# QoS over IP 的机制与协议结构

Mechanism QoS over IP and its Protocol Architecture

王海兵　陈良刚　黄　凯　张根度

（复旦大学计算机科学系　上海200433）

**Abstract**　This paper first analyzes the need for QoS over IP, then various IP QoS protocols are introduced and compared. Last, we discuss a QoS model integrating all IP QoS protocols which can support "Top to Bottom, End to End" QoS.

**Keywords**　QoS, IP, RSVP, MPLS, DiffServ, SBM

## 1　QoS over IP 的需求

传统的运行 IP 协议的网络提供"尽力而为"(Best Effort)的数据传输服务。这种服务方式把复杂性留在端结点上，而网络内部则保持相对的简单，可扩展性良好是这种方式的特点，这一点已由传统因特网的成功证明了。当越来越多的宿主机连入因特网而使得需求超过网络负载能力时，新的服务请求不会被拒绝，而是导致网络服务"优雅"地降级，这个特点对于传统的因特网应用并没有太大影响，例如：电子邮件，文件传输和 Web 访问等。

端到端原则是传统 IP 网络成功的一个原因，其内容是："网络最终能实现什么功能由用户自己来决定，试图通过网络内在的智能性来增强网络功能的做法是多余的；网络功能尽可能地在网络之外实现。"这个原则与"尽力而为"是一致的。

当新的因特网应用，诸如多媒体传输之类的实时应用出现时，"尽力而为"服务不能很好地适应，传输延迟、抖动和包丢失极大地影响了实时应用的效果。虽然单纯的提高带宽对于实时应用是必需的，但却远不是足够的。即使在没有其他负载的情况下，网络延迟也会很大幅度地变化，给实时应用带来影响。于是，对因特网的基础需要作一些修改，在网络内部结点加入某些智能来区分出那些有 QoS 要求的数据包并对其作相应的处理来达到 QoS 要求，这就是 QoS 协议要做的事情。

从学术角度来看，改变因特网的服务模型是一个巨大的变化，但是 IETF 的工作组希望它的影响能尽量小。于是新的成分与机制并不取代原有的因特网结构，而只是作为补充。

## 2　QoS 协议

QoS 的定义有很多种，简单地说，QoS 就是网络元素(包括路由器、宿主机等)对数据传输提供一定程度的一致性保证的能力。

一些应用比其他应用有更严格的 QoS 要求，从这个角度出发，有两类基本的 QoS 类型。

·资源预留(Resource Reservation)：网络资源由应用的 QoS 请求来分配。

·分配优先级(Prioritization)：网络流量被分类并赋予不同的优先级别来进行处理。

QoS 可以对单个的应用"流"(flow)实现，也可以对流聚集(aggregate)实现，从这个角度看，QoS 又分为另两种类型。

·对于单个流：流定义为单独的，单方向的，在两个应用(如发送者与接收者)之间的数据流。由一个五元组唯一标识，五元组定义为(传送协议，源地址，源端口号，目标地址，目标端口号)。

·对于流聚集：一个流聚集简单地说，就是两个或者更多的流。一般来说，这些流有一些相同的元素(例如五元组中一个或多个元，一个标记或优先级等)。

目前已经有的 QoS 协议有以下几种：

·RSVP(ReSerVation Protocol)：提供网络资源预留的信令。

·DiffServ(Differentiated Services)：对网络数据流作简单粗略的分类并赋予优先级。

·MPLS(Multi Protocol Label Switching)：给网络数据流分配标记(Label)并以标记作为交换基础，而不是 IP 地址。

·SBM(Subnet Bandwidth Management)：在共享

王海兵　硕士生，主攻方向为计算机网络。张根度　教授，博士生导师。

及交换的IEEE802网络的第二层实现分类与优先级。

这些协议的出现是因不同的需要而产生的，它们实现的QoS等级也有所不同，表1给出了比较。

表1

| QoS程度 | 网络层 | 应用层 | 描述 |
|---|---|---|---|
| 最大 | X | | 规定的端到端资源(例如私有网) |
| | X | X | RSVP |
| | X | | MPLS |
| | X | | DiffServ |
| | X | | 公平队列算法(例如CFQ、WFQ、RED) |
| 最小 | | | 尽力而为服务 |

### 2.1 资源预留RSVP

RSVP被宿主机用来为应用的数据流向网络申请特定的资源需求，也被路由器用来在数据流传输的网络通路上建立并维持一些资源预留状态从而提供所要求的服务。在协议栈中，RSVP运行在IPv4或IPv6之上，占据运输层位置。但它本身并不传输任何高层的数据。它只是一个因特网控制协议，类似于ICMP、IGMP或路由协议。RSVP是在后台实现的，如图1所示。

