

# 流量控制和流量整型的比较

彭涤 (深圳市宝安区信息中心 518000)

摘要: 本文介绍了在 IP 网中进行 QoS 流量限制的两种方法——流量整型和流量控制的区别和联系以及表现形式, 同时介绍了这两种方法使用的令牌桶算法的机制, 帮助正确选择和运用合适的速率限制方法。

关键词: QoS 速率限制 流量控制 流量整型 令牌桶 Shaping Policing

## 1 引言

在实现宽带 IP 城域网时, 实施 QoS 非常重要, 在对流量进行速率限制时, 常用有两种方法: 流量控制(Traffic Policing)和流量整型(Traffic Shaping), 两种概念容易混淆, 使用不当不仅不能达到预期的目的, 严重的还会影响网络的正常运行。本文就流量控制和流量整型功能上的异同作了比较, 从而根据具体情况做出正确选择。

流量整型需要队列因而要有足够的内存来存放延迟的数据而流量控制不用, 因此, 当你实现流量整型时要考虑有足够的内存。另外, 流量整型还需要有调度的功能来传输延迟的数据包。

## 2 两种方法的区别

表 1 列出了流量整型和流量控制的比较, 帮助选择最佳的限制速率的方法。

## 3 令牌桶原理

流量整型和流量控制在使用令牌桶算法时的主要区别在于令牌是如何补充的, 这一节说明其差别。

简单说来, 流量整型和流量控制都使用令牌桶, 令牌桶本身没有丢弃或优先级的策略, 让我们看看令牌桶是如何工作的:

- (1) 令牌按一定速率放入令牌桶。
- (2) 每个令牌允许源发送一定数量的数据

位至网络中。

(3) 为了发送一个数据包, 流量测量工具必须从令牌桶中取走与该数据包同样数量的令牌。

(4) 如果令牌桶中没有足够数量的令牌来发送数据包, 此数据包要么等到令牌桶中有足够数量的令牌(流量整型的情况)要么丢弃或再标记该数据包(流量控制的情况)。

	流量整型 Shaping	流量控制 Policing
目的	对超过速率的流量缓存并放入输出队列	不缓存, 对超过速率的流量丢弃或标记
令牌刷新率	令牌在每个时间间隔开始时增加(需要配置最小时间间隔)	令牌连续增加, 基于公式: $1 / \text{committed information rate}$
令牌值	位/秒 (bits per second)	字节数 (bytes).
配置选项	<ul style="list-style-type: none"><li>·shape 在命令行中用于实现基于类的流量整型</li><li>·frame-relay traffic-shape 用于实现帧中继流量整型 Frame Relay Traffic Shaping (FRTS).</li><li>·traffic-shape 用于在以太网等线路上实现流量整型 Generic Traffic Shaping (GTS).</li></ul>	<ul style="list-style-type: none"><li>·police 在模块化 QoS 命令行中实现基于类的流量控制</li><li>·rate-limit 实现 CAR (committed access rate).</li></ul>
是否适合输入流量	不适合	适合
是否适合输出流量	适合	适合
突发流量	使数据流平缓流出, 使用类似漏斗的算法。	对超出的流量过滤
优点	尽量不丢包, 超过速率的数据被缓存, 只有当持续超速并且缓存满的情况下才会丢包, 这时依靠上层应用程序重传丢弃的数据	通过丢包来控制速率, 防止队列带来的延迟
缺点	会带来队列的延迟, 特别是队列很深的情况下, 延迟较严重	对超过速率的数据包丢弃会减小 TCP 的 Windows 滑动窗口大小, 从而减小传输速率, 这样会降低传输效率, 特别是基于 TCP 的数据流
能否对数据包进行再标记	不能	能(与传统的 CAR 特性配合)

表 1

(5) 令牌桶有指定的容量,如果令牌桶已满,新来的令牌被丢弃,不能给数据包使用,这样,在任何时间内,最大的突发数据不能超过令牌桶的容量,令牌桶允许突发流量,但是受限制。

有了令牌桶的概念以后,我们看看流量整形和流量控制分别是如何添加令牌到令牌桶中去的。

流量整形按时间间隔(timed intervals)来增加令牌,单位是位/秒(bps),公式如下:

$$Tc = Bc / CIR \text{ (in seconds)}$$

在此公式中, Bc是指顺从突发速率(burst committed), CIR 是指平均速率或承诺信息速率(committed information rate)。Tc 的值定义了当你发送 Bc 个数据位在 CIR 速率允许下的最小时间间隔。

Tc 值的范围是 10 ms 到 125 ms 之间。在 Cisco 7500 系列路由器上的分布式流量整形(distributed traffic shaping)中,最小的 Tc 值是 4 ms。路由器内部利用 Bc 和 CIR 计算 Tc 值,如果 Bc/CIR 小于 125 ms,则使用计算出来的 Tc 值,如果 Bc/CIR 大于或等于 125 ms,则由 Cisco IOS®确定一个使数据流稳定的更小值。使用 show traffic-shape 命令来检验路由器是否使用你配置的 Tc 值。

