利用资源预留协议 RSVP 实现 QoS

黎春阳
北京邮电大学电信工程学院，北京（100876）
E-mail: yangyang200622@yahoo.com.cn

摘 要： 随着信息时代的到来，各种类型数据业务的增多，互联网的服务质量保障（QoS）得到越来越广泛的重视。RSVP(Resource Reserve Protocol)是开发在 Internet 上的资源预留协议，能有效的把 QoS 引入到网络中来。本文介绍了资源预留协议（RSVP）的信令及消息特点，对其工作原理、消息的传播流程、节点处理流程、QoS 的实现原理也进行了较为详细的论述，并且提出了一种动态 QoS 调整算法，有效提高网络资源利用率。

关键词：资源预留协议、QoS 保障、资源预留

1．引言

当今世界，是一个数据资源爆炸的世界。但现有的基于 IP 的互联网，只能向用户提供尽最大努力传送（Best of Effort）的服务。随着网络用户数量和种类的不断增加，需要在 TCP/IP 体系中支持具有不同特性的业务，尽最大努力传送服务无法满足不同特性业务对服务质量的不同要求，由此带来的网络不稳定性将加大网络维护的难度和成本。如何使互联网能够提供服务质量 QoS（Quality of Service）保障，确保网络提供可预测的稳定性服务，对于网络运营者和网络使用者而言，都具有重大意义。

传统的互联网业务只需要网络提供 BS 服务即可。但越来越多音频和视频数据业务的出现，尤其是军事网络，要求网络在包丢失率、时延、带宽等方面提供端到端的保障。基于 TCP/IP 的资源预留协议 RSVP，正是为了满足这种需求而产生的。RSVP 是网络控制协议，非路由协议，但同路由协议协同工作，属 OSI 七层协议栈中传输层。利用 RSVP 预留一部分网络资源（即带宽），能在一定程度上为流媒体的传输提供 QoS。

2．RSVP协议概述

RSVP 协议是一种可以提供音频、视频、数据等混合服务的互联网络综合服务（IIS Internet Integrated Service) [RSVP97,RFC1633]。RSVP 工作在 IPv4 或 IPv6 上，处于 OSI 七层协议中的传送层。其基本原理是：数据发送端利用 RSVP 协议，向网络申请有一定参数的 QoS；RSVP 消息沿数据流传送的路径逐跳传送，最终到达数据的接收方；接收方对申请的 QoS 进行确认，并沿数据流传输的反向逐跳预约；各个路由器节点，利用 RSVP 对资源进行预留，并维持该状态直到应用程序释放这些资源，从而为特定的应用程序提供有保障的数据流服务。

RSVP 对资源进行单向申请，由数据流的发送端发起资源预留申请，接收端对预约确认。这就意味着 RSVP 在申请资源的过程中，发送端和接收端是逻辑上完全不同的两个部分。

RSVP 工作在 IPv4 或 IPv6 上，处于 OSI 七层协议中的传送层。但是，RSVP 并不处理传送层的数据，其实现通常在后台执行，而不是出现在数据传送的路径上。RSVP 本身并不是路由协议，它和路由协议协同工作，通过本地的路由数据库来获取路由信息。

RSVP 采取由接收端发起服务质量（QoS）申请的策略。接收端根据即将传输的数据流的特性（从发送方发出的 RSVP 消息中提取），提出 QoS 请求。该请求将递交给数据传送的反向路径（接收端至发送端）上的各个节点（路由器或是主机）进行资源的申请。

总体而言，RSVP 有三种基本功能：

- 路径的建立和维护；
- 路径拆除；
- 错误通知。[2]

2.1 RSVP 信令

RSVP 的信令由 5 部分组成，如表 1 所示。

表1 RSVP 信令

协议对象	
源地址	源端口
目的地址	目的端口

其中协议对象部分，由公共的头部和不同的对象体（不同种类的 RSVP 消息有所不同）构成。具体如表 2、表 3 所示。

表2 RSVP 协议的公共头部

0	1	2	3
Vers	Flags	Msg Type	RSVP Checksum
Send_TTL		(Reserved)	RSVP Length

RSVP 消息的公共头部有2个长32比特（bit）的字组成。具体定义如下：

Vers: RSVP 版本号，4 比特

Flags: 标识，4 比特

Msg Type: RSVP 报文类型，8 比特

RSVP Checksum: RSVP 校验值，16 比特

Send_TTL: 被发送信令的 IP 生存期，8 比特

Reserved: 保留，未定义

RSVP Length: RSVP 包的字节长度，包括公共头和随后的可变长度对象。

表3 RSVP 协议消息对象体

0	1	2	3
Length(bytes)		Class-Num	C-Type
Object contents			

每一个 RSVP 协议包含了 1 个长 32 比特的字的对象体头部和 1 个或多个长 32 比特的字的具体对象体。对象体头部具体定义如下：

Length: 以字节形式标识对象长度。

Class-Num: 分类号，表示对象类型，共定义了 19 类对象类型。如会话信令 Session、预约风格 Style、QoS 描述信令 FLOWSPEC 等。

C-Type: 与 Class-Num 一起，定义具体信令。

不同的 RSVP 消息，其信令的对象体部分，具有不同的 Object contents。

2.2 RSVP 的消息类型

表4 RSVP 的消息类型

消息类型	说明
Path	用来建立和维护保留
Resv(Reservation)	响应path消息, 已建立和维护保留时发送
PathTear	类似于 Path 消息，但用来从网络中删除保留
ResvTear	和 Resv 类似，但用于从网络中删除保留
PathErr	由 Path 消息中检测到错误的该消息的接收者发送
ResvErr	由 Resv 消息中检测到错误的该消息的接收者发送
ResvConf	返回给 Resv 消息的发送者以确认给定的保留已经实际安装了，该消息为可选发送的
ResvTearConf	和 ResvConf 类似，以确认已经从网络中删除了给定的保留
Hello	RFC 3209 中定义的一个扩展，考虑在两个直连的 RSVP 邻居之间的 linklocal(链路局部)的 keepalive

表 4 给出了目前已经定义的九种 RSVP 消息。不同的 RSVP 消息，其信令结构中的对象体部分，包含不同的 Object contents。其中 Path 消息和 Resv 消息是最重要的两种 RSVP 消息，在这里给出更详细的说明。