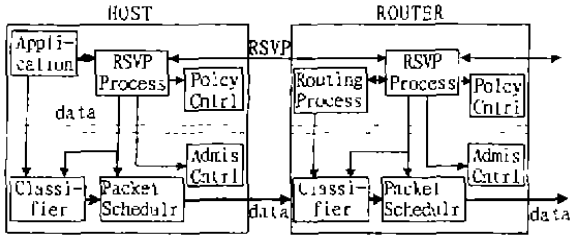


图1 宿主机及路由器中的RSVP

RSVP不是一个路由协议，它需要与现有的和未来的路由协议一起工作，从本地路由协议那里获得路由，RSVP只和在这些路由上传送的数据流获得的QoS相关。为了有效地适应多种接收者要求，RSVP让接收者负责申请特定的QoS。图2是RSVP建立预留的过程。

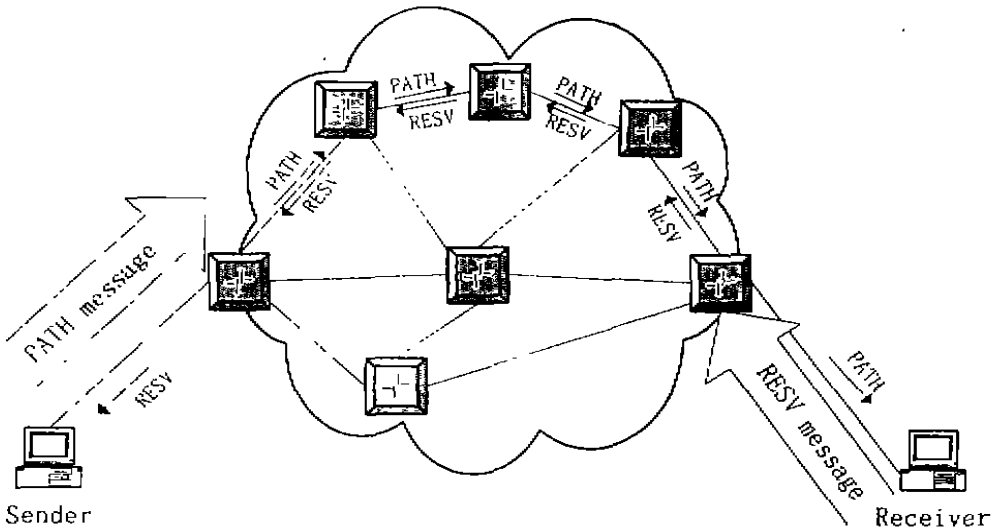


图2 RSVP建立预留的过程

简单来说，RSVP的工作过程是这样的：

·发送者确定要发送的数据流具有什么样的特性，对于带宽、延迟和抖动确定其上界和下界。发送者把这些规格(TSpec)放在PATH消息里向目标地址传输，在这条数据通路上，每一个支持RSVP的路由器都为它建立一个"路状态"(Path State)，其中包含上一跳的地址。

·为了进行资源预留，接收者向上游(upstream)发送RESV消息，其中包含请求规格(RSpec)，指示需要的服务类型，RESV消息还包含一个过滤器(filter spec)，用来对预留的包进行限定。RSpec和filter spec一起构成流描述(flow descripter)，路由器用它来标识每一个预留。

·上游的路由器收到RESV消息后，使用admission control过程鉴别请求并分配必要的资源。如果请求不能被满足(由于缺少资源或者权限不够)，路由器返回一个错误给接收者。如果请求可以满足，路由器继续向上游的下一个路由器传送RESV消息。

·当最后一个路由器接到RESV消息并接受请求时，它发一个确认消息给接收者。

·当一个RSVP会话结束时，显式的拆除过程被实行。

RSVP 提供的 IP QoS 是最高等级的，同时也最为复杂，实现 QoS 的代价过高使得它不可能在网络的所有范围内实行，简单的 QoS 技术仍然是需要的，例如 DiffServ。

### 2.2 区分服务 DiffServ

区分服务提供一种简单、粗略的方法对服务和各种应用进行分类，并对它们赋以不同的优先级，DiffServ 的协议机制体现在 DS 字节中，使用 IPv4时，DS 字节是 TOS(Type of Service)字节，使用 IPv6时，DS 字节是 Traffic Class 字节，如图3所示。DS 字节并没有保留 RFC1349中 TOS 比特的定义，但 IP Precedence 比特被保留了。