另外,流量整形分两种方式来堆积令牌,第一种方式是当传输速率低于 CIR 时,第二种方式是当配置了扩展突发量 Be(excess burst)时,在第一个 Tc 期间, Cisco IOS 向令牌桶添加 Bc + Be 个令牌,这样在一个 Tc 期间能传输的最大数据位是 Bc + Be,超过的数据位将丢弃。

与流量整形相比较,基于类的流量控制连续增加令牌到令牌桶,令牌的增加速率按如下公式计算:

$$\text{(数据包间隔时间} < t-t1 > * \text{流量控制速率)} / 8 \text{ 位每字节}$$

如果前一个数据包到达的时间是 t1,现在的时间是 t,则令牌桶增加的令牌(t-t1)\*流量控制速率/8,注意的是流量控制的突发数据值的单位是数据位,所以上述公式要除以 8 转换成字节。

下面是一个流量控制的示例,控制速率为 8000bps,正常的突发数据量为 1000 字节,配置如下:

```
Router(config)# policy-map police-setting
Router(config-pmap)# class access-match
Router(config-pmap-c)# police 8000 1000
conform-action transmit exceed-action drop
```

令牌桶开始装满 1000 字节的令牌,如果有一个 450 字节的包到达,数据包是允许发送(conform)的,因为令牌桶中有足够的令牌,则此数据包发送出去了,这时令牌桶也相应地减少了令牌,只有 1000-450=550 个令牌了,如果下一个数据包在 0.25 秒之后到达,根据公式则有 250 字节的令牌增加到令牌桶中:

$$(0.25 * 8000) / 8 = 250$$

这时,令牌桶中还有 550+250=800 字节的令牌,如果这个数据包是 900 字节,超过令牌所允许,则此数据包丢弃,令牌桶中的令牌不减少。

#### 4 流量整形的形式

Cisco IOS 支持如下几种形式的流量整形

- (1) 一般流量整形(Generic Traffic Shaping),
- (2) 帧中继流量整形(Frame Relay Traffic Shaping),
- (3) 基于类的流量整形(Class-Based Shaping)和分布式基于类的流量整形(Distributed Class-Based Shaping)。

所有流量整形的方法都是类似的,尽管在命令行中语法有些不同,并且队列的类型和整形的数据流类型有不同之处,建议使用基于类的流量整形,这种配置方法是使用 IOS 中的 MQC(Modular QoS CLI)来配置的,具有统一的配置方法便于使用和维护。

下图显示了 IOS 如何对数据流进行分类并利用令牌桶对数据的速率进行限制后将其放入不同的队列中去:

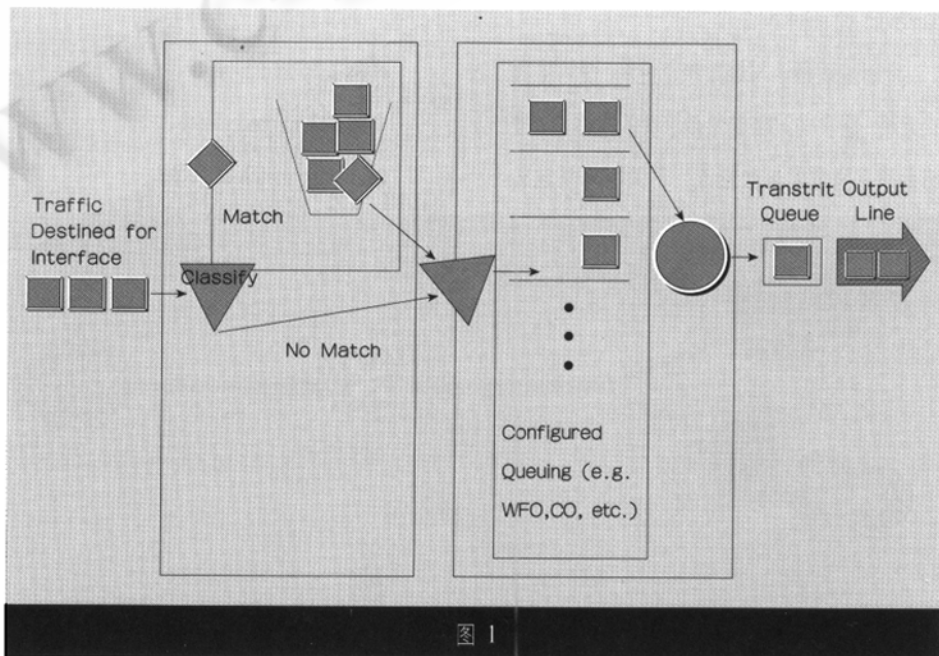


图 1

# Comparing Traffic Policing and Traffic Shaping

## 5 流量控制的形式

Cisco IOS 支持如下几种流量控制方法:

- (1) 流量速率控制(Committed Access Rate)。
- (2) 基于类的流量控制(Class-Based Policing)。

这两种方法在数据流分类和对超出流量限制时的操作上有所不同,建议使用基于类的流量控制,该方法使用MQC,更加灵活和方便。

使用 `police` 关键字定义要限速的数据,如果超过速率限制,将会立即执行相应的动作,没有缓存等选项,不象 `shape` 命令。

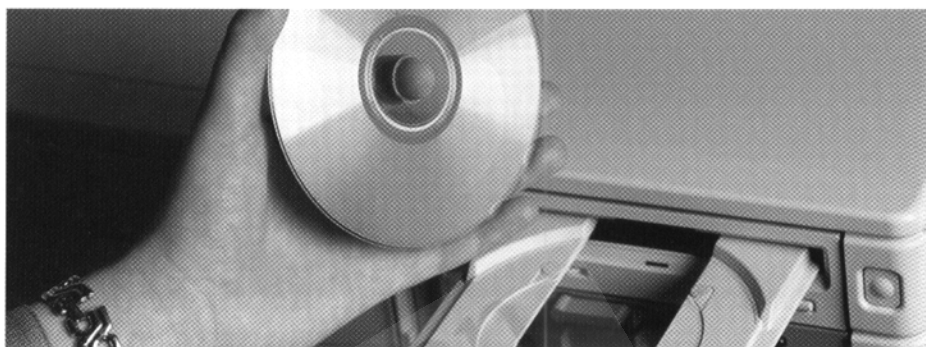
另外,利用流量控制,令牌桶来测量数据流是否超过预定的速率,并且可以用来设定IP的优先级或设置IP的DSCP(Differentiated Services Code Point)值。

下图显示了对流量控制的过程,通过流量控制,使得图中一部分数据流以较小的速率访问 Internet。

## 6 最大与最小带宽控制

利用 `shape` 和 `police` 命令可用来限制输出的最大传输速率,两者都不能保证在拥塞时的最小带宽,要实现最小带宽的保证,使用 `bandwidth` 或 `priority` 命令。

同时实现最小与最大带宽控制,使用层次式策略,层次式策略提供多层次的策略:父策略用来定义或汇聚数据流,子策略用来定义



QoS或进一步细化数据流,对于逻辑接口,如子接口或隧道接口,需要用分层的策略,父策略用来作数据流速率控制而子策略用于放数据于不同的队列中,流量控制的特性用于限制输出数据的速率而队列的特性用于将超过速率的数据置于队列中。

下面仅是一个示例,为了说明 `police` 和 `shape` 的差别,而不是优化的配置,在此配置中, `police` 命令发送子策略基于包大小和令牌桶中令牌数的数据包,其结果是语音的数据(VoIP)和IP并不能保证速率,因为 `police` 的特性覆盖了 `priority` 特性。

但是如果使用 `shape` 命令,所用的特性都可保证,换句话说,如果输出的数据流超过了预定的速率,则VoIP和IP按照规定的速率传输,而其他数据流则会丢包。

```
class-map match-all IP
  match ip precedence 3
class-map match-all VoIP
```

```
match ip precedence 5
```

```
policy-map child
```

```
class VoIP
  priority 128
```

```
class IP
  priority 1000
```

```
policy-map parent
```

```
class class-default
```

```
police 3300000 103000 103000 conform-action transmit exceed-action drop
```

```
service-policy child
```

## 7 小结

本文通过对流量控制和流量整形的比较,同时介绍了令牌桶算法在两者应用的异同,有了这些知识,便能根据需求正确选择合适的方法加以应用。 ■

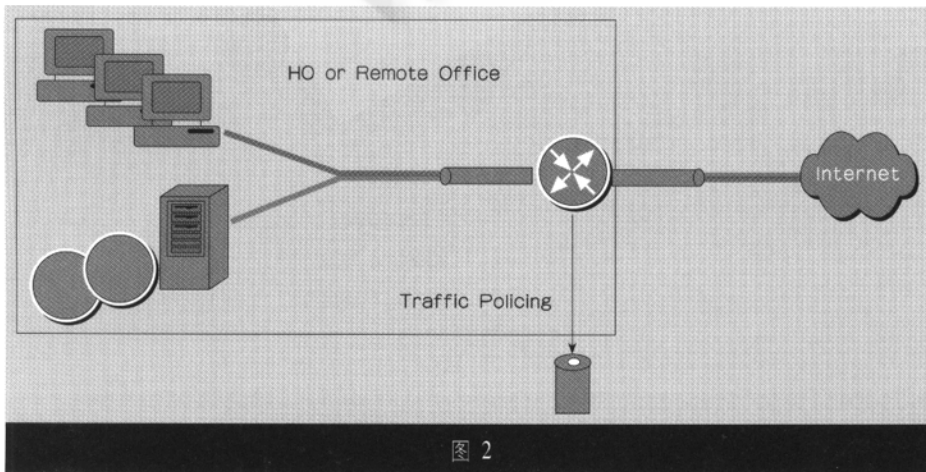


图 2



### 参考文献

- 1 RFC791: "Internet Protocol Specification," J.Postel, 1981.
- 2 RFC896: "Congestion Control in IP/TCP Internetworks", J.Nagle, 1984.
- 3 RFC1349: "Type of Service in the Internet Protocol Suite," Almquist.