2.3 Path 消息

每个 Path 消息含有如下的信息：（包含在对象体中）

- Session：指明了目的 IP 地址（(DestAddress)、使用的 IP 协议 ID 号和可选的目的端口号，以确定一个特定的会话。该信息也是在每个 RSVP 消息中必须包含的对象。
- Time_Values：包含了消息创建者使用的刷新周期。
- Phop：指出转发该 Path 消息的最近一个支持 RSVP 节点的地址。这个地址在该路径的每个支持 RSVP 路由器上都被更新。
- Sender_Template：定义一种发送者的过滤器规范，包括发送者的 IP 地址和可选择的发送者端口。
- Sender_Tspec：定义发送者的传输特性。（参看 GS 服务中的 Tspec 参数说明）
- Adspec：可选项，含有 OPWA(One Pass With Advertising)信息，使得接收者能计算出应保留的资源级，以获得给定的端到端 QoS。这些信息在该路径的每个支持 RSVP 的路由器上都被更新。尽管 Adspec 是一个可选项，但它含有接收者用来计算 QoS 的重要信息。

Adspec 是由一个消息头、一个缺省通用参数 DGP (Default General Parameters)段以及至少一个 QoS 段组成。目前，RSVP 支持保证服务(GS)和被控负载服务(CLS)两个基本的 QoS 类，省略 QoS 段的 Adspec 是无效的。

（1）Adspec 的缺省通用参数段 DGP

所包含字段如表 5 所示，在路径上每个支持 RSVP 的路由器都要更新这些参数，以便将端到端的值表示成接收者。

表5 Adspec 的缺省通用参数段 DGP

字段	说明
最小路径等待时间	指在路径上单个连接等持时间的累加和，表示无任何排队延迟的端到端等待时间。在 GS 中，接收者可以使用该值计算端到端排队延迟限制，以及所有端到端延迟限制。
路径带宽	指在路径上单个连接带宽的最小值
全局中止位	标志位。发送者创建 Adspec 时，该位置 0。路径上任何不支持 RSVP 的路由器都可将该位置 1，以通知接收者 Adspec 是无效的。

综合服务(IS)网段(hop)计数	在路径上每个支持 RSVP/IS 的路由器都将该值加 1
路径最大传输单元(PathMTU)	路径上单个连接最大传输单元(MTU)的最小值

（2）保证服务段 GS

保证服务段 GS 所含字段如表 6：

表6 保证服务段 GS

字段	说明
C_{tot}	端到端偏差项 C 的总和
D_{tot}	端到端偏差项 D 的总和
C_{sum}	自上次刷新点开始 C 的总和，在分布树的某些点上，被用于刷新处理。
D_{sum}	自上次刷新点开始 D 的总和，在分布树的某些点上，被用于刷新处理。
GS 中止位	标志位，发送者创建 Adspec 时，该位置为 0。路径上任何支持 RSVP/IS 但不支持 GS 的路由器将其置为 1，以通知接收者 Adspec 是无效的，服务不能得到保证。
GS 通用参数头/值	可选项。就接收者所希望的 GS 保留而言，如果选择了其中的任何一个，都会忽略 DGP 段所给定的相应值。

说明：GS 为合法数据分组提供一种保证的带宽级、恒定的端到端延迟范围和无排队丢失的服务。主要用于严格实时传输需求的场合。数据流传输路径上的每个路由器，通过分配一个带宽 R 和数据流可能占用的缓冲区空间 B 为一个特定的数据流提供保证服务。采用漏桶流量模型控制排队延迟。参数采用 RSVP 中的传输规范 Tspec 和保留规范 Rspec。

- Tspec 参数：（与 Path 消息中携带的发送者的传输特性 Sender_Tspec 一致）

  数据流峰值速率 P（bytes/s）

  桶深 b（bytes）（类似于节点的缓冲空间）

  漏桶速率 r（bytes/s）

  最小管理单元m（bytes）[3]

  最大数据报长度 M（bytes）

- Rspec 参数：

带宽 R（bytes/s）

时隙 S（ms）

通过上述的Tspec参数和Rspec参数，可以计算出数据流的排队延迟。理想模型中，数据流排队延迟限制的定义为：$Q_d=b/R$（$R\geqslant r$）

路由器引入两个偏差量C、D建立近似模型：$Q_d=b/R+C/R+D$（$R\geqslant r$）

考虑到 Tspec 和 Rspec 各个参数对排队延迟的影响，更为精确的端到端排队延迟限制的定义为：

$Q_d\approx(b-M)(p-R)/R(p-r)+(M+C_{tot})/R+D_{tot}$（$p>R\geqslant r$）

$Q_d\approx(M+C_{tot})/R+D_{tot}$（$R\geqslant p\geqslant r$）

其中C_{tot}和D_{tot}分别是端到端各个路由器C和D的总和（这两个参数存在于Adspec的GS字段中）。这样，就为数据流提供一个端到端的专线带宽。为了便于路由器计算和保留足够的本地资源，引入统计量：C_{sum}和D_{sum}（存在于Adspec的GS字段中）。分别表示该路径上各路由器自上次更新后的C和D之和。这样，计算出端到端的排队延迟限制，如果用户选择GS服务（Path消息中的Adspec字段中的GS字段有效），则数据流的端到端排队延迟将严格限制在该范围中。

（3）被控负载服务段 CLS

CLS 所含字段如下表：

表7 被控负载服务段 CLS

字段	说明
CLS 中止位	标志位。发送者创建 Adspec 时，该位设置为 0。路径上任何支持 RSVP/IS 但不支持 CLS 的路由器可将其置为 1，以通知接收者 Adspec 无效，服务不能得到保证。
CLS 通用参数头/值	可选项。就接收者所希望的 CLS 保留而言，如果选择了其中的任何一个，都会忽略 DGP 段所给定的相应值。

Path 消息沿着数据传输的路径，逐跳传播。路径上每一支持 RSVP 的节点，将根据 Path 消息所携带的信息，在本地建立相应的 Path 状态，并对 Path 消息进行修改后，传递给下一跳。如果 Path 消息出现错误，发现错误的节点将向其上一跳发送 PathErr 消息，并停止向下一跳发送 Path 消息。

2.4 Resv 消息

接收者可以从接收到的 Path 消息提取 Sender_Tspec（参看 3.3 节）所含的传输特性参数和 Adspec 所含的 QoS 参数。利用这些参数可建立起接收者保留规范 Rspec。