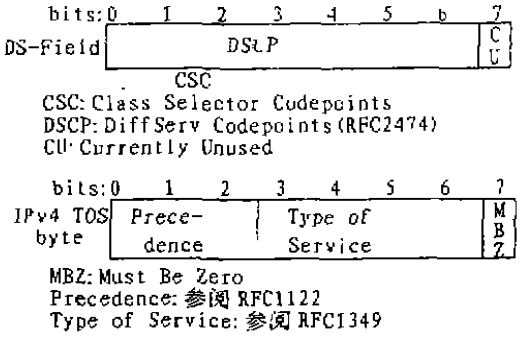


图3 DS 字节与 TOS 字节

DiffServ 简单且具有强大的灵活性，当 DiffServ 使用 RSVP 参数或者特定的应用类型来标示流量时，将很有可能建立优化定义的流聚集，实现确定带宽的管道，使得在高效共享网络资源的同时，仍提供保证服务。

### 2.3 多协议标记交换 MPLS

MPLS 与 DiffServ 从某种角度来看很相象，比如说它们都在网络入口处打上标记，在出口处去掉标记，但是，这标记的作用是不同的，DiffServ 的标记作用是决定数据包的投递优先级，MPLS 的标记作用却是决定下一跳的目的地址。MPLS 不由应用层控制，也就是说，没有 MPLS API，在宿主机内也不会有 MPLS 存在，MPLS 只在路由器中发挥作用。MPLS 可以和多种网络层协议协同工作，如 IP、IPX、ATM、PPP，或者直接处于数据链路层之上。

与其说 MPLS 是 QoS 协议，倒不如说它是流量工程协议。MPLS 用来建立象 ATM 或 Frame Relay 中的虚电路那样的确定带宽管道。

MPLS 简化了路由工作，它主要是通过减小数据包头来提高效率的，以下是支持 MPLS 的路由器(LSR)的处理过程概略。

·在 MPLS 网络里的第一跳，路由器根据目标地址或者报头里的其它信息作出投递决定，然后确定一个标记(label)，标记标识了数据包属于哪个等价类(FEC)，标记附着在数据包上被送往下一跳。

·在下一跳中，路由器使用标记值作为索引查下一跳的地址与新的标记值，新的标记又附着在数据包上被送往下一跳。

有 MPLS 标记的数据包在网络里所走的路被称为标记交换通路。MPLS 的思想是通过使用短的标记来确定下一跳的地址而不是根据较长的 IP 地址。这样路由器的工作可以减轻，并使得在 IP 网络里，包的传输更加类似于简单的电路交换。

一个比较复杂的方面是在 MPLS 路由器之间的标记分配与管理。必须保证路由器对各种标记的解释是一致的，标记分配协议专门被设计用来达到这个目的，但是也有其他一些方案，例如使用 RSVP、BGP，所以很可能会有多个标记分配方法。

### 2.4 子网带宽管理 SBM

QoS 保证只相当于整个数据通路上最差的链路，所以端到端的数据通路上每一个路由器都必须支持 QoS。一些链路层技术是支持 QoS 的，例如 ATM，但是其他更多的局域网技术例如以太网是不支持 QoS 的，它是共享广播媒体的技术，即使在交换形式下，它提供的服务仍然类似于"尽力而为"的 IP 服务，不确定的时延会影响实时应用。但 IEEE802.1p、802.1Q、802.1D 标准已经定义了以太网交换机把帧分类从而加快实时数据传输的方法。IETF 的 ISSLL(Integrated Service over Specific Link Layers)工作组正在定义高层的 QoS 协议怎样映射到链路层，例如以太网。这导致了"子网带宽管理协议"(Subnet Bandwidth Manager，SBM)的发展。SBM 是一个信令协议，它使得802局域网映射到高层 QoS 协议成为可能，图4是 SBM 的结构。

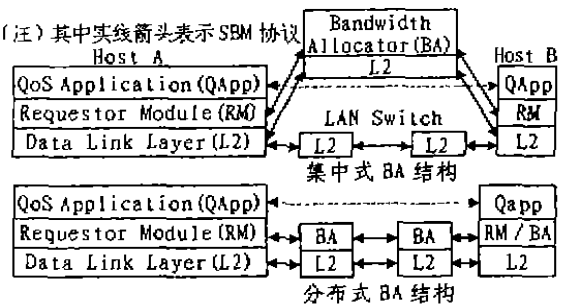


图4 SBM 的两种结构

从图中可以看出，在 SBM 结构中，除了应用层和链路层，最主要的两个逻辑模块是：

·带宽分配模块(BA)，负责维护子网内资源分配

状态并根据可用的资源和预定的权力限定进行资源请求的管理。

·请求模块(RM)只存在于端系统中，负责把高层的 QoS 协议参数映射到链路层参数，例如若使用 RSVP，它可以根据不同的 RSVP 参数(TSpec、RSpec 或者 FilterSpec)进行映射。

