

在有混合单播、组播和语音流量的Cisco 12000系列互联网路由器体系结构上的WRED和MDRR配置示例

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[用于测试的优先等级](#)

[IP 包的着色](#)

[预期结果](#)

[网络设置](#)

[3 端口 GbE 引擎2 排队实施](#)

[Cisco IOS 软件的引擎 2 WRED 算法](#)

[QoS 配置](#)

[Rx CoS](#)

[Tx QoS](#)

[接口映射](#)

[无丢弃启动](#)

[四个每路 151Mb 的数据流](#)

[四个每路 160Mb 的数据流](#)

[四个每路 167Mb 的数据流](#)

[四个每路 191Mb 的数据流](#)

[四个每路 244Mb 的数据流](#)

[卸除所有 QoS](#)

[四个每路 153Mb 的数据流](#)

[四个每路 158Mb 的数据流](#)

[添加入口负载](#)

[12-RP#show interfaces g 6/0](#)

[更改流的大小](#)

[用 10 端口千兆以太网引擎 4 线路卡验证](#)

[相关信息](#)

简介

本文解释如何配置加权随机早期检测(WRED)的—Cisco 12000系列线卡，描述在[RFC 2309](#)，在多

业务环境。

[先决条件](#)

[要求](#)

本文档的读者应具备以下方面的知识：

- [了解和配置 Cisco 12000 系列互联网路由器上的 MDRR 和 WRED](#)
- [如何解读show controller frfab的输出|在Cisco 12000系列互联网路由器体系结构的tofab queue命令](#)
- [服务类型在互联网协议套件，优先\(RFC-1349\)](#)
- [加权 随机 早期 检测](#)

[使用的组件](#)

本文档中的信息基于以下软件和硬件版本：

- 支持Cisco 12000SERIES互联网路由器的任何Cisco IOS软件版本。通常这些是12.0S和12.0ST版本。
- 所有Cisco 12000平台由本文包括。这些包括12008，12012，12016，12404，12406，12410和12416。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

[规则](#)

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

[背景信息](#)

Cisco 12000系列是其中最普遍的平台用于构件高带宽IP核心网络。此平台提供不包括可能性配置服务质量(QoS)。

因为它是越来越普通混合不同种类的IP数据流(例如基于IP的语音- VoIP -，并且组播)在同一网络，优先级的需求和一种受控的下降的行为，在许多情况下，变得非常重要，和是成败之间的差异，当启动一新的服务例如VoIP时。

不同种类的网络要求的IP数据流是超出本文的范围之外。本文着重实验室测试进行查找是可适用的在不同线路卡，包括Cisco 12000系列，3波尔特千兆以太网的配置(3波尔特GbE)线卡。这些测验结果显示3波尔特GbE引擎2线卡为介入语音、数据和组播数据流的混合网络环境是非常合适的。它也证明那配置QoS做在堵塞的网络的一个实际区别。

[用于测试的优先等级](#)

分配到不同的类的优先值需要是相同的在整个网络。您需要确定一项通用的策略。

中集集团	优先级	流量
邪恶的流量		所有未确定净流量(网外呼叫)
在网---在网	1	在SP网络的流量(网内)内坚持
互联网服务提供商服务	2	ISP流量, SMTP, POP, FTP, DNS, Telnet, SSH, www, https
SME (中小企业)	3	企业用户, 金牌服务服务
实时, 无声	4	TV, 实时游戏
语音	5	RTP VoIP流量
网络控制消息	6-7	边界网关协议(BGP)和其他控制消息

IP 包的着色

如果QoS将实现在网络的核心, 前提条件是服务提供商在优先值的完全控制中在网络传输的所有IP信息包。要执行此的唯一方法是标记所有信息包, 当他们进入网络时, 进行没有差异他们是否来自客户/最终用户侧或互联网。标记或着色在核心不应该完成。

预期结果

与此设计的目标是有在类的真的WRED行为0-3。这意味着我们希望有我们开始下降优先0数据包在拥塞时的情况。在那以后, 我们应该也开始也下降优先1, 如果拥塞继续, 然后优先2和3。这在下面图表所有描述。

您不应该有低延时可能为语音数据包和丢包语音和组播数据流的。

网络设置

要测试和评估配置, 我们与从Agilant的一个信息包生成器一起使用了Cisco 12410。Cisco 12000路由器运行根据Cisco IOS软件版本12.0(21)S1的工程版本。

3 端口 GbE 引擎2 排队实施

引擎2卡通常有八个fromfab队列和一低延时队列和八个tofab队列每目的地地址槽。也有一个分开的tofab组播队列。在3波尔特GbE卡, 只有每个物理端口一个fromfab队列。在测验中, 应用的配置指定更多队列。结果显示3波尔特GbE卡接受此配置, 并且队列自动地被映射对是可行的队列。

Cisco IOS 软件的引擎 2 WRED 算法

当配置最低和最大队列深度值时, 您必须充分地了解用于WRED的算法在引擎2线卡。代码对最小值不关心配置; 反而, 它使用其自己的公式(根据配置最大值)设置最小值。

