

区分服务路由器软件结构研究

Study on the Software Architecture of DiffServ Router

李振武 王 健 白英彩

(上海交通大学计算机系 上海 200030)

Abstract With the transformation of Internet from research network to commercial network and the emergence of all kinds of applications, which have different QoS needs, Internet is strongly required to provide QoS service. This paper compares the two solutions: Intserv and Diffserv and makes a brief introduction on Diffserv. Then based on these, software architecture of Diffserv router is outlined and some critical function modules are discussed in detail. At the end of the paper we discuss the future work which should be done on Diffserv router.

Keywords QoS, Diffserv, DSCP, Classifier, Meter, Marker, Scheduler

1. 引言

Internet 已经延伸到了世界的每一个角落,成为了各种应用的承载工具。在这些应用中,近年不断发展的多媒体应用如视频会议、视频点播(VOD)、VOIP 等受到了广泛的关注,这些应用和普通的应用不同,它们对网络带宽、延迟、抖动、丢包率等有一定的要求,因此,网络必须对多媒体应用提供一定的服务质量(QoS)保证。

当前的 Internet 采用的是 best-effort(尽力服务)模型。在这个模型中,网络平等地处理所有数据包,尽它的最大努力转发数据包,其首要目标是实现对链路的统计复用以达到高网络利用率,路由器作为核心设备,它是无状态的,也就是说它不保存关于客户机连接的属性,如连接时间、连接使用的带宽等,连接的状态信息存放于实现 TCP/IP 协议栈的端点主机。网络资源的分配依赖于主机和路由器的拥塞控制机制,路由器维持一个 drop-tail 队列,当队列满的时候就丢掉到来的报文。端点主机 TCP 检测到丢失了报文,认为是拥塞信号因此降低它的发送速率,所有的主机获得的带宽之和等价于拥塞点的带宽,因此用户获得的服务是不可预测的,网络用户没有带宽和延迟的保证。

为了解决网络对 QoS 的支持问题,Internet 工程任务组(IETF)已经开发了一些新的模型,最初提出的是集成服务模型(IntServ),它通过资源预留协议是在路由路径上从端主机应用向路由器运载 QoS 请求,QoS 请求用来为应用的数据流创建、维持和拆除资源预留状态。为了达到这个目的,文[1]提出了 RSVP 协议。RSVP 是一个接收方发起资源预留的协议,适用于单播和组播。RSVP 的思想是数据接收方在数据传输的相反路径上发起一个 QoS 请求,请求包括一组参数说明 QoS 需求,称为流说明符。沿途的任一路由器如果没有足够的资源接收这个请求,它的许可控制模块就会拒绝请求,拒绝报文将发回请求发起方。如果所有的路由器都接收请求,RSVP 负责维持并刷新由路由器建立的 QoS 状态,这个在路由器中保存的状态成为软状态,因为一旦不被动态刷新状态就会过期,这个特性能动态适应路由的改变。

但是集成服务模型具有如下问题:

(1)路由器需要维持每流状态,在骨干网络上流的数量非常大的情况下几乎是不可能的,因此不可扩展。

(2)RSVP 协议非常复杂,路由器需要耗费大量资源解释

RSVP 请求并且处理许可控制以建立状态。

(3)如果在请求路径上有一个路由器没有实现 RSVP,那么就不能实现资源预留。

为了解决集成服务模型的可扩展性和鲁棒性问题,文[2]提出了区分服务(Differentiated Service,简称 DiffServ)模型。本文首先介绍区分服务模型的要点,然后详细分析了实现该模型的网络元素-支持区分服务的路由器结构和相关元素,最后指出了今后一段时间关于区分服务的研究方向。

