基于网格的流媒体服务 QoS 管理框架及实现*)

邱建林　陶　烨　陆桑璐　陈道蓄
（南京大学计算机系　软件新技术国家重点实验室　南京 210003）

摘　要　流媒体服务是 Internet 上一类高带宽需求和高实时性约束的应用，对服务质量(Quality of Service，QoS)有较高的要求。流媒体服务的发展导致传统的 QoS 管理框架难以适应平台的异构性和复杂性。本文提出了一种基于网格的流媒体服务 QoS 管理框架，为由异构的系统构成的流媒体服务提供集成的、平台无关的 QoS 管理机制。在该框架的基础上，我们设计了一个基于网格的流媒体服务 QoS 管理系统。

关键词　流媒体服务，服务质量，网格

A Grid-based QoS Management Framework and Implementation of Stream Media Services

QIU Jian-Lin[1]　TAO Ye[1]　LU Sang-Lu[1]　CHEN Dao-Xu[1]
(State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210003)

Abstract　Stream media services are such a kind of services over Internet that require high bandwidth and have strict time constraints. The quality of service (QoS) of stream media services, embodied by the link delay, package loss rate and so on, must be carefully dealt with. The conventional QoS management frameworks, however, don't work in the extensive heterogeneous environment of stream media services nowadays. In this paper, we present a grid-based QoS management framework for stream media services which can be deployed on heterogeneous platforms. Our goal is to provide an integrated and platform-independent QoS management mechanism for stream media services. We also describe a prototype system based on this QoS management framework.

Keywords　Stream media service, Quality of Service(QoS), Grid

1　引言

近年来，随着高速存储和交换技术的发展及多媒体压缩技术的进步，流媒体服务(Stream Media Services)得以在 Internet 上广泛部署，代表性应用如远程教学、远程医疗、视频会议、视频点播等。这些服务的共性主要有：(1)较高的实时性约束。媒体数据的交付必须在一定的时限内完成。(2)高带宽需求。很多流媒体应用传输的是经过压缩的音视频数据，具有较高的信息密度。因此，必须提供一种有效的机制，保证流媒体服务的数据包丢失率和延迟等与服务性能相关的指标被控制在应用本身许可的范围之内，即提供一定的服务质量(Quality of Service，QoS)保证。

影响流媒体服务质量的因素来源于多方面，包括网络状况、服务器的 CPU 负荷、磁盘带宽，以及流媒体应用本身的特点等。流媒体应用要求提供端到端(End-to-end)的 QoS 保证，即必须同时满足通信链路上的 QoS 需求和端系统(End System)的 QoS 需求。传统的综合服务模型(Integrated Service Model，IntServ)[1]主要解决的是通行链路上的 QoS 管理问题，不能实现流媒体服务所要求的端到端的 QoS 需求。

OMEGA[2]是最早提出的一个提供流媒体服务端到端 QoS 保证的系统框架。它基于传统的 Internet 的分布式计算模型、抽象网络、应用以及操作系统自身的一些 QoS 特征，通过专门的 QoS 代理实现 QoS 监测与控制。

伴随着流媒体服务的发展，流媒体服务的系统结构从原始的单服务器、多客户机的模式向更为复杂的模式演进。流媒体服务器可以分布在不同的域(domain)内，并且以机群(cluster)的形式存在。每个域有各自不同的管理策略，各 cluster 对外提供服务的应用层协议多样化，如 http，rstp，mms 等。此外，cluster 内部各节点也是异构的，遵循不同的性能评价模型(Performance Evaluation Model)。这些问题对流媒体服务的 QoS 管理提出了新的挑战，如跨域之间的 QoS 协商没有统一的协议、服务节点的异构性加大了 QoS 监测数据的采集和交换的难度。基于 OMEGA 框架的流媒体服务很难解决这些问题。

