

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

本文档说明如何在多服务环境中配置Cisco 12000系列线卡以进行加权随机早期检测(WRED)。如

[RFC 2309](#) 中所述。

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 | Cisco 12000系列互联网路由器上的tofab queue命令](#)
- [Internet协议簇中的服务类型，优先级\(RFC-1349\)](#)
- [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：

- 任何支持Cisco 12000系列互联网路由器的Cisco IOS®软件版本。通常是12.0S和12.0ST版本。
- 本文档涵盖所有Cisco 12000平台。这包括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系列是用于构建高带宽IP核心网络的最常用平台之一。此平台提供配置服务质量 (QoS)的独有可能性。

由于在同一网络中混合不同类型的IP流量（如IP语音 — VoIP和组播）越来越普遍，因此对优先级和受控丢弃行为的要求变得极其重要，在很多情况下，启动VoIP等新服务时，成功与失败之间的区别就很大。

不同类型IP流量的网络要求不在本文档的范围之内。本文档重点介绍为查找适用于不同线卡(包括Cisco 12000系列、3端口千兆以太网 (3端口GbE) 线卡)的配置而进行的实验室测试。这些测试的结果表明，3端口GbE引擎2线卡非常适合包含语音、数据和组播流量的网络环境。它还证明配置QoS在拥塞网络中起着真正的作用。

用于测试的优先等级

分配给不同类的优先级值在整个网络中必须相同。您需要确定通用策略。

类	优先	Traffic
---	----	---------

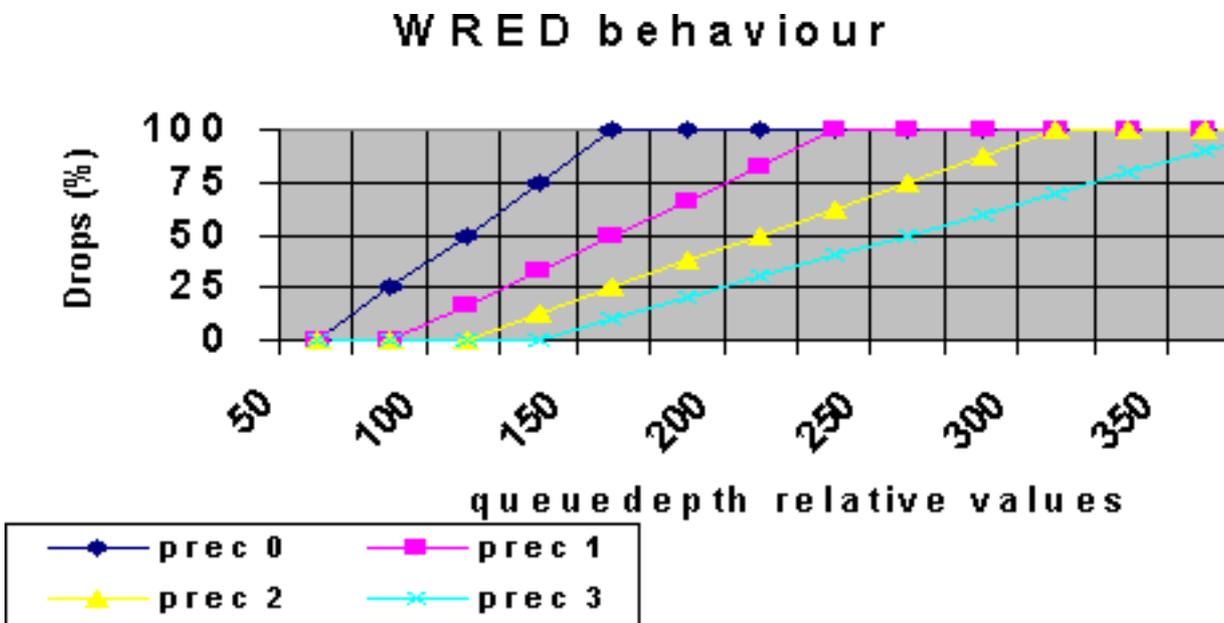
	级	
恶劣交通		所有未识别的网络流量 (网外)
网上 — 网上	1	SP网络内的流量 (网内)
Internet服务提供商(ISP)服务	2	ISP流量、SMTP、POP、FTP、DNS、Telnet、SSH、www、https
SME (中小型企业)	3	企业客户 , 金牌服务
实时、非语音	4	电视、实时游戏
语音	5	RTP VOIP流量
网络控制消息	6-7	边界网关协议(BGP)和其他控制消息