Rspec 由如下参数组成：

- 带宽 R：根据 Sender_Tspec 参数计算而成。如果得到的 R 值大于 Adspec 中的路径带宽值，则 R 值必须相应地减小。R 值将保存在各个路由器上。
- 时隙 S：表示端到端延迟限制与应用所需端到端延迟的差值，初始为 0。通过设置 S 值，将为各个路由器在确定局部保留上提供更多的弹性，提高端到端保留的成功率。

利用 Rspec 可以创建一个 Resv 消息。一个 Resv 消息包含如下的内容：

- 保留模式：可以是 Fixed Filter、Wildcard Filter 或 Shared Explicit 保留模式中的任何一类。
- 过滤器规范(Filterspec)：用来标识发送者和格式，该格式和相应 Path 消息中的 Sender_Template 相同。
- 数据流规范(Flowspec)：由 Rspec 和 Tspec 组成。通常，将 Tspec 设置成接收到的 Sender_Tspec（存在于 Path 消息中）。
- 保留确认对象(ResvConf)：可选项，含有接收者的 IP 地址，用于指示接收这个保留请求的节点。ResvConf 消息在分布树上向上传播，最终达到该接收者，表明端到端保留的成功。

Resv 消息按所存储的路径状态被送往 Phop 指出的上游路由器。在该路由器上，Resv 消息可以和到达同一接口的其它 Resv 消息合并，再传送到下一个上游路由器。

2.5 RSVP 协议的特点

RSVP 协议具有下列特性：

（1）RSVP 可以在点对点传播和多点组播的网络通信应用中进行预留资源的申请，它可以动态调节资源的分配以满足多点组播中组内成员的动态改变，和路由状态改变的特殊需求。

（2）RSVP 只为单向的数据流申请资源。这就意味着，在双向对话中，需为不同方向的数据流，分别进行资源预留。

（3）RSVP 是面向接收端的，由数据流的接收端进行资源申请并负责维护该数据流所申请的资源。

（4）RSVP 在路由器和主机端维持“软”状态，解决了组群内成员的动态改变和路由的动态改变所带来的问题。（参看 2.6 节）

（5）RSVP 并不是一种路由协议，它依赖于目前或将来出现的路由协议。

（6）RSVP 本身并不处理流量控制和策略控制的参数，而仅把它们送往流量控制和策略控制模块。

（7）RSVP 提供几种资源预留的模式供选择以适应不同的应用需求。（参看 4.2.1 小节）

（8）RSVP 对不支持它的路由器提供透明的操作。

（9）RSVP 支持 IPv4 和 IPv6。

2.6 RSVP 协议中的“软状态”

“软状态”是 RSVP 用来管理和维护路由器和主机中的资源保留状态和路径状态的机制。具体而言，“软状态”由保留信息和路径信息定期创建和刷新，如果在清除时间到期前没有合适的刷新信息到达，那么“软状态”信息就将被删除。该状态还可以由“卸载”命令直接删除。每次过期情况发生或是状态改变后，RSVP 将重新建立它的资源保留状态和路径状态并将它发往后继节点。

当路由发生变化时，下一个路径会建立新路由的路径状态。原来路径上的信息会由于超时而被删除。

当某一个发送端服务质量的请求发生变化时，主机端仅仅送出一个修改过的路径信息和保留信息。然后这个信息会沿着网络传送到所有相关节点，更新这些节点中的相应信息，而那些无用的状态或是被显式删除或是由于超时而删除。

在一个稳定中的系统下，状态刷新信息沿着传送路径一个接一个进行合并。当节点接收到的刷新信息和节点中存有的信息不同时，就将节点中存有的信息更新，如果更新同时需要改变后续节点的信息时，刷新信息必须立即产生并向下传送，如此，这些刷新信息就从端到端无延时地传送下去。当刷新信息对该节点后续节点的刷新信息不产生影响时，该刷新信息的传送就可以终止了。这可以减少组群改变时刷新信息对网络带来的负载，对庞大的组群尤其有用。

3. RSVP消息传播流程

RSVP的基本原理是发送者在发送数据前，首先发送Path消息与接收者建立一个传输路径。Path消息含有对即将传送的数据流的描述和其它控制信息。沿途的各个路由器都记录这些信息（更新或新建），并为它做好保留资源的准备。接收者收到Path消息后，则回送一个Resv消息进行应答。Resv消息沿相反的路径传送给数据发送者，途经各个路由器时，对Path消息指定的QoS级别给予确认。预留完成后，发送者和接收者之间通过这条路径传输数据流，沿途的各个路由器为该数据流保留资源，按协商的QoS级提供转发服务。[3]

3.1 RSVP 协议的数据流

“对话”是 RSVP 协议中一个重要的概念，RSVP 的每一次预留都针对某一特定对话。“对话”是在特定的目标地址和传送协议上的数据流。RSVP 对每次对话的处理都是独立的。一次 RSVP 对话可以由一个三元组（目的地址，协议号，[协议端口]）来表示。

- ✧ **目的地址**：也就是 IP 目的地址，既可以是多点组播地址，也可以是点对点传播地址。
- ✧ **协议号**： 即 IP 协议号。
- ✧ **协议端口**：可选参数，为目的地址的端口号（例如一些协议上所定义的多路 复用点），它可以直接利用 UDP 或 TCP 的端口，或是其他协议中类似的域、甚至是应用层信息来定义。目前只能支持 TCP/UDP 的协议端口。由于不同的广播地址一般对应不同的对话连接，所以针对广播地址，一般不必包含协议端口。而单一接收端有多个点

对点传播对话同时存在的可能，这时必须使用协议端口来区分不同对话。

典型的组播 RSVP 对话如图 1 所示。采用分布环境下的多点组播传送，任一发送端 Si 送出的数据，经过分布环境下的组播路由，传递给每一个接收端 Rj。而分布环境下的点对点传播，只有一个接收端，每一个发送端是网上唯一标识的主机，而接收端（同一目的 IP 地址）可以通过协议端口的区分功能（不同端口号），接收多个发送端的数据。这就意味着，对于点对点传播，一个接收端地址（同一目的 IP）可以被多个发送端所共享，此时需通过不同的目的端口号区分不同对话。RSVP 可以处理这类“多对一”的传送模式。

