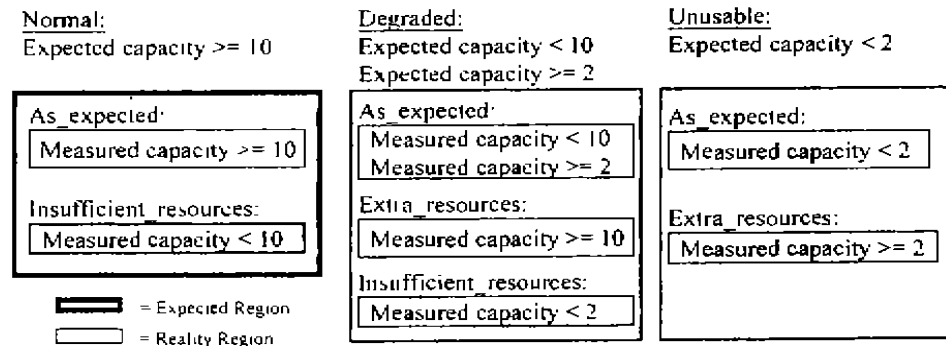


图5 对象的服务质量参数说明

系统条件服务主要是对分布式系统的资源进行测量、管理;同时接受合约服务对系统中资源的状态的设置。

代表服务检查合约服务的 QoS 参数是否合理,具体执行高层的服务质量参数到 IP QoS 的映射,同时进行资源预留。

3.2 支持 IP QoS 服务质量的 CORBA 的交互界面扩展

分布多媒体应用一般都有一定程度的服务质量要求,并且能够控制和管理 QoS. CORBA 仅在异步消息服务中才增加了简单的 QoS 框架,但它仅用于方法调用,而且多媒体的 QoS 内容要丰富得多,为此, CORBA 应在以下方面作扩展:①增加多媒体服务所需的 QoS 规范,定义不同级别的 QoS 参数,不同级别 QoS 参数对应不同的系统资源,如图5所示,②支持不同层次之间的 QoS 映射和交换机制,③支持静态和动态 QoS 管理功能,其内容包括资源协商、资源预留、QoS 监控等。

3.3 支持服务质量的 CORBA 的交互过程

我们下面具体说明支持服务质量程序的 CORBA 交互过程:

1)用户通过 CORBA 的 QoS 扩展界面提出多媒体应用的 QoS 要求,要求建立连接,服务质量参数包括延迟时间、丢失率、速率等等。

2)CORBA 系统用户启动合约服务,该服务通过系统条件服务得到系统的资源状态,同时根据对象的服务质量参数启动代表服务进行服务质量保证的资源预留过程。

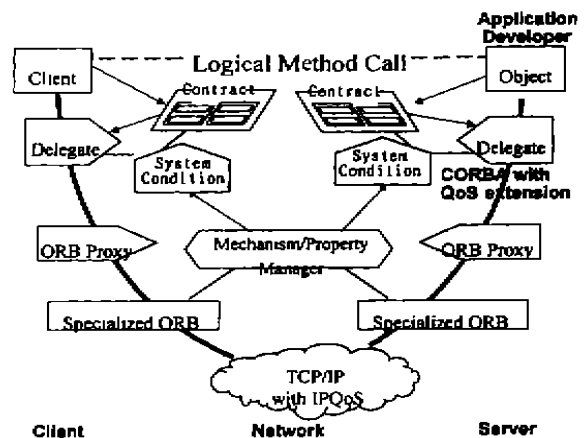


图6 支持服务质量的 CORBA 的交互过程

3)代表服务通过 IP QoS 机制与接收方建立一条具有服务质量保证机制的路由;同时交换机和路由器进行相应接纳控制,如果资源许可则接纳控制成功;同时还要进行策略接纳控制,即该用户进程是否有权力预留这么多的资源;如果成功,进行资源预留。

4)代表服务通过 RSVP. exe 向 Packet Classifier 注册这次交互的类型(即 GS、或 CLS、BES 业务等);

5)具体执行双方的网络多媒体交互或实时交互;

6)在具体数据交互过程中,IP QoS 中的 Packet Scheduler 根据这个数据包类型(Class)号对应不同资源(调度优先级,缓存大小,延迟和峰值大小等)进行数据包的发送与接送,以保证实时服务于多媒体服务的 QoS 要求。