IP 包的着色

如果要在网络核心中实施QoS，前提条件是服务提供商完全控制网络中传输的所有IP数据包的优先级值。唯一的方法是在所有数据包进入网络时对其进行标记，不区分它们是来自客户/最终用户端还是来自Internet。核心中不应进行标记或着色。

预期结果

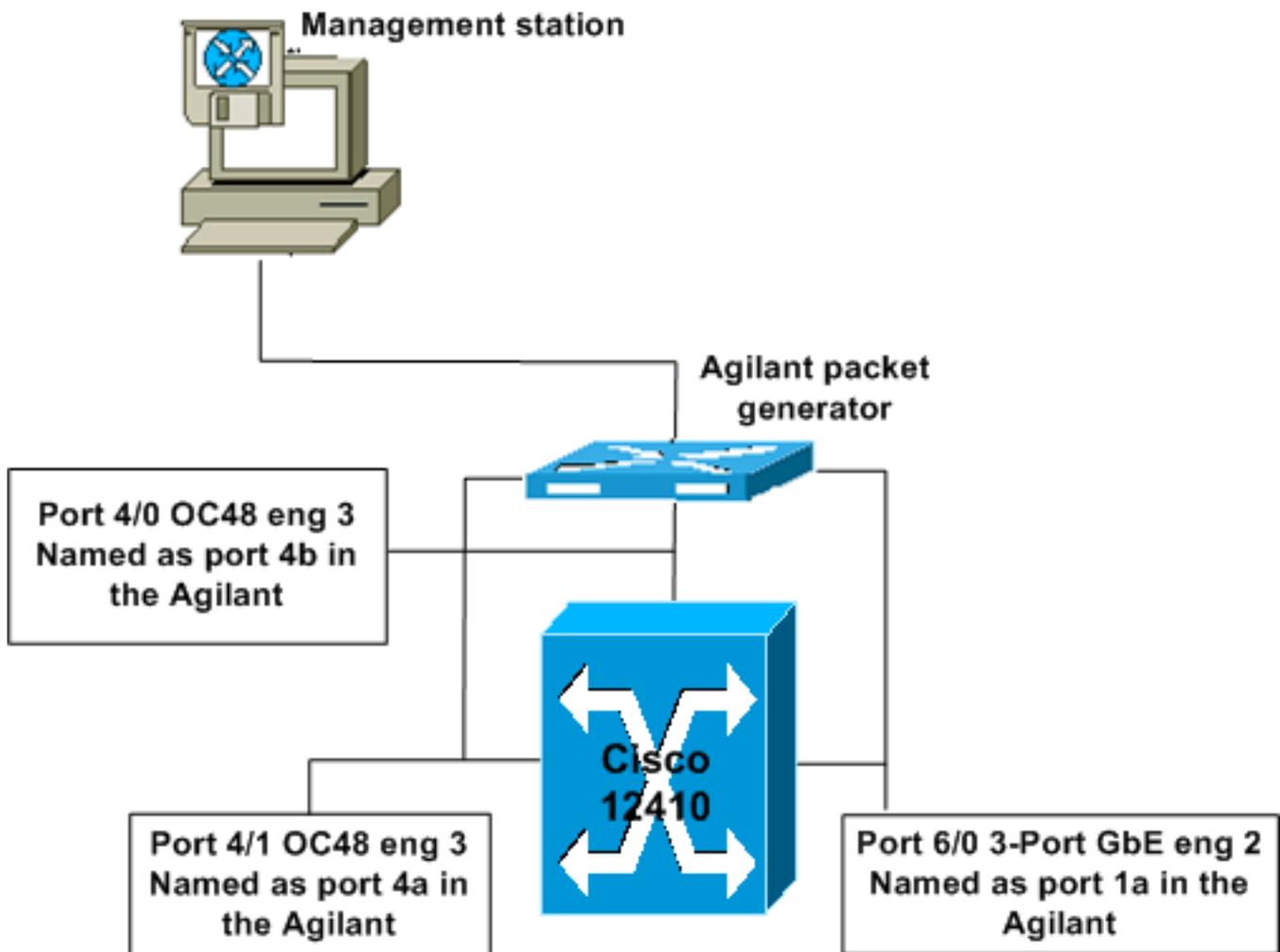
此设计的目标是在0-3类中具有真正的WRED行为。这意味着我们希望出现这样的情况：在拥塞期间开始丢弃优先级为0的数据包。之后，如果拥塞继续，我们还应开始丢弃优先级1，然后丢弃优先级2和优先级3。下图对此进行了说明。