本文提出了一种基于网格的流媒体服务 QoS 框架(Grid-based QoS Management Framework for Stream Media Services，G-QMF)，旨在提供网格环境下流媒体服务的端到端 QoS 保证。在此基础上，我们实现了一个基于此框架的初步原型系统。这个系统的最终目标是为流媒体服务提供平台无关的、集成的 QoS 管理。

本文第 2 部分介绍基于网格的流媒体服务系统的 QoS 管理框架的结构。第 3 部分给出原型系统的描述，并提出系统设计中的一些问题。最后是小结并提出了下一步的工作。

2　G-QMF 框架

G-QMF 框架的设计考虑到网格服务的一些基本特征，如互操作性、可移植性、组件的可重用性，同时针对流媒体服务独有的高实时性和高带宽要求，借鉴传统的流媒体服务的一些 QoS 管理方法，如端系统上的资源预留、接纳控制以及通信链路上的带宽预留、拥塞控制等。我们认为流媒体服务

*)本文的研究受国家 973 重点基础研究发展计划(编号：2002CB312002)资助。邱建林　硕士生，研究方向为网格计算和传感器网络；陶　烨　硕士生，研究方向为分布式信息处理；陆桑璐　博士生导师；陈道蓄　博士生导师。

系统中基于网格的 QoS 管理框架至少应符合如下两个基本的原则：

• QoS 管理的集成性原则

QoS 管理的集成性即端到端的 QoS 管理应同时保证通信链路和端系统的 QoS 需求。在通信链路上，QoS 管理必须保证应用所需的带宽和延迟等与传输相关的性能；在端系统上资源的状态被抽象为一定的服务能力，QoS 管理框架必须能根据应用对服务能力的要求合理地定位和调度资源。

• QoS 管理的平台无关性原则

基于网格的 QoS 管理框架应能以一种与平台无关的方式部署。QoS 管理框架为上层的流媒体服务应用提供标准化的接口来定义服务的特征，因此应用无需了解系统的 QoS 管理的细节即能保证与其相关的 QoS 约束得到满足。

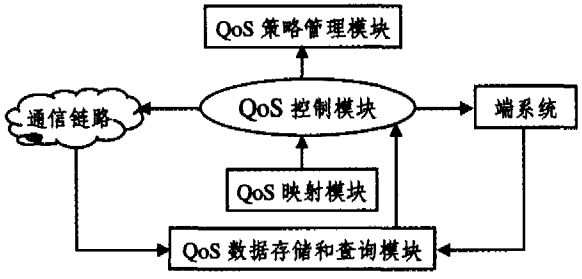

图 1　G-QMF 框架结构

我们提出的基于网格的流媒体服务的 QoS 管理框架 G-QMF 如图 1 所示。该框架由 4 个功能模块组成：QoS 映射模块；QoS 数据存储和查询模块；流媒体服务 QoS 控制模块；QoS 策略管理模块。下面对这 4 个功能模块分别做详细的说明。

2.1　QoS 映射

流媒体服务系统的不同层次所关注的 QoS 参数是不同的。G-QMF 提供一种机制，将上层 QoS 需求自动转化为底层的 QoS 需求。这样，上层的流媒体应用和客户无需了解其所请求的 QoS 在底层的细节表示。这种机制称为 QoS 的映射机制。

QoS 映射可形式化定义如下：

定义 1　设三元组 $S=\langle L, P, V\rangle$，其中 $L\in$｛用户级，应用级，操作系统级，设备接口级｝，对应 QoS 参数所在的系统层次，用户级为最高级别，设备接口级为最低级别。P 为 QoS 参数名，V 为 QoS 参数 P 的值。这样定义的 QoS 参数 S 若可被引入系统，用于 QoS 描述，称 S 在系统中。令 $M=\{S_i \mid S_i$ 在系统中$\}$。

定义 2　翻译规则 R 可定义为：$R: Q_i.P \rightarrow Q_j.P$，这里 $i>j$，Q_i，Q_j 分别为 i 级别和 j 级别的所有 QoS 参数的集合。$Q_i.P$ 和 $Q_j.P$ 分别表示 i 级别和 j 级别的 QoS 参数中参数名的集合。