公式:

最小值=最大值- (不生成一种负结果)的大功率2

用于此测验的值导致以下最小值被编程对Application-specific integrated circuit (ASIC) :

优先级	已配置的最小值	配置的最大值	大功率2	在ASIC的最小值
0	50	5000	4096	5000-4096=904
1	60	6000	4096	6000-4096=1904
2	70	7000	4096	7000-4096=2904
3	80	8000	4096	8000-4096=3904

使用此公式计算最小值意味着您能最终获得不正确的信息包处理行为，如果不考虑到此，当配置您的WRED参数时。这在以下示例显示：

优先级	已配置的最小值	配置的最大值	大功率2	在ASIC的最小值
0	50	150	128	150-128=22
1	75	225	128	225-128=97
2	100	300	256	300-256=44
3	125	375	256	375-256=119

这意味着，即使值配置开始首先丢弃根据规则0，然后1，2和最后3(见上)，当值写入对ASIC时，您实际上开始下降优先0，然后优先2，然后优先1和最后优先3。没有办法看到哪些值在引擎2卡的ASIC配置。如果运用在引擎3卡的配置，显示在配置里的值将是实时值(重新计算的最小值)。

QoS 配置

欲了解更详细的信息关于QoS配置，请读[了解和配置MDRR和WRED在Cisco 12000SERIES互联网路由器](#)。

Rx CoS

```
rx-cos-slot 2 B2-Table
rx-cos-slot 3 B2-Table
rx-cos-slot 6 B2-Table
```

在大多数情况下，您能使用rx-cos-slot all命令。在我们的测试案例中，我们不支持tofab排队的一些卡，因此我们不可能总是使用rx-cos-slot all命令。反而，我们分配我们的插槽表到用于测验的线卡。

```
!
slot-table-cos B2-Table
destination-slot all B2
multicast B2 !--- If you don't fulfill the steps above, you will not be able to reach the goal of 0 drops for Multicast traffic. With no rx-cos configured, multicast will be treated in the default queue, meaning it will drop as soon as there is congestion. !
```

Tx QoS

现在您可以配置您的tx-cos。选择一名称对于您的tx qos，例如“cos-queue-group B2”。

每个优先值您要配置需要的一种丢弃行为能被映射到分开random-detect-label。

```
precedence 0 random-detect-label 0
```

```
precedence 1 random-detect-label 1
```

```
precedence 2 random-detect-label 2
```

```
precedence 3 random-detect-label 3
```

对于改进的差额轮询(MDRR)，请映射每优先对MDRR队列。在我们的示例中，我们映射优先0-3对同一个MDRR队列为了保留视频的(组播数据流)带宽。此映射提供请求的动作。

```
precedence 0 queue 0
```

```
precedence 1 queue 0
```

```
precedence 2 queue 0
```

```
precedence 3 queue 0
```

```
precedence 4 queue 4
```

语音标记用优先5，是优先5为什么被映射对低延时队列。

```
precedence 5 queue low-latency
```

```
precedence 6 queue 6
```

```
precedence 7 queue 6
```

现在您必须配置其中每一的下降的行为random-detect-labels。在测试期间，这些编号更改，直到的值给希望的丢弃模式找到。关于详细信息，请参阅[Expected Results部分](#)。队列深度被测量在物理队列和不在MDRR或红色标记队列。

```
random-detect-label 0 50 5000 1
```

```
random-detect-label 1 60 6000 1
```

```
random-detect-label 2 70 7000 1
```

```
random-detect-label 3 80 8000 1
```

在Cisco 12000上，通过给另外MDRR队列每重要性创建基于类的，加权公平排队(CBWFQ)是可能的行为。默认权重是10每个队列。字节数传送每个MDRR周期是权重值的功能。值为1含义1500个字节每个周期。值为10含义1500+(9*512)字节每个周期”。

```
queue 0 20
```

```
queue 4 20
```

```
queue 6 20
```

```
!
```

[接口映射](#)

每个接口需要为WRED配置。使用命令，这执行：

- **configure terminal**
- **接口gig 6/0**
- **tx-cos B2**

无丢弃启动

除非其他陈述，产生的流使用以下值：

MTU all three data streams 300byte, MTU voice 80byte, MTU MC 1500byte

126Mb MC, 114Mb voip

这导致240Mb背景流(VoIP和组播)。

我们然后添加相同大小的四数据流，但是与优先0-3 (每数据流一个优先值)。

四个每路 151Mb 的数据流

此配置给总带宽844Mb。下面的图表显示有0丢包和非常低延时(大约50我们-微秒-每数据流的，包括语音)。

四个每路 160Mb 的数据流

此配置给总带宽880Mb。下面的图表显示数据包开始丢弃从优先0类和语音的延迟增加对6毫秒-毫秒。

四个每路 167Mb 的数据流

此配置给总带宽908Mb。丢包为优先当前开始1类。语音流量的延迟仍然是相同的。

注意： 数据流未在增加前被终止，因此丢包之间数量的差异在数据流0和1的是渐增的。

四个每路 191Mb 的数据流

当总带宽增加时，数据包开始从优先2队列丢弃。我们尝试为千兆以太网接口到达的总带宽当前是1004Mb。这在下面图表的顺序错误计数器说明。

四个每路 244Mb 的数据流

优先的3顺序错误开始增加。这是第一个符号丢包从该队列将开始。我们尝试派出GbE接口的带宽总量当前是1216 Mb。注意在组播(MC)的丢包，并且语音队列零和语音队列的延迟未增加。