2. 区分服务模型简介

区分服务(DiffServ)模型结合了“尽力服务”模型和“集成服务”模型的优点,其首要目标是保留现有 Internet 的统计复用特性,同时使用一种灵活的、可扩展的机制提供大范围的 QoS 服务,第二个目标是使得网络资源的使用可以计费。区分服务的提出基于这样的观察:很多实时应用对网络的 QoS 要求并不是很严格。集成服务模型中,网络本身支持非常细粒度的 QoS 要求,而区分服务仅在边界网络支持这样的细粒度 QoS,并把它映射到一些包的类别。在网络内部,区分服务仅仅对不同类别的报文区别对待。这个机制简化了内部网络的实现并使得这个机制是可扩展的。

区分服务模型的基本思想是在网络的入口处为分组标记一个区分服务编码码点(DSCP),码点用于指示分组在网络转发路径的中间结点上应该被处理的方式。这样对每个分组进行的复杂处理被推到了网络边缘,核心网的主要任务只是根据分组首部的码点对其采用相应的转发措施。在区分服务模型中由于不需要网络中间结点管理每个正在工作的流状态,因此具有良好的可扩展性。每个用户同 ISP 有一个“服务层协议”(SLA),它确定了用户能得到的服务级别,同时也规定了用户的流量应符合 SLA,用户为了得到 SLA 必须付相应的费用。在 SLA 规定范围内的流量将得到 ISP 的良好保证,超出的部分被视为“机会流量”,不能得到服务保证。

网络的入口点一般是 ISP 的入口边界路由器(Ingress Router),网络转发路径的中间节点是 ISP 的内部路由器(Interior Router),因此入口边界路由器采用一个称为“流量整形”的机制实现对数据包的分类、监控、标记。内部路由器采用简单的机制转发报文并根据不同的 DSCP 值给予不同的服务。如果通信跨越两个域,则在第一个域的出口路由器(Egress Router)处,报文需要根据两个域之间签订的 SLA 重

李振武 博士研究生,研究领域为分布式系统,高性能网络,网络存储。王 健 硕士,中兴通信上海所工程师,研究领域为图像处理,网络管理。白英彩 教授,博士生导师,主要研究领域为分布式系统,网络管理,网络安全。

新标记 DSCP 值。

以上分析可以看出,区分服务模型具有以下优点:

(1)由于网络中间节点不需要维持每流状态,因此是可以扩展的。

(2)不需要复杂的资源预留信令协议,边界节点分类、监控、标记报文,内部节点只是根据不同的 DSCP 值实行简单转发。

(3)不要求报文传输路径上的所有域都支持区分服务,不支持区分服务的域中的路由器可以忽略 DSCP 字段,因此鲁棒性好。

(4)在区分服务结构中,一旦一个数据包通过边界路由器进入网络的核心部分,内部路由器以“聚集”对待它们,对于网络资源是高度的统计复用,这保留了当前“尽力服务”模型的高效的资源利用率的特点。

因此,区分服务是解决网络 QoS 的一个有效的方法,目前这方面的研究是一个热点。

3. 区分服务路由器软件结构

网络支持区分服务,关键的是需要有支持区分服务的路由器,包括入口路由器、内部路由器和出口路由器。在区分服务体系结构中,这三种路由器的功能不一样,入口路由器负责报文的分类、标记、监控,内部路由器根据报文的 DSCP 值给予不同的服务级别,出口路由器根据两个域之间的 SLA 重新标记报文。下面详细讨论这三种路由器的软件结构及功能要素。

3.1 入口边界路由器软件结构

入口边界路由器的软件结构如图 1 所示,一个没有标记的报文进入入口路由器后,经过如下流程:

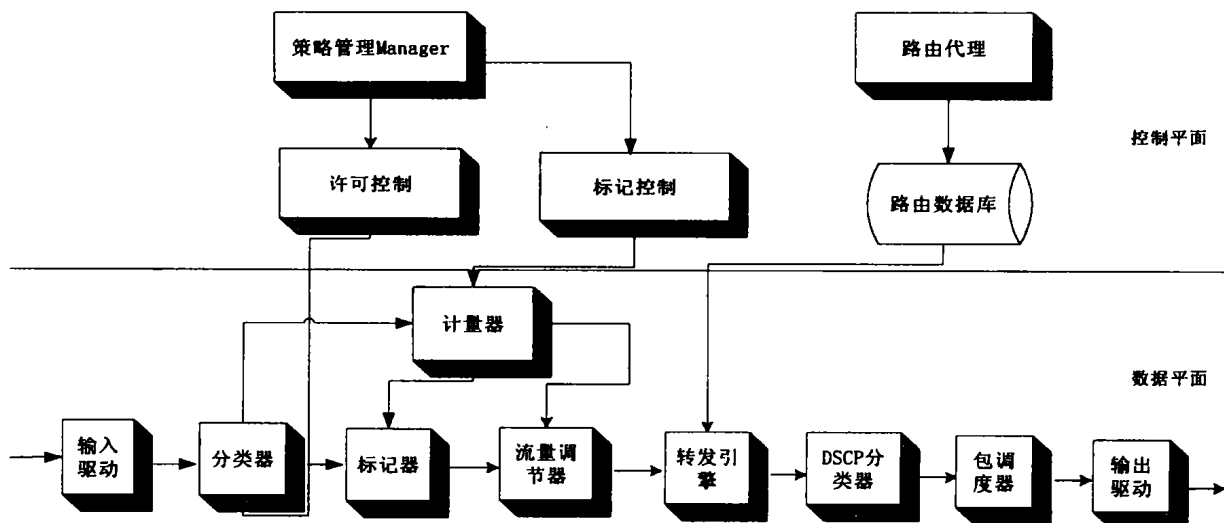


图 1 入口边界路由器软件结构

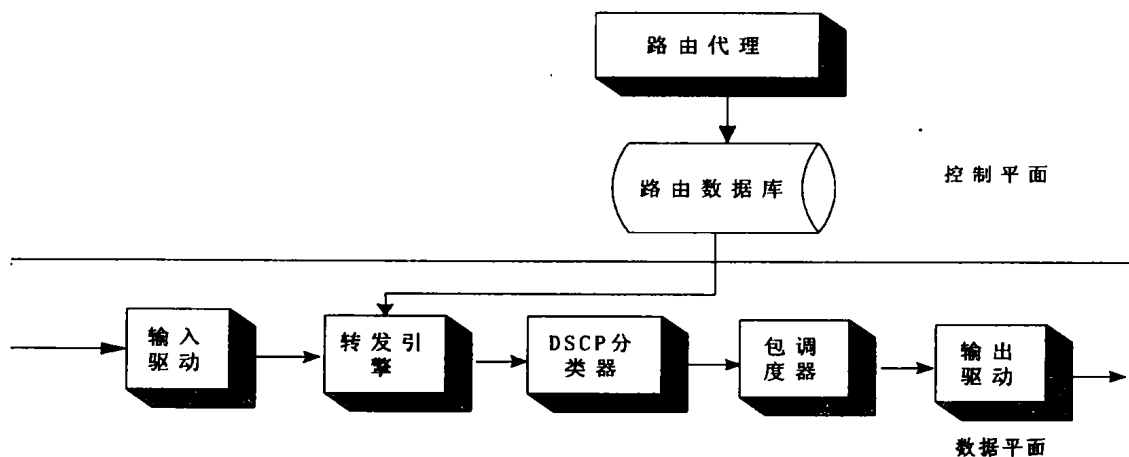


图 2 内部路由器软件结构

(1)输入驱动得到报文,交给分类器,分类器根据策略管理 Manager 的信息把输入报文流分成若干个子集。已经有两种分类器被定义:行为聚集分类器,仅根据 DSCP 进行分组分类;多域(Multi-Field, MF)分类器,根据分组报头的多个字段进行分类,如源 IP 地址、目的 IP 地址、源端口、目的端口、输入网络接口和协议字段等。