定义 3　定义 QoS 映射 f_{map} 为：$f_{map}: Q_i \times R \rightarrow Q_j$。这里符号的含义同定义 2。

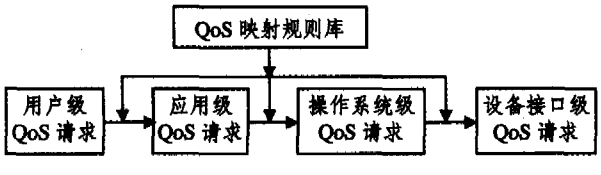

图 2　G-QMF 的 QoS 映射管理

图 2 所示的是 G-QMF 框架中的 QoS 映射管理。上一级的 QoS 请求通过查询映射规则库，被转化为一个或多个的下一级的请求。例如，用户级的 QoS 请求：〈用户级，影片质量，高〉可通过翻译规则影片质量→｛编码率，声道数｝转换为应用级 QoS 请求：｛〈应用级，编码率，512kbps〉，〈应用级，声道数，5〉｝。其他各层的 QoS 映射与此类似。

QoS 映射规则库是 G-QMF 框架中 QoS 映射管理的核心。我们将流媒体服务系统抽象成由多个 cluster 构成的一个虚拟组织（Virtual Organization，VO）。从 VO 管理者的角度来看，每个 cluster 对 QoS 请求的映射应保持一致性。鉴于此，cluster 内部的所有节点共享一个映射规则库，对外提供统一的 QoS 映射。规则库的规则来源于 VO 的管理者预定义的一组服务等级协议（Service Level Agreement，SLA）[3]。通过 SLA，提供流媒体服务提供者的 VO 与用户之间建立关于资源发现、获取和绑定的可度量的协定。规则库通过形式化 SLA 为自身添加映射规则。

2.2　QoS 数据的存储和查询

流媒体服务系统中与系统性能相关数据可按其时效性分为静态数据和动态数据两种。静态数据是系统中相对固定的数据，一般反映了系统固有的服务能力，如磁盘的设计带宽、链路的设计带宽、最大可接入的用户数等。动态数据反映系统当前的性能，如磁盘带宽占用率、链路上分组交换的延迟和流量的抖动等。G-QMF 框架应对这两类数据采取不同的管理方式。

此外，流媒体服务系统中的 QoS 数据来源于不同的数据源，在数据结构上可能存在很大的差异性，加大了 QoS 数据管理和交换的难度。QoS 管理框架需将底层的异构 QoS 数据转化为统一的数据结构。G-QMF 框架的数据采集和管理建立在 Globus Toolkit[4] 的基础上，并提供了 Globus Toolkit 的监控与发现服务（Monitoring and Discovery Service，MDS）[5] 与流媒体服务系统之间的 QoS 数据转换接口。

2.2.1　QoS 数据的存储

如图 3 所示，G-QMF 的 QoS 数据存储层次由目录层和转换层构成。

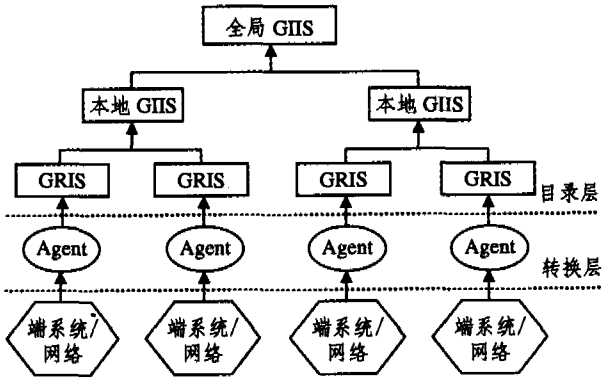

图 3　QoS 数据的存储

目录层位于数据采集模块的上层，主要提供对流媒体服务 QoS 数据的存储和查询功能。GRIS（Grid Resource Information Service）负责存储系统中的 QoS 数据。本地的 GIIS（Grid Index Information Service）提供对本地 QoS 数据的索引，全局 GIIS 提供整个流媒体 QoS 数据的索引。可以通过查询本地 GIIS 获得局部 QoS 状态，通过查询全局 GIIS 获得全局或局部的 QoS 状态。