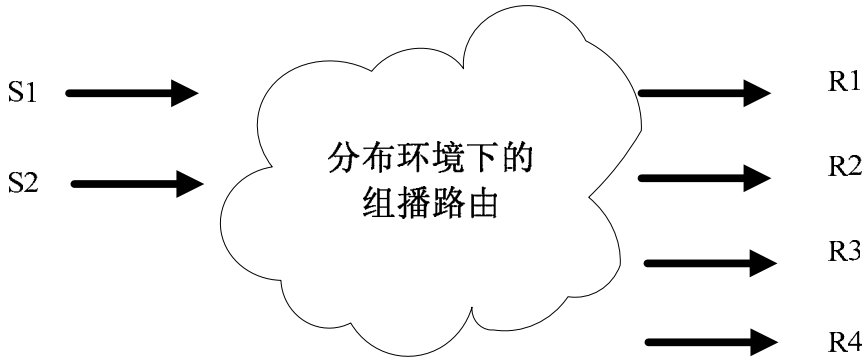

图 1 典型的组播 RSVP 对话

3.2 RSVP 消息传播流程

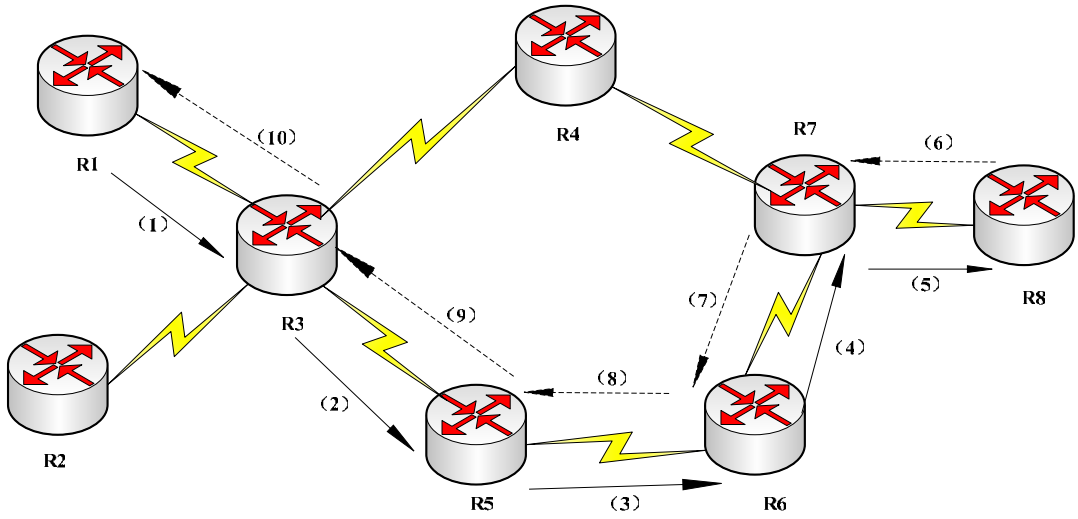

图 2 RSVP 消息的传播流程

如图 2 所示的网络，利用 RSVP 消息在 R1 与 R8 之间建立资源预留，设已根据路由协议找到数据传输路由为 R1—R3—R5—R6—R7—R8。Path 消息和 Resv 消息的传输过程如下表所示。

表8 RSVP 消息传播流程说明

序号	消息类型	消息方向	描述
1	Path	R1 到 R3	R1 向 R3 发送 Path 消息，R3 截获该消息后，检查其有效性。如果发现错误，则卸下 Path 消息，并发送 PathErr 消息给 R1。若验证通过，进行下一步。
2	Path	R3 到 R5	R3 向 R5 发送一个 Path 消息，R5 对其进行同样的验证。
3	Path	R5 到 R6	R5 向 R6 发送一个 Path 消息，R6 对其进行同样的验证。
4	Path	R6 到 R7	R6 向 R7 发送一个 Path 消息，R7 对其进行同样的验证。
5	Path	R7 到 R8	R7 向 R8 发送一个 Path 消息，R8 对其进行同样的验证。
6	Resv	R8 到 R7	R8 作为尾端，向 R7 发送一个 Resv 消息，指明需要预留的模式、QoS 参数及资源等。R7 对 Resv 消息进行验证。若出现错误，则发送 ResvErr 消息给 R8，并停止向上游发送 Resv 消息。若准许预留且有足够资源，则进行下一步。
7	Resv	R7 到 R6	R7 向 R6 发送 Resv 消息，R6 对其进行同样验证。
8	Resv	R6 到 R5	R6 向 R5 发送 Resv 消息，R5 对其进行同样验证。
9	Resv	R5 到 R3	R5 向 R3 发送 Resv 消息，R3 对其进行同样验证。
10	Resv	R3 到 R1	R3 向 R1 发送 Resv 消息，R1 对其进行同样验证，若通过，则在 R1 和 R8 之间成功建立了预留。