(2)分类器把分过类的报文同时交给计量器(Meter)和标记器(Marker)。计量器根据该报文的分类向策略管理 manag-

er 查询该分类报文的流量调约定(TCA),TCA 规定相关的流量特征描述以及应用于分类器所选流的计量、标记、丢弃和整形的规则。计量器将根据测量所得的状态信息传递至标记器和流量调节器以对该分组触发特定的 DSCP 标记和调节行动。这个分组可能符合或不符合相应的流量特征描述,流量调节器根据计量器的状态信息进行调节,最后报文交给转发引擎。

(3)转发引擎通过路由代理查询路由表,得到输出接口

号,报文交给 DSCP 分类器。

(4)DSCP 分类器根据 DSCP 的值把报文放入不同的调度队列。

(5)输出驱动得到并输出该报文。

3.2 内部路由器软件结构

数据流在入口边界路由器分类且获得相应 DSCP 值后进入内部网络则汇成了“聚集”,相同 DSCP 值的报文属于同一个“聚集”。内部路由器执行简单的转发机制,对不同 DSCP 值的报文实行不同的服务级别,如图 2 所示,流程如下:

(1)输入驱动得到报文,交给转发引擎。

(2)转发引擎通过路由代理查询路由表,得到输出接口号,报文交给 DSCP 分类器。

(3)分类器根据 DSCP 的值把报文放入不同的调度队列。

(4)输出驱动得到并输出该报文。

3.3 出口边界路由器软件结构

出口边界路由器除了要和内部路由器一样执行基于 IP 的路由查找和根据不同 DSCP 值执行不同调度级别外,还需要对进入下一个区分服务域(DS Domain)的报文根据两个域之间签订的 SLA 修改报文的 DSCP 值,如图 3 所示,流程如下:

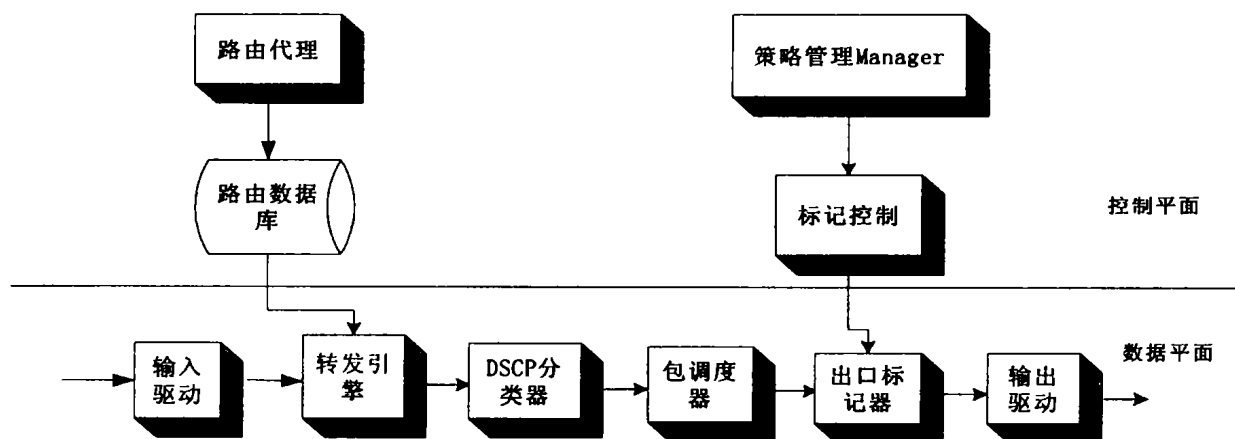


图 3 出口边界路由器软件结构

(1)输入驱动得到报文,交给转发引擎。

(2)转发引擎通过路由代理查询路由表,得到输出接口号,报文交给 DSCP 分类器。

(3)DSCP 分类器根据 DSCP 的值把报文放入不同的调度队列。

(4)被调度的报文交给出口标记器,出口标记器向策略管理 Manager 发出该报文的标记请求。

(5)策略管理 Manager 的标记控制模块查询策略数据库,根据和下个 DS 域签订的 SLA 发回标记应答。

(6)标记器根据标记应答对该报文的 DSCP 字段重新标记。

(7)报文交给输出驱动输出。

4. 相关功能元素讨论

4.1 分类器(Classifier)