转换层位于目录层和流媒体服务系统底层硬件之间，由部署在端系统上的多 Agent 系统构成。Agent 提供的主要功能包括：(1) QoS 数据的抽取。Agent 按照既定的策略主动获得端系统和链路的性能数据。(2) QoS 数据的整理。Agent

获得的端系统和链路的原始性能监测数据可能不符合 QoS 控制所需参数的要求，如 QoS 控制过程需要查询最近一分钟内网络的平均流量，而 Agent 采集的是每秒的网络流量，因此 Agent 需要提供一些简单的统计功能，实现原始 QoS 数据的进一步抽象。(3)QoS 数据的标准化。Agent 通过 GRIS 提供的标准接口(如通过 XML 进行信息注册的接口)向目录层注册 QoS 信息。

从存储的角度看，QoS 数据从下向上流动：Agent 从下层的实际系统中获得原始 QoS 数据，GRIS 从 Agent 获得整理后的规格化 QoS 数据，GIIS 又为下层的 QoS 数据提供索引。

2.2.2 QoS 数据的查询

G-QMF 通过 GIIS 提供的接口进行 QoS 数据的查询。为加快查询的速度，GIIS 本身提供了对查询结果的缓存功能。对缓存的控制需要考虑如下两个因素：

·缓存替换算法的选取。缓存的替换算法应考虑 QoS 数据的性质。对于静态的数据，由于在 QoS 控制中被查询的频率较高，缓存替换算法应尽量避免替换这些静态信息；对于动态数据，缓存替换算法的选取则应使得最近最可能被访问到的数据被替换的概率降低。

·缓存大小的设置。缓存的大小影响着缓存查询的命中率。采用较大的缓存，可提高查询的命中率。但随着缓存的增大，消耗的存储资源增多，且缓存匹配的开销增大。因此，必须在命中率和处理器开销之间做平衡，根据查询的特点和缓存匹配算法的特点确定一个较为合理的缓存大小。

2.3 流媒体服务系统的 QoS 控制

QoS 控制需同时提供端系统和通信链路的 QoS 保证。G-QMF 框架主要通过服务器端的资源预留和接纳控制提供端系统的 QoS 保证；通过 IntServ/DiffServ 模型提供通信链路上的 QoS 保证[6,9]。

2.3.1 端系统 QoS 控制

G-QMF 框架的端系统 QoS 控制模块的构成如图 4 所示。QoS 控制模块调用 QoS 映射模块将高级的 QoS 需求转化为底层的 QoS 需求，同时调用 QoS 状态库即 MDS 查询当前的流媒体服务系统 QoS 的状态，通过比较 QoS 需求与当前的 QoS 状态，对系统的资源进行分配和调度，使得系统遵循 SLA 所定义的服务等级约定。

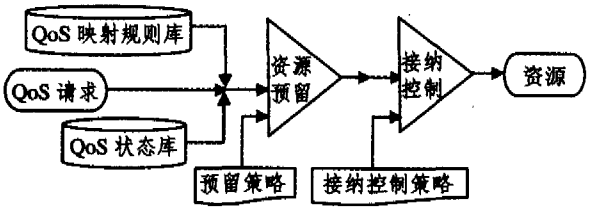

图 4 端系统 QoS 控制模块工作流程

2.3.1.1 资源预留管理

流媒体服务系统中的到达端系统的用户 QoS 请求通过 QoS 映射被转化为一定的低级 QoS 请求。根据用户和资源提供者的协定，可将 QoS 请求划分成不同的服务等级。例如，按照传输占用的带宽的大小，可将服务等级从高到低划分成 A 级、B 级、C 级、……，分别对应 512kbps，256kbps，128kbps，……的流媒体服务。资源预留的功能是为不同服务等级的 QoS 请求预留其所需的资源。端系统的总的资源量被看作一个恒定的值。当系统中的资源仍可满足当前 QoS 请求对应的服务类别需要的资源时，该请求被接受，并在当前可用的资源量中扣除该请求对应的服务类别所需的资源量。服务完毕之后，其所占用的预留资源被归还。反之，若系统中的可用资源量小于当前 QoS 请求对应服务类别的资源需求量，则该请求被拒绝。