4. RSVP中节点状态分析

4.1 节点对 RSVP 消息的处理

支持RSVP的节点在接收到RSVP消息后的处理流程如下图。

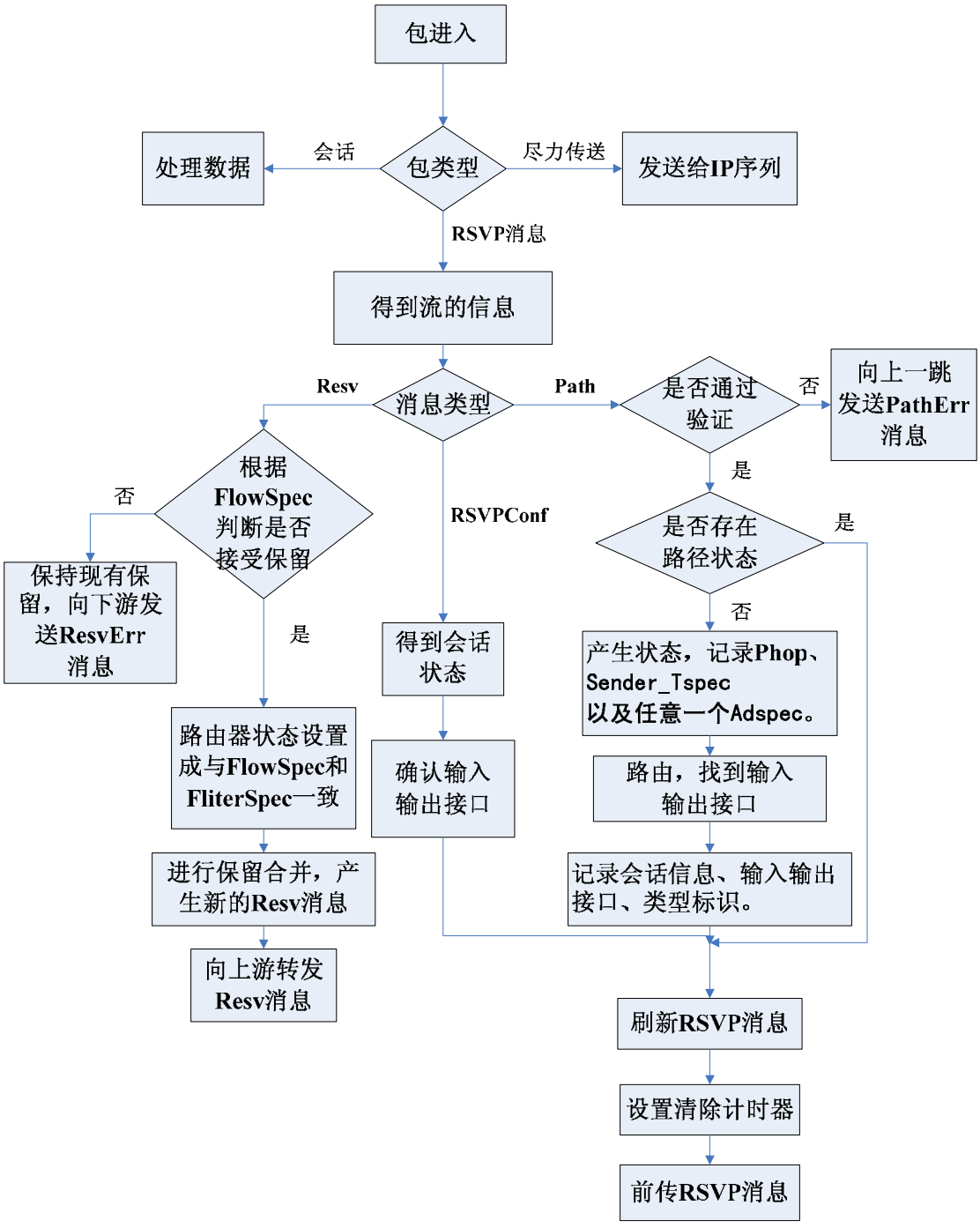

图 3 节点对 RSVP 消息的处理流程

4.2 合并流

由于支持RSVP的节点，在处理流合并时，根据不同的资源预留风格（Resv消息中指明），所做的操作有所不同，故先对RSVP中的资源预留风格进行说明。

4.2.1 RSVP协议中的资源预留风格

资源预留请求选项中定义了如何对待对话中不同发送端的不同的资源保留请求。RSVP支持两种资源预留形式：独占资源预留和共享资源预留。支持两种发送端选择方式：可以是分别列出所涉及的发送端，或是利用通配符来选择对话中的所有发送端。在前一种情况下，筛选规格说明（Resv消息中的Filter_spec字段）必须完全符合其中的每个发送端，而在后一种情况下则不需要考虑规格说明。不同的预留风格，将导致RSVP传输路径上的各个节点，对于流合并的操作有所不同。

资源预留风格的分类如表9所示。

表9 资源保留的属性和类型

发送端选择	保留方式	
	独占	共享
分别列出	Fix Filter Style （FF 类型）	Share-Explicit （SE 类型）
通配附	（未定义）	WildCard-Filter （WF 类型）

（1） WF 类型

WF 类型具有的特性是：共享式的保留模式、通配符式的发送端选择模式。因此，WF 类型的保留模式创建单一的保留资源供所有的“上传”数据流使用。这个保留资源被认为是一个共享的资源，它的大小是所有接收端申请资源的最大值，而和发送端的数量无关。

WF 类型的保留将通知所有的发送端，当出现新的发送端时，这个值将自动提供给它。

WF 类型的资源保留申请记作：WF（*{Q}）。其中 * 代表通配附，Q 代表流规格。

（2） FF 类型

FF 类型具有的特性是：独占的保留模式、分别列出的发送端选择模式。因此，FF 类型为每个发送端创建唯一的资源预留，而不和其他发送端共享。

一个基本 FF 类型的资源保留申请记作：FF（S{Q}）。其中 S 代表所选的发送端，Q 代表流规格。这两个符号代表了一个流描述。

RSVP 允许多个 FF 类型的资源保留同时进行，记为：FF（S1{Q1}，S2{Q2}，……）。这样一个对话的资源保留的总数量为 Q1、Q2、Q3……的总和。

（3） SE 类型

SE 类型具有的特性是：共享的保留模式、分别列出的发送端选择模式。因此，SE 类型的保留模式创建单一的保留资源供所有的“上传”数据流使用。但和 WF 不同的是，它以分别列出的方式来选择可以使用该保留资源的的发送端。

我们把一个具有 Q 的流描述和 S1、S2、S3……发送端的 SE 类型的资源保留记作：SE((S1，

S2，S3，……）{Q}）

共享的保留方式，WF 或 SE 类型，适合于那些基于广播并且广播的发送端不太可能同时发送数据的应用程序。例如分组音频就是一个非常适合这种类型的应用，由于很少有人在会谈时同时说话，因此每个接收端可以为发送端申请一个 WF 或 SE 类型的双倍带宽保留（允许某些超载情况）。而为每个发送端创建唯一保留资源的 FF 类型，则十分适合于视频信号。

4.2.2 不同资源预留风格的流合并

如图 4 所示，为RSVP传输路径上任意支持RSVP的路由器。数据流的方向如图所示，Path消息的传输方向与数据流的传输方向一致，Resv消息的传输方向与数据流的方向相反。A、B、B' 为数据流的发送者，C、D、D' 为数据流的接收者，a、b为接入端口，c、d为输出端口。

对于同一会话，若存在来自不同下一跳的多个预留申请，且这些申请具有相同的过滤参数（filter spec），那么 RSVP 将进行“流合并”，从而在收到该申请的接口，为这一会话保留唯一的一个预留。具体的“流合并”过程，根据不同的预留风格而不同。