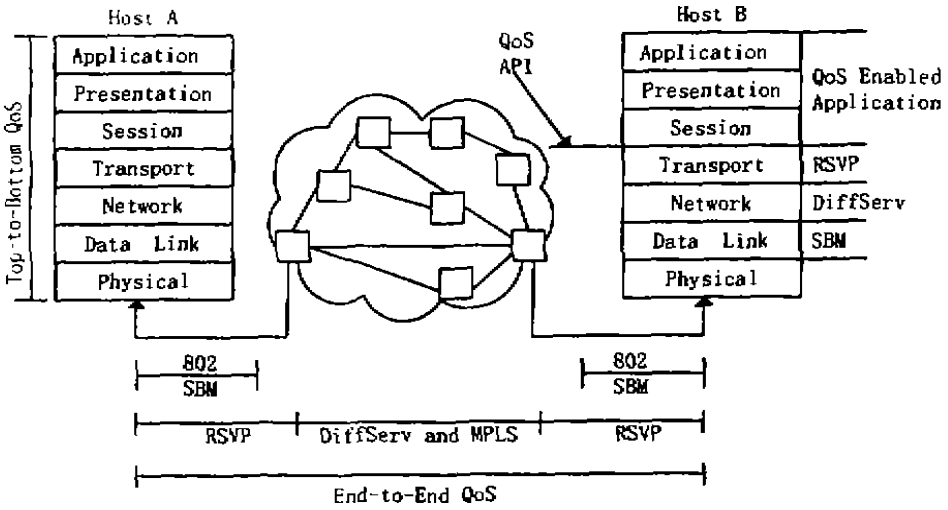


图5 自顶向下、端到端 QoS 协议结构

## 3 QoS 结构

以上一些 QoS 协议都能够端到端独立使用，但实际上它们相互之间并不是独立的，在发送者和接收者之间，多种 QoS 协议都会被使用，一起提供"自顶向下，端到端"的 QoS，图5所示为这些协议在一起协同使用的结构。

把所有 QoS 协议紧密联系在一起的定义还没有标准化，但这些工作都在进行当中，图5给出的结构是较有代表性的一种。RSVP 为网络流量规定带宽，DiffServ 则只是为网络流量指定优先级。RSVP 相对 DiffServ 来说，复杂得多，对路由器的要求也高得多，这样势必会对主干网路由器的性能造成影响。所以主干网上 RSVP 的使用是应该受到限制的，这就是为什么 DiffServ 在主干网上能够存在的原因。DiffServ 是 RSVP 很好的一个补充。宿主机可以进行 RSVP 请求，主干网入口路由器把它映射到 DS 字节，而主干网出口路由器又把它复原为 RSVP 并送至最终目标。这种以 RSVP 作为边缘，以 DiffServ 作为核心的结构目前有较大的支持。其他还有一些 QoS 结构，本文不再赘述。

**总结** 直到今天，IP 仍然在提供"尽力而为"服务，网络资源被平等地使用，但是对 QoS 的要求越来越急迫，一些基于 IP 的 QoS 协议已经出现并不断发展。由于各种应用对 QoS 的要求不同，出现了多种 QoS 协议。这些 QoS 协议在未来的网络中将结合起来工作，为用户提供"自顶向下，端到端"的 QoS。QoS over IP 的标准尚不完备，也还有许多问题需要进一步的考虑、例如支持多播、但是工作已经在许多 IP 网络上开始进行了。

## 参 考 文 献


1 Stardust.com Inc White Paper-QoS protocols & architectures. QoSforum.com 1998

2 Braden R,et al. Resource ReSerVation Protocol (RSVP)-Version 1 Functional Specification,RFC2205,1997

3 Braden R,Clark D,Shenker S. Integrated Services in the Internet Architecture:an Overview,RFC1633,1994

4 Shenker S,Partridge C,Guerin R. Specification of Guaranteed Quality of Service,RFC2212,1997

5 IETF Multiprotocol Label Switching Working Group. Available at:http://www.ietf.org/html.charters/mpls-charter.html
http://www.ietf.org/ids.by.wg/mpls.html

6 Huitema C. Routing in the Internet,Prentice Hall PTR,1995