文[7,8]中提出了区分服务模型下的资源预留的一些算法。资源预留算法不同，系统的吞吐量和资源利用率以及各服务类别之间的公平性差异明显。G-QMF 框架不定义具体的资源预留算法，而是为资源预留提供抽象的接口。具体的算法根据流媒体服务提供者的资源预留策略，由框架的实现者制定。

2.3.1.2 接纳控制

G-QMF 的接纳控制模块主要功能是保证 cluster 内部节点的负载均衡化。接纳控制模块接收到资源预留模块的接纳控制请求之后，查询 MDS 中当前各内部节点的负载状况，包括流量、CPU 占用率、可用磁盘带宽等，按照一定的调度算法为当前的服务请求分配可用的内部服务节点，为用户服务。

与资源预留模块一样，G-QMF 框架不定义具体的接纳控制算法，只提供接纳控制的接口。实际的算法根据流媒体服务的管理者自定义的接纳控制策略去实现。

2.3.2 链路 QoS 控制

我们根据文[9]所提出的 IntServ/DiffServ 混合模型对链路上的 QoS 进行控制。在端系统所在的子网内，采用 IntServ 模型，通过 RSVP 协议进行链路上的资源预留。在主干网(Backbone)上则采用 DiffServ 模型，只在网络的边缘节点(Edge)上进行服务的区分，RSVP 消息将不被中间的路由节点所识别。

2.4 QoS 策略管理

流媒体服务系统一般由多个 cluster 组成，整个系统在逻辑上被映射成一个 VO。各个 cluster 内部遵循各自的 QoS 管理策略，如资源分配策略决定如何进行资源预留和接纳控制、QoS 规则映射策略限制不同等级的用户对 cluster 内部资源的访问能力等。

cluster 之间存在多种交互的可能性。例如，为实现整个流媒体服务系统的负载平衡，需要多个 cluster 进行协商，决定由哪个 cluster 为当前的客户请求提供服务。另一个例子是流媒体服务的迁移。当前的 cluster 由于某种原因不能继续为客户提供服务时，需要将当前的服务迁移到新的 cluster 上去。

由于流媒体服务系统自身的负载平衡和服务迁移只考虑各 cluster 的当前负载状况，而不考虑这种交互可能对 cluster 内部 QoS 管理策略的影响，从而导致 cluster 内部 QoS 管理策略被损害。如 cluster 1 的 QoS 管理策略规定 B 类的服务可以访问 256kbps 或高于 256kbps 的影片，cluster 2 的 QoS 管理策略规定 B 类的服务只能访问 128kbps 的影片。当某个 B 类的服务从 cluster 1 迁移至 cluster 2 时，如强行为该 B 类服务提供 256kbps 或高于 256kbps 的影片，则与 cluster 2 的 QoS 管理策略发生冲突。

G-QMF 框架通过上层的 QoS 策略管理模块(QoS Policy Manager，QPM)实现各 cluster 之间的 QoS 策略分配和协商。加入流媒体系统的 cluster 需向 QPM 提交其 QoS 管理策略。当服务从 cluster 1 转移到 cluster 2 上时，由 cluster 1 向 QPM 发起 QoS 策略协商请求。QPM 根据 VO 管理者制定的验证策略对 cluster 1 和 cluster 2 的 QoS 管理策略进行验证。如 QoS 管理策略完全匹配，或者按照 VO 的管理者既定的 QoS 策略调整原则，可以将 QoS 控制策略降级或升级到 cluster 2 的 QoS 控制策略，则向 cluster 1 报告策略协商成功，否则向其报告策略协商失败，不允许服务从 cluster 1 迁移到 cluster 2 上去。