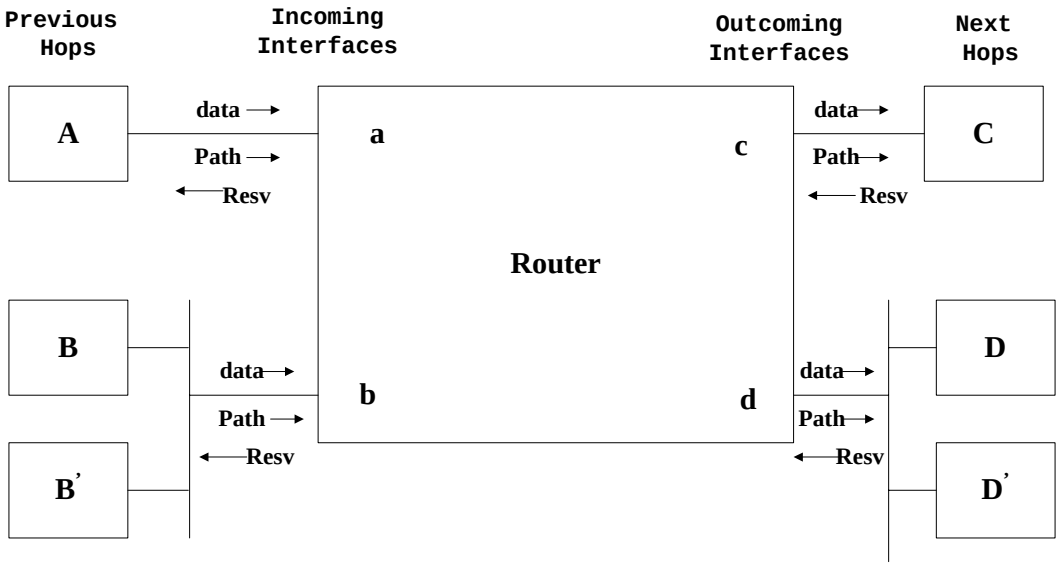

图 4 支持 RSVP 的节点示意图

（1）WF 风格

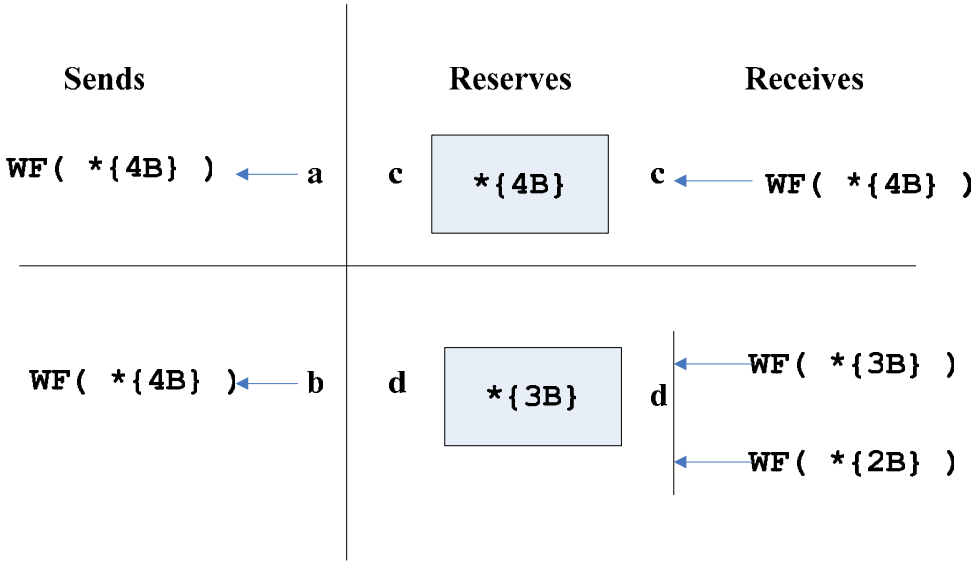

图 5　WF 风格的预留

采用 WF 风格的预留，如图 5 所示。在接口 d 收到来自接收者的对于同一会话的资源预留申请 3B 和 2B，进行流合并后，在接口 d 的预留标记为 3B。此后，接口 c 和 d 的预留资源进行合并，标记为 4B，然后再转发给上游。最后，在接口 a 和 b 的资源预留均标记为 4B。

（2）FF 风格

采用 FF 风格的预留，如图 6 所示。在接口 d 收到对于同一会话的资源预留申请：FF(S1{3B}, S3{B})和 FF(S1{B})。由于是独占的资源预留方式，故在 d 接口标记的资源预留为 S1{3B}、S3{B}，即仅将同一发送者的资源预留合并，不同发送者的资源预留申请不合并。随后，c、d 接口的资源预留申请进行合并，同样仅对相同发送者进行合并，最终得到 a 接口的 S1{4B}和 b 接口的 S2{5B}、S3{B}的资源预留标记。