分类器的功能是把未分类的输入报文根据一定的规则分为多个流,可以有如下多种类型的分类器:

(1)BA(行为聚集)分类:仅仅根据 DSCP 的值进行分类。

(2)MF(多域)分类:根据报文的多个字段分类,根据分组报头的多个字段进行分类,如源 IP 地址、目的 IP 地址、源端口、目的端口、输入网络接口和协议字段等。也可以根据 MAC 地址,VLAN 标记和其它标志。

(3)可以根据基于用户定义的过滤规则分类,比如说位的模式。一个位的模式包括三个部分,起始位、偏移量(offset)和掩码。

4.2 计量器(Meter)和标记器(Marker)

计量器的作用是对分过类的报文特征比如说流量进行测量,得到的测量结果和该分类报文 SLA 规定的特征比较,得到符合或不符合 SLA 的判断,该结果返回给标记器和流量调节器。标记器根据一定的标记算法对报文进行标记,符合

SLA 的报文标记为 IN,不符合的部分可以标记为 OUT,也可以以一定的概率标记为 OUT,IN 和 OUT 报文具有不同的调度级别。计量器和标记器一起实现对报文的监控,一个理想的计量和标记算法应该具有如下特性:

- 通用性。应该适合各种性质的流量,无论是 TCP、UDP、突发的、长期的,还是交互式流量。

- 流量评估的长期特性。应该在一个较长的时间内对流量进行评估而不是短期行为。

可以有如下几个算法用来实现计量标记模块:

(1)令牌桶计量标记 假设有一个深度为 D(以字节计量)的桶,令牌以 R(字节/秒)填充,桶中的令牌不超过 D。报文通过令牌桶标记器的时候,如果桶中的令牌大于等于报文长度,则报文标记为 IN,否则标记为 OUT。在 DS 网络中,如果令牌填充速率等于流目标速率 R_t ,可以实现对流量的监控。但这个算法有两个问题:

- 桶深 D 值难于设置。D 的作用是允许桶有“正记忆”功能,也就是说如果流发送速率低于目标速率,令牌可以在桶中聚集,这样可以允许一定的突发流量,如何设置 D 关系到允许流量的突发程度。在不知道流的行为的情况下,无法设置 D 值。

- 过量流量没有记录。假设一个流以 $R(R > R_t)$ 速率发送了一段时间,耗尽桶中的令牌,在这之后 $(R - R_t)/R_t$ 百分比的报文被标记为 OUT。但是超过 R_t 值多少却没有记录,无法反映到将来的标记算法中,不符合理想算法的流量评估的长期特性。

(2)加权计量标记 加权计量标记用如下公式计算流的平均速率:

$$R = (1 - w) * R + w * R_{\text{inst}}$$

其中 R 是估计的速率, R_{inst} 是瞬间速率,可以通过报文尺寸除以包到达间隔得到。如果 R 大于等于 R_t 则以概率 $P = (R -$

Rt)/Rt 标记报文为 OUT, 否则标记报文为 IN。

这个算法有如下问题:

- 端主机的报文可能突发, 造成报文“背靠背”到达, 也就是说报文之间的间隔接近于 0, 计算 R_{inst} 困难。

- 对“历史”遗忘偏向于发送快的流, 流速率越快对“历史”遗忘也越快; 相反, 如果一个流在一段时间内没有数据发送, 无论时间多长, 所使用的发送速率仍然是“静默期”之前记录的数据。