3 ND-3SQ：G-QMF 框架的实现

3.1 ND-3SQ 系统结构

根据上面提出的 G-QMF 框架，我们设计了一个名为“ND-3SQ”的原型系统，以实现对流媒体服务系统有效的 QoS 管理。系统的结构如图 5 所示。

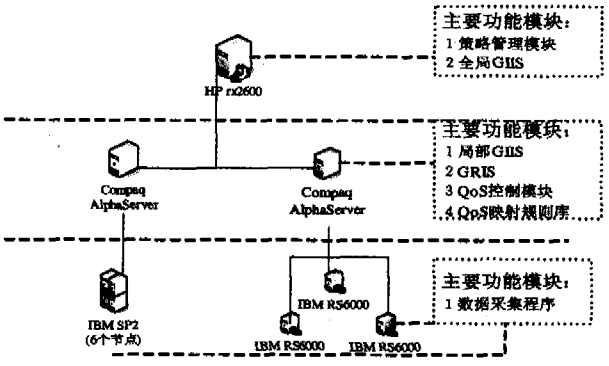

图 5　ND-3SQ 原型系统

ND-3SQ 系统中的策略控制模块部署在一台 HP rx2600 服务器上。QoS 控制模块分别部署在两个 Compaq AlphaServer 上，对我们的流媒体服务系统中的两个 cluster 进行 QoS 控制。后台的流媒体服务的两个 cluster 分别由一台 IBM SP2 和 3 台 IBM RS6000 服务器构成。

MDS 的 GRIS 部署在 Compaq AlphaServer 上，负责存储后台的 cluster 的 QoS 数据。GIIS 服务部署在上述两个 Compaq AlphaServer 和 HP rx2600 服务器上，Compaq AlphaServer 上的 GIIS 索引 GRIS 中的 cluster 的 QoS 数据，以及接受本地的 QoS 控制模块的 QoS 查询；HP rx2600 服务器上的 MDS 提供对两个 Compaq AlphaServer 的 QoS 数据的索引。值得一提的是，MDS 服务还负责对整个系统的影片资源进行管理，支持对系统中的各类别的影片的位置和副本分布状况的查询。

QoS 映射规则库同样放置在上述两个 Compaq AlphaServer 上。此外，我们为 SP2 各节点和每个 RS6000 服务器上提供了专门的数据采集程序，按标准的 XML 格式向 Compaq AlphaServer 上的 GRIS 提供 QoS 数据。

SP2 和 RS6000 机群加入流媒体系统中时，用 XML 文档向 Compaq AlphaServer 上的 QoS 规则库提交 SLA，并由规则库负责将其转化为库中的映射规则。同时，SP2 及 RS6000 机群的 QoS 管理策略以 XML 文档形式提交给 HP rx2600 服务器上的 QoS 策略管理模块。

流媒体会话开始阶段，用户的请求提交给 Compaq 服务器上的 QoS 控制模块后，QoS 控制模块根据 QoS 映射规则库和 MDS 中的 QoS 状态信息决定是否接受该请求。请求被接受后，系统为该请求进行资源预留。如资源预留成功，转入接纳控制，为该请求分配合理的内部服务节点。随后，系统通过 MDS 对流媒体服务的会话状态进行定期刷新。

流媒体会话结束时，Compaq AlphaServer 上的 QoS 控制模块负责收回预留的资源，并更新 MDS 中的会话状态信息。

3.2　QoS 数据查询缓存管理

在 G-QMF 框架的讨论中，我们提到了缓存替换策略对 QoS 数据查询效率的影响。在缓存一定的情况下，我们分别按照先进先出(FIFO)、后进先出(LIFO)以及随机选择三种替换算法进行缓存的替换。在 ND-3SQ 中，我们用 10 个客户端在 10min 之内分别对一个 Compaq 服务器后台的不同节点随机产生请求，每个客户端发送请求间隔的时间为 20s。我们的数据采集程序被设定为每 1s 报告一次后台节点的流量。在 Compaq 服务器上的 QoS 控制模块中测得到 3 种算法的平均响应时间结果如图 6 所示。