4. 实现

在以上讨论的基础上,我们实现了一个支持服务质量的 CORBA 的原型,其结构如图7所示。

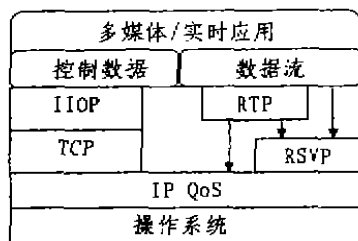


图7 一个支持服务质量的 CORBA 原型

图7中的左边部分是一般的 CORBA,主要是合约服务、系统条件服务和代表服务等之间传输控制数据和管理数据的传递(通过 ORB)。而右边部分主要针对流数据而设计。

连续媒体对于网络通信的需求主要包括足够的带宽,快速的数据转发,服务质量的保证和多投的支持,而现有的 TCP/IP 网络协议设计上的缺陷已无法完全满足这些要求,建立在现有 TCP/IP 网络之上的资源预留协议(RSVP)和实时传输协议(RTP)是针对以上需求而设计的支持连续媒体的重要支撑协议。因此,在流的通信的实现中,我们采用了 RSVP 协议和 RTP 协议。

结束语 随着高速网络技术和多媒体技术的发展,分布式多媒体应用具有巨大的潜力,支持服务质量的分布式环境的研究成为一大热点。本文结合流行的

IP QoS 技术,对当前被广泛采用的分布式对象平台 CORBA,进行了相应的改造,以实现支持服务质量的 CORBA 系统。本研究具有现实的意义。本文实践证明,在 CORBA 上构筑支持服务质量的分布式环境是可行的,而且继承了 CORBA 的开发代价小,效率高等优点。

参考文献

- 1 Barzilai T P, et al. Design and Implementation of a RSVP Based Quality of Service Architecture for an Integrated Service Internet. IEEE JSAC, 1998, 16(3): 397~413
- 2 Chatterjee S, Sydir J, Sabata B. Modeling Applications for Adaptive QoS-Based Resource Management. In: Proc. of the 2nd IEEE High Assurance Systems Engineering Workshop. Bethesda, Maryland, August 1997
- 3 Braden R, et al. Resource Reservation Protocol (RSVP)——Version 1. Functional Specification. IETF RFC2205, Sept. 1997
- 4 Wroclawski J. Specification of the Controlled-load Network Element Service. IETF RFC2211, Sept. 1997
- 5 Shenker S, et al. Specification of Guaranteed Quality of Service. IETF RFC2212, Sept. 1997
- 6 Busse J, Deffner B, Schulzrune H. Dynamic QoS Control of Multimedia Application Based on RTP. Computer Communication, 1996, 19(1): 49~58
- 7 Braden R, Clark D, Shenker S. Integrated Services in the Internet Architecture. IETF RFC 1633, June 1994
- 8 Object Management Group. Common Object Request Broker: Architecture and Specification, 2. 2 ed., Feb. 1998
- 9 Object Management Group. Control and Management of A/V Streams, OMG Technical Document Telcomm/97-05-07, May. 1997
- 10 Coulson G, et al. Supporting the Real-Time Requirements of Continuous Media in Open Distributed Processing. Computer Networks and ISDN System, 1995, 27(8): 1231~1246
- 11 王兴伟,张应辉,刘积仁,李华天. 分布式多媒体系统服务质量管理机制的研究. 软件学报, 1998, 9(2): 86~90

(上接第121页)

的操作系统中,(4)式中32比特的异或、移位都可以一步完成。

经过对 I. 363 Section 6 中给出的 3 个例样进行 CRC-32 编码,证明本方法完全正确。用两种方法在微机 Pentium-233 上对长度为 192 个字节(相当于四个信元)的数据帧循环计算 10000 次所需的时间分别为 6.56 秒和 0.72 秒。150Mbit/s 接口的信令产生速率为 2000cell/s,所以,如果采用 Pentium 233,这两种方法所需的时间分别为 0.328 秒、0.036 秒。改进法将计算速度提高了 9 倍,让出了大部分 CPU 的资源。

同时计算 16 比特的方法与同时计算 8 比特一样,但是需要 $2^{16} \times 32$ 比特(256k 字节)的内存空间作存放 $([x_{n-1}x_{n-2} \dots x_{n-16}] \oplus Z(t))D$ 各种可能的结果用,占用空间太大,不合适,这里不再赘述了。

总之,我们以牺牲很小的代价(占用 256×32 比特的内存),就可以获得较高的编、解码速度。

结束语 CRC-32 检验有很强的检错能力,实现其编码与解码的方法有许多种。本文提出的用软件模拟硬件电路实现 CRC 是在不修改通信协议的情况下较快的一种,计算机模拟计算的结果表明,编码与解码的速度提高了 9 倍。

参考文献

- 1 Seong L N, Dewar B. Parallel realization for the ATM cell header CRC. Computer communications, 1996(19): 257~263
- 2 Tong B P, Zukowski C. Parallel CRC Circuits in VLSI. IEEE Trans. Commun., 40: 653~657
- 3 ITU-T Recommendation I. 363. B-ISDN ATM Adaptation Layer Specification, 1993