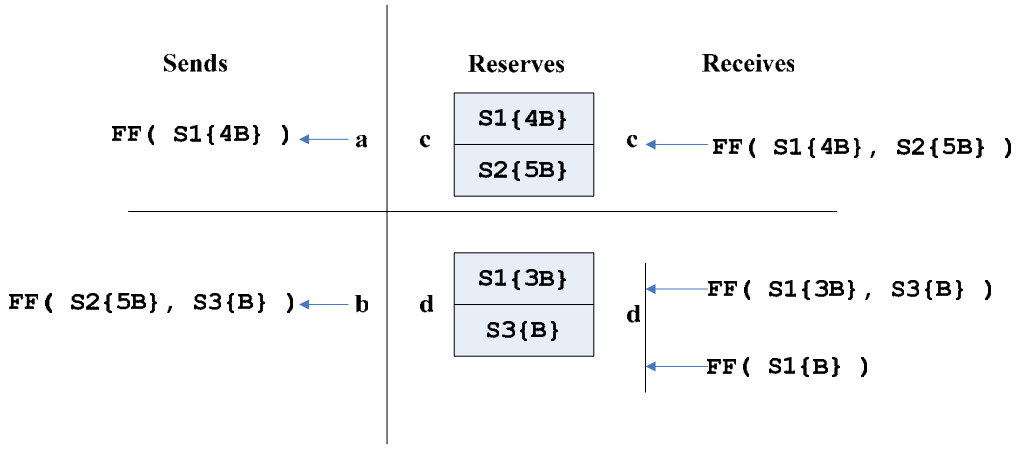

图 6　FF 风格的预留

（3）SE 风格

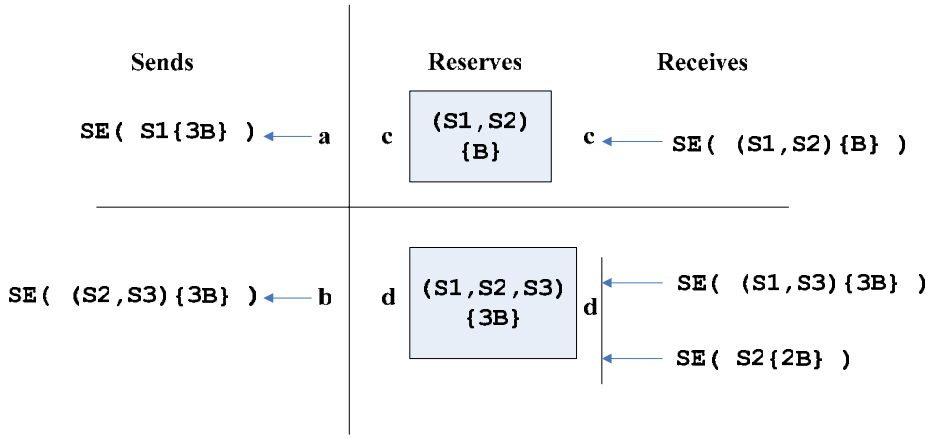

图 7　SE 风格的预留

采用 SE 风格的预留，如图十所示。接口 d 收到对于同一会话的资源预留申请：SE((S1,S3){3B})和 SE((S2){2B})。因为是共享式的资源预留，对其进行合并时，不同发送者的资源预留取最大值，d 接口的资源预留最后标记为：(S1,S2，S3){3B}。此后，c、d 接口的资源预留进行合并，取对不同发送者的资源预留申请中的最大值，最后得到 a 口的 SE((S1){3B})和 b 口的 SE((S2,S3){3B})。

5．基于RSVP的QoS实现

5.1 QoS 的实现原理

RSVP协议的一个重要特点是：其本身并不处理流量控制和策略控制的参数，而仅把它们送往流量控制和策略控制模块。一个典型的RSVP通信模型如图8。

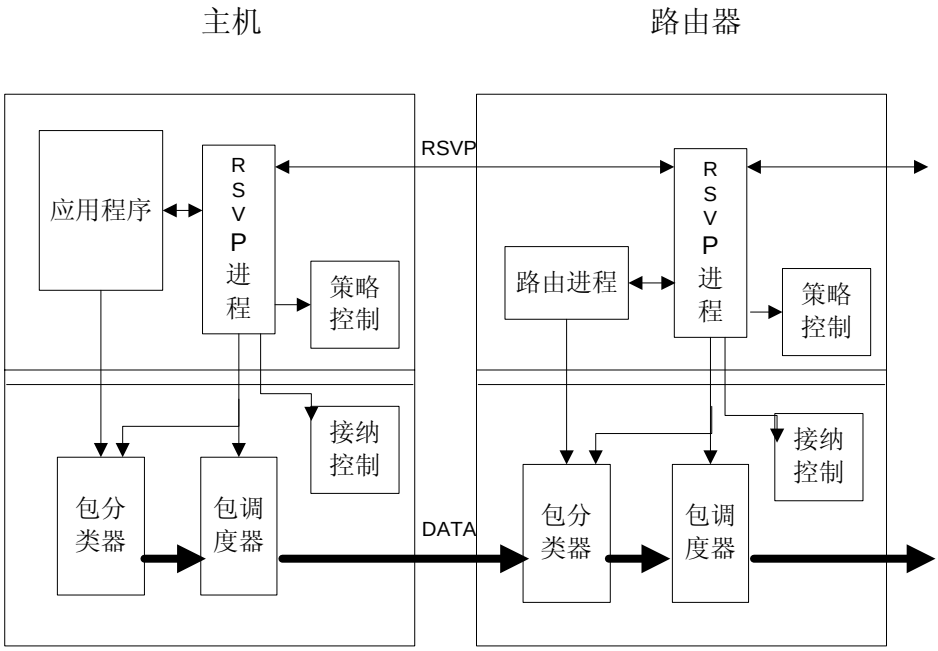

图 8　主机和路由器中的 RSVP 通信模型

QoS是由一组特定的称为流量控制的机制来实现的。这些机制包含了1、包分类器。2、

接纳控制。3、包调度器或是其他与链路层相关的机制。其中包分类器决定每个包的QoS的分类。包调度器或其他和链路层相关的机制用来获取所请求的QoS。

在资源申请建立的过程中，RSVP请求被传送到两个本地模块：接纳控制模块和策略控制模块。由接纳控制模块决定该节点是否有足够的资源可以满足该RSVP请求。策略控制机制决定用户是否有权限申请这类服务。如果通过了这两个模块的检测，那么RSVP请求的QoS参数就会输入到包分类器中，再由链路层接口（如包调度器）来获得申请的服务质量。如果任一模块的检测没有通过，那么提出该RSVP请求的应用程序进程将会得到一个错误的返回。由此可见，接纳控制模块必须采取一定的算法，来判断本地是否有足够的资源，以满足RSVP中申请的资源（如Resv消息中Flowspec字段中的Rspec参数）。

其中，接入控制算法对于QoS的实现具有重要意义。考虑到网络资源的有限性，这里采用基于参数的简单和接纳控制算法（simple sum）。此算法常常基于预知或简化的流模型。如把所有流的到达都视为泊松过程并假设它们服从独立同分布。据此获得描述每个流的流量特性的参数，然后根据这些参数估计所需资源(如最坏情况下所需带宽的峰值)的总和。当新的流发出接纳请求时，就利用这些估值加上新的流的资源请求。如果所有流请求的资源之和不超过链路的能力，那么该流可以被接纳，反之，该流不被接纳。

其接纳条件的数学表达式为

$V + r < U$

V表示已预留的带宽，r表示新的请求带宽，U表示链路带宽。

5.2 QoS 的动态调整

为了获得满足QoS要求的媒体流，服务器需要为每一个流预约资源。如果系统有足够的资源，一个流就可以分到想要的QoS要求的资源，其数量可能比仅仅满足可接受的QoS要求的资源要大得多。随着流数目的增加，系统资源可能会耗尽。此时，一个新的流请求不能得到足够的资源来满足它的QoS请求。

当对请求的流可得到的资源不足时，可以在接入控制中对QoS进行折衷，将每一个流的QoS请求分成两套：最佳值和可接受值。如果现有资源不能满足新增流的QoS要求，则在已有的流间进行调整：服务器通过与用户的再协商，或按预先达成的QoS降级许可，从已有的流中回收一部分资源，从而有足够的资源来允许新的流请求。