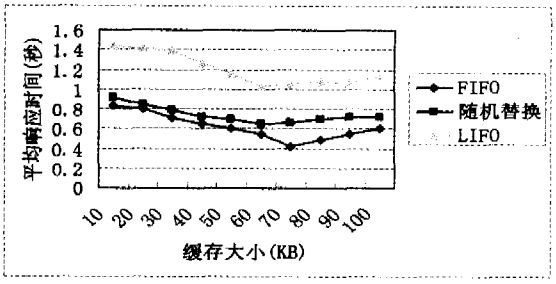

图 6　不同缓存替换算法的平均响应时间

从图 6 中可以看出，LIFO 算法具有较长的响应时间。这是由于我们设计的接纳控制算法需要查询最近 30s 内各后台节点的流量，因此最近被查询的那些关于流量的 QoS 数据被访问到的概率较高，所以 LIFO 算法采用的替换最近一次查询的结果的方法导致缓存命中率降低，从而表现出较长的响应时间。相比之下，FIFO 算法的命中率较高，因此响应时间最短。

另外，从图中可以看出，随着缓存的增大，QoS 状态查询的平均响应时间经历了从快速下降到缓慢增加的过程。这表明，缓存的增加在提供命中率的同时，导致了系统对缓存的管理的开销增大。

小结及下一步的工作　为了解决复杂结构的流媒体服务系统中的 QoS 管理问题，我们提出了一种基于网格的流媒体服务的 QoS 管理框架——G-QMF。根据该框架，我们开发了一个实际的 QoS 管理系统——ND-3SQ，为我们的流媒体服务系统提供平台无关的、集成的 QoS 保证。本文详细介绍了 G-QMF 各模块的功能和 ND-3SQ 的结构，并对 ND-3SQ 设计中的一些问题做了初步的探讨。

我们所设计的 ND-3SQ 系统只是初步的原型系统，许多完善系统的工作还要去做。如链路的 QoS 控制在我们的系统中仍未实现。近期的工作主要实现系统对链路 QoS 的控制。另外，我们将考虑对 G-QMF 的结构做拓展，使其可以应用到除流媒体服务之外的其它服务领域中去。

参 考 文 献

1　IETF Working Group on Integrated Services. http://www.ietf.org/html.charters/intserv-charter.html

2　Nahrstedt K, Smith J. Design, Implementation and Experiences of the OMEGA End-Point Architecture. IEEE JSAC, Special Issue on Distributed Multimedia Systems and Technology, 1996,14(7):1263～1279

3　Keller A, Ludwig H. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management,2003,11(1):57～81

4　Foster I, Kesselman C. Globus: A Metacomputing infrastructure toolkit. International Journal of Supercomputer Applications, 1998,11(2):115～129

5　Czajkowski K, Fitzgerald S, Foster I, et al. Grid Information Services for Distributed Resource Sharing. In:10th IEEE International Symposium on High Performance Distributed Computing, San Francisco, IEEE Computer Society Press, Los Alamitos, CA, 2001.181～184

6　Borgonovo F, Fratta L, et al. End-to-end QoS Provisioning Mechanism for Differentiated Services. IETF Internet Draft, 1998

7　Choudhury A K, Hahne E L. Dynamic Queue Length Thresholds for Shared-Memory Packet Switches. IEEE/ACM Trans Networking, 1998.130～140

8　Hahne E L, Choudhury A K. Dynamic Queue Length Thresholds for Multiple Loss Priorities. IEEE/ACM Trans Networking, 2002,10(3)

9　Bernet Y, Yavatkar R, Ford P, et al. A Framework for Integrated Services Operations Over DiffServ Networks. Internet RFC 2998, 2000