(3) TSW (Time Sliding Window) 计量标记^[3] TSW 这个算法的优点是能够考虑到数据突发和没有数据发送的情况, 网上的绝大多数数据流是基于 TCP 的并且数据的突发性是 TCP 的主要特征, 因此这个算法能够真实地计量数据源的真实发送速率, 并恰当地监控。

TSW 算法维持如下变量:

- Window_length, 单位是时间单元, 系统将只记忆在这个时间长度内收到的数据包。

- Current_rate, 这是当每个包到达时系统计算的速率。

- T_pre, 上一个包到达的时间。

Window_length 是系统配置的参数, Current_rate 和 T_pre 在每个包到达时都要修改, 系统只记忆 Window_length 时间范围内的系统数据, 并且随着时间衰减。算法描述如下: 初始化:

```
Window_length = 常量;
Current_rate = 策略的控制速率;
T_pre = 0;
当每一个包到达时, TSW 算法如下动作:
Bytes_in_TSW = Current_rate * Window_length;
New_Bytes = Bytes_in_TSW + packet_size;
Current_rate = New_Bytes / (now - T_pre + Window_length);
T_pre = now;
如果 Current_rate 小于 Rt 则报文标记为 IN, 否则以概率  $P = (Current\_rate - Rt) / Rt$  标记报文为 OUT
```

从这个算法我们可以看出:

(1) 假如数据突发, 由于该算法记忆了 Window_length 时间内的数据, 则该窗口内的数据量也会相应增加, 即 New_Bytes 增加, Current_rate 值也增加。

(2) 假如一段时间内没有数据流量, 则记忆在 Window_length 内的字节数目会降低, 同样 New_Bytes 降低, 进而导致 Current_rate 值降低, 因此这个计量标记算法能对流进行较准确的计量和标记。

4.3 队列调度器(Scheduler)

在区分服务结构中, 不同 DSCP 值的报文获得不同的服务级别, 在同一个 DSCP 值的报文中, IN 和 OUT 报文也需要区别对待, IN 标记报文的优先权大于 OUT 报文, 网络拥塞的时候首先丢掉 DSCP 优先权低的 OUT 报文。

我们对每个 DSCP 值的报文设置一个队列, 在每个队列之间采用 WFQ 算法调度, 每个队列有和其 DSCP 值对应的权值, 获得和其权值成比例的带宽, 由此可以对不同的 DSCP 值报文区别对待。在每个队列内部, 由于有 IN 和 OUT 两种不同类型的报文, IN 报文的优先权大于 OUT 报文, 因此在一个队列内部也需要实现服务区分, 我们采用的是 RIO 算法^[4]。RIO 算法是 RED 算法的变种, RED 算法通过检测路由器的队列长度, 一旦发现队列长度超过一定限制就认为有拥塞的可能, 通过丢包或者设置 IP 报头的 ECN 标志位通知源端降低发送速度, 源端减少拥塞窗口大小, 以减少进入网络的数据流量, 避免了更多的包丢失。对于一个连接来说, RED 可以避免丢失同一个连接的连续数据包, 提高了连接的吞吐量。执行 RED 算法的路由器不象 FIFO 算法那样等队列满了以后才丢包, 而是当队列长度超过一定限度时就以一定概率丢包, 整个算法包括两部分: 对队列长度的监控和如何丢包。如

果采用 RED 算法, 队列的状态将被监视。如果队列接近它的容量, 包就被丢失, 注意到包的选择是完全随机的。这可以避免队列完全满而 Drop Tail 进而发生全局同步现象。RIO 和 RED 算法的不同之处是在计算丢弃概率的时候配置了两套参数, 一套用于 IN 包, 一套用于 OUT 包。每个核心路由器对 IN 包维持一个虚拟队列, 对 IN 和 OUT 包维持一个物理队列。对于每个队列使用 RED 算法。当一个包到达的时候, RIO 按照如下算法工作:

(1) 首先检查报文是 IN 包还是 OUT 包:
如果是 IN 包, 计算 IN 包的平均队列长度 avg_in :
 $avg_in = avg_in * w_q + instqueue(in) * (1 - w_q)$
如果是 OUT 包, 计算 OUT 包的平均队列长度 avg_total :
 $avg_total = avg_total * w_q + instqueue(total) * (1 - w_q)$
(2) 比较平均队列长度和阈值的关系:
如果是 IN 包, avg_in 小于 $minTh_IN$ 的时候, 报文不丢弃; avg_in 大于 $minTh_IN$ 小于 $maxTh_IN$ 的时候, 报文按照如下概率丢弃:
 $P_{drop_in} = (avg_in - minTh_IN) / (maxTh_IN - minTh_IN) * P_{max_in}$ 当 avg_in 大于 $maxTh_IN$ 的时候, 报文以概率 1 丢弃。
如果是 OUT 包, avg_total 小于 $minTh_OUT$ 的时候, 报文不丢弃; avg_total 大于 $minTh_OUT$ 小于 $maxTh_OUT$ 的时候, 报文按照如下概率丢弃:
 $P_{drop_out} = (avg_total - minTh_OUT) / (maxTh_OUT - minTh_OUT) * P_{max_out}$ 当 avg_total 大于 $maxTh_OUT$ 的时候, 报文以概率 1 丢弃。

通过以上两套参数实现 RED 算法实现对不同标记报文的区分对待。

因此, 如图 4 所示, 当报文经过 DSCP 分类器放入对应 DSCP 调度队列的时候, 首先需要经过 RIO 算法决定是否丢弃该报文, 经过 RIO 算法判定保留的报文最终进入 DSCP 队列, WFQ 调度器按照每个队列的权值调度报文交给驱动转发。

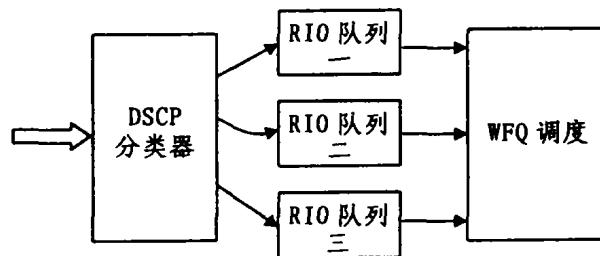


图4 报文调度

结束语 区分服务路由器是区分服务网络的主要构成元素, 根据其在区分服务网络体系中的不同位置及功能划分, 有入口边界路由器、内部路由器、出口边界路由器三种类型。由于区分服务网络的复杂性在网络边缘, 因此入口边界路由器的软件结构复杂, 内部路由器的软件结构简单, 保证了网络的可扩展性。目前对这三种路由器的软件结构研究是一个热点, 主要集中在如何保证带宽分配的公平性。产生公平性问题的主要原因是各个微流的不同特性, 比如说 RTT、报文尺寸、是否实行末端拥塞控制、目标速率等等, 导致各个微流竞争带宽能力的不同。因此需要在边界路由器和内部路由器中加入一个可扩展的智能报文标记、报文丢弃模块动态跟踪流的各种特性以避免带宽的不公平分配。

参考文献

- 1 Zhang L, et al. RSVP: A new resource reservation protocol. IEEE Network, 1993, 7(5): 8~18
- 2 Blake S, Black D, et al. An Architecture of Differentiated Services. RFC2475, Dec. 1998
- 3 FANG Wenjia. Differentiated services: architecture, mechanisms and an evaluation. 66-69, Dissertation for doctor degree, Computer Science of Princeton University, 2000
- 4 Clark D D, Fang W. Explicit allocation of best effort packet delivery service. IEEE/ACM Transaction on Networking, 1998, 6(4)