正在停止和开始

所有数据流被终止了并且开始生成清除了计数器的图表。这显示如何在严重拥塞时将查找。正如您下面看到的行为是希望的一个。

卸除所有 QoS

要证明，QoS在拥塞时确实改进性能，QoS当前删除，并且接口被堵塞了。结果下面(产生的流使用以下值，除非其他陈述)。

MTU all three data streams 300byte, MTU voice 80byte, MTU MC 1500byte

126Mb MC, 114Mb VoIP

这导致240Mb背景流(VoIP和组播)。

我们然后添加相同大小的四数据流，但是与优先0-3 (每数据流一个优先值)。

四个每路 153Mb 的数据流

这给总共852Mb。有0 drops和较少的延迟然后50我们。

四个每路 158Mb 的数据流

我们开始丢弃在大致同样的利用率和，当WRED应用时(872Mb)。差异当前是有语音数据包延迟14毫秒(更多比两倍同等数量正如在WRED测验)，并且丢包从所有类均等地发生，包括VoIP和组播。

添加入口负载

到目前为止，所有测验通过千兆以太网接口只包括传送。要验证接口如何在我们也堵塞在另一个方向的接口的情况起反应，以下测验进行了。

对于此测验，我们装载与一个的总数1056 Mb的千兆以太网接口。这导致在优先0-2的丢包，在优先3流量的没有丢包。(即，MC和VOIP保持同样没有丢包)。我们然后添加了在另一个方向的流量，同样多流量，象信息包生成器能通过千兆以太网接口派出。结果是相当印象深刻的：接收的拥塞不干涉传送的队列，并且接收的流量的延迟极低的，少于13我们语音的。

12-RP#show interfaces g 6/0

使用show interfaces命令，您能监控在千兆链路的负载：

```
Router#show interfaces gig 6/0 GigabitEthernet6/0 is up, line protocol is up Hardware is GigMac
3 Port GigabitEthernet, address is 0004.de56.c264 (bia 0004.de56.c264) Internet address is
178.6.0.1/24 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec, rely 255/255, load 232/255
Encapsulation ARPA, loopback not set Keepalive set (10 sec) Full-duplex mode, link type is
force-up, media type is SX output flow-control is unsupported, input flow-control is off ARP
type: ARPA, ARP Timeout 04:00:00 Last input 00:00:05, output 00:00:05, output hang never Last
clearing of "show interface" counters 08:52:40 Queueing strategy: random early detection (WRED)
Output queue 0/40, 2174119522 drops; input queue 0/75, 0 drops 30 second input rate 838969000
bits/sec, 792079 packets/sec 30 second output rate 910819000 bits/sec, 464704 packets/sec
7584351146 packets input, 1003461987270 bytes, 0 no buffer Received 0 broadcasts, 0 runts, 0
giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 watchdog, 514
multicast, 0 pause input 11167110605 packets output, 2241229569668 bytes, 0 underruns 0 output
errors, 0 collisions, 0 interface resets 0 babbles, 0 late collision, 0 deferred 0 lost carrier,
0 no carrier, 0 pause output 0 output buffer failures, 0 output buffers swapped out
```

更改流的大小

要验证测试结果不归结于带宽是相同的为所有数据流，我们更改数据流，以便他们传送不同的相当数量数据。我们也设法更改最大传输单元(MTU)，因此为每数据流是不同的。使用已配置的队列值，结果仍然是相同的，首先，然后下降优先0 1，然后2和最后优先3。

用 10 端口千兆以太网引擎 4 线路卡验证

因为VoIP队列(低延时队列)的延迟在测验相当高，我们执行了同一测验与10波尔特千兆以太网引擎

4线卡。正如所料，与此线卡的结果是更加好的看待在低延时队列(LLQ)的延迟。结果是相同的关于丢弃如何发生了。LLQ的延迟是在10us附近，是1:1000在3波尔特千兆以太网引擎2线卡的延迟。

[相关信息](#)

- [了解和配置 Cisco 12000 系列互联网路由器上的 MDRR 和 WRED](#)
- [如何解读show controller frfab的输出|在Cisco 12000系列互联网路由器体系结构的tofab queue 命令](#)
- [服务类型在互联网协议套件，优先\(RFC-1349\)](#)
- [加权 随机 早期 检测](#)
- [技术支持和文档 - Cisco Systems](#)