但新的流的接入不应该使现存的流的QoS变得不可接受。因此，服务器应该适当的降低现存的流的QoS。另外，服务器处理动态的QoS适应还包括：当服务器有多余的资源，而流的QoS没有达到希望的QoS需求时，服务器应该能够分配更多的资源给已有流来提高它的QoS；当一个流结束时，一些资源会归还给系统，服务器可以将这些资源分配给其他的流，提高它们的QoS。这样可以使系统的资源利用最大化，并且可以允许更多的流接入。

可采用如图9的动态接入控制算法。算法流程中的完全预约与最小预约分别对应于媒体请求QoS最佳值和可接受值。如果能够得到的资源可以实现QoS的最佳值，显然可以接入。如果可以得到的资源能至少满足QoS的可接受值，也可以接入。如果无法达到上述要求，就需要通过再协商，试图通过QoS降级，得到资源。如果仍然无法得到资源，接入失败。

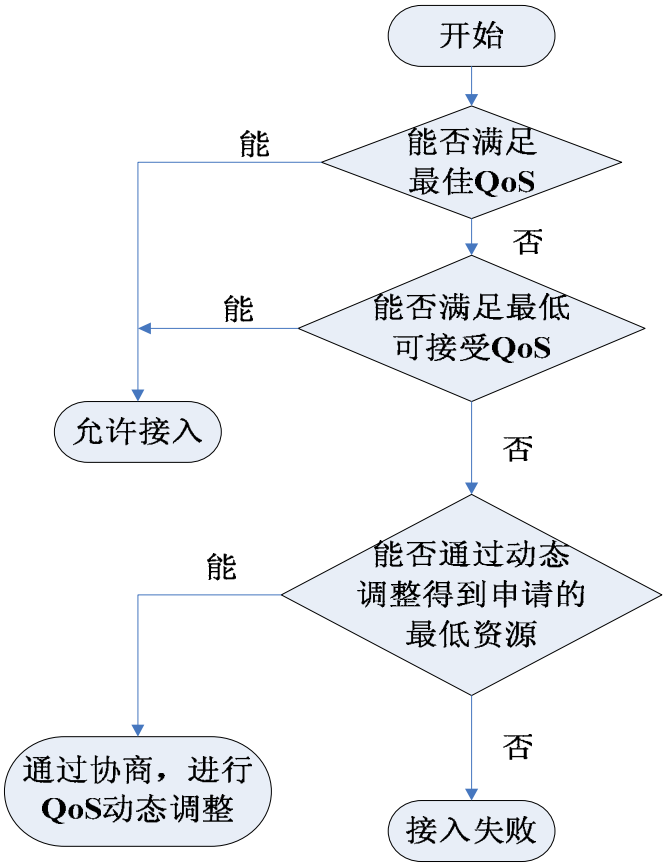

图 9 动态接入控制算法

具体而言，QoS动态调整步骤如下：

（1）检查流请求是否可以被接入。每一个流有一个最小的流QoS要求。如果可能得到的流QoS满足最小流QoS要求，请求能被允许接入；否则，请求就会被拒绝，现有的流不需要降级。

（2）收回某些已有流的一部分资源，降低它们的QoS。那些采用最佳QoS值的流被优先选择。对每一个这样的流，分配资源来符合它可接受的QoS要求。在回收了流的资源后，计算可得到的QoS。如果仍不满足新增流的最低要求，可以选择另一个流来收回它的资源。继续这个过程，直到符合最小要求的QoS。用这种方式，降低最小数目的流的QoS来允许接入新的流。

（3）新的请求被接入后，将可用的资源分配给新的媒体请求。同时QoS管理器为相关的媒体流产生一个相关的调度。

（4）文件I/O管理程序和代理服务器根据这个变化更新动态调度，完成QoS降级的过程。

当流结束时，为其预约的资源返还给系统，这样就有更多的资源可用。此时检查在系统中是否有一个流需要更多的资源来达到想要的QoS要求。如果存在，则更新调度，直到没有更多的资源可用或所有的流都分配了理想的资源，达到最佳的QoS要求。

这样，通过QoS的动态调整，可以在不降低现有QoS级别的基础上，允许更多请求的接入，从而在确保QoS的基础上，提高网络资源利用率。

6. 总结

RSVP 工作在传输层，在 IP 上层，是一个控制协议。RSVP 的组成元素有发送者、接收者和主机或路由器。发送者负责让接收者知道数据将要发送，以及需要什么样的 QoS；接收者负责发送一个通知到主机或路由器，以准备接收即将到来的数据；主机或路由器负责留出所

有合适的资源。RSVP 只负责参数的传递， 并不实际操作这些参数，QoS 的实现，还需要本地的一些策略控制及流量控制模块协同工作。

RSVP 领域的发展是非常迅速的，但目前并没有在任何一种网络上得到证实，它的应用只是局限在测试的小 Intranet 网络上。另外，对于其可扩展性运用，也有待进一步研究。

参考文献

[1] Campbell, A.T., De Meer, Kounavis H., et al. A Review of Programmable Networks[J], ACM Computer Communications Review, April 1999

[2] Braden R, Zhang L, Berson S, Herzog S, Jamin S. Resource Reservation Protocol(RSVP) – Version 1 Functional Specification[S]. RFC 2205, IETF. September 1997

[3] 蔡皖东. RSVP协议的多媒体综合服务机制研究[R]. 南京:第十届中国计算机学会网络与数据通信学术会议论文, 1998。

[4] 段晓宇. QoS的研究和IntServ/RSVP的实现[D]. 北京：北京邮电大学硕士学位论文，2005

[5] 张世滢、周建中、陶琳. 基于RSVP实时应用的QoS性能研究[J]. 微机发展 Oct 2005 Vol.15 No.10

[6] 刘轶，肖凯平，刘楚达，等. QoS接纳算法性能分析与比较[J]. 计算机工程 Oct 2005 Vol.31 No.20

Achieving QoS Control via Resource Reserve Protocol

Li Chunyang

Beijing University of Posts and Telecommunications, (100876)

Abstract

Information Times is coming with various data service. So the quality of service provided by Internet is more and more important. Resource Reserve Protocol is just designed to ensure the quality of service in the net which is used in Internet. This paper includes a introduction about the message used in RSVP，the principle of QoS implement as well as its functional specification. Besides, the act of message flowing and node transacting is specified. Especially, a method of dynamic QoS adjusting is put forward that the net resource can be used more effectively.

Keywords: *Resource Reserve Protocol, Quality of Service, Resource Reserve*