语音数据包的延迟应尽可能低，语音和组播流量应完全不丢弃。

网络设置

为测试和评估配置，我们使用Cisco 12410和Agilant的数据包生成器。Cisco 12000路由器运行的工程版本基于Cisco IOS软件版本12.0(21)S1。



3 端口 GbE 引擎2 排队实施

引擎2卡通常有8个fab队列和1个低延迟队列，每个目标插槽有8个tofab队列。此外，还有一个单独的tofab组播队列。在3端口GbE卡上，每个物理端口仅有一个fab队列。在测试中，应用的配置指定了更多队列。结果显示，3端口GbE卡接受此配置，队列会自动映射到可用的队列。

Cisco IOS 软件的引擎 2 WRED 算法

配置最小和最大队列深度值时，必须完全了解引擎2线卡中用于WRED的算法。代码不关心配置的最小值；相反，它使用自己的公式（基于配置的最大值）来设置最小值。

公式：

最小值=最大值 — (不产生负结果的2的最大幂)

本测试中使用的值产生了以下编程到专用集成电路(ASIC)的最小值：

优先级	已配置最小值	配置的最大值	2的最高功率	ASIC中的最小值
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，如果在配置WRED参数时未考虑此问题，则最终可能会导致数据包处理行为不正确。以下示例中显示了这一点：

优先级	已配置最小值	配置的最大值	2的最高功率	ASIC中的最小值
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，即使将值配置为首先根据规则0、1、2和3（以上）开始丢弃，当值写入ASIC时，您实际上开始丢弃优先级0、优先级2、优先级1和最后优先级3。无法查看引擎2卡的ASIC中配置了哪些值。如果在引擎3卡上应用配置，则配置中显示的值将是实际值（重新计算的最小值）。

QoS 配置

有关QoS配置的详细信息，请阅读[读了解和配置Cisco 12000系列Internet路由器上的MDRR和WRED](#)。

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，可以使用**rx-cos-slot all**命令。在测试案例中，我们有一些卡不支持tofab排队，因此我们不能始终使用**rx-cos-slot all**命令。相反，我们将插槽表分配给测试中使用的线卡。

```
!
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tx-cos。为tx qos选择名称，例如“cos-queue-group B2”。

要配置丢弃行为的每个优先级值需要映射到单独的随机检测标签。

```
precedence 0 random-detect-label 0
precedence 1 random-detect-label 1
```

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

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

对于修改的差额轮询(MDRR)，将每个优先级映射到MDRR队列。在我们的示例中，我们将优先级0-3映射到同一MDRR队列，以便为视频（组播流量）保留带宽。此映射提供请求的行为。

```
precedence 0 queue 0
```

```
precedence 1 queue 0
```

```
precedence 2 queue 0
```

```
precedence 3 queue 0
```

```
precedence 4 queue 4
```

语音标有优先级5，因此优先级5映射到低延迟队列。

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

```
precedence 6 queue 6
```

```
precedence 7 queue 6
```

现在，您必须为每个随机检测标签配置丢弃行为。在测试期间，这些数字被更改，直到发现值提供所需的丢弃模式。有关详细信息，请参阅[预期结果](#)部分。队列深度在物理队列上测量，而不是在MDRR或RED-Label队列上测量。

```
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，可以通过给不同的MDRR队列赋予权重来创建基于类的加权公平队列(CBWFQ)行为。默认权重为每个队列10。每个MDRR周期传输的字节数是权重值的函数。值1表示每个周期1500字节。值10表示每个循环1500+(9*512)字节。”

```
queue 0 20
```

```
queue 4 20
```

```
queue 6 20
```

```
!
```

[接口映射](#)

每个接口都需要配置为WRED。这是使用以下命令完成的：

- `configure terminal`
- `interface gig 6/0`
- `tx-cos B2`

无丢弃启动

除非另有说明，否则生成的流使用以下值：

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

126Mb MC, 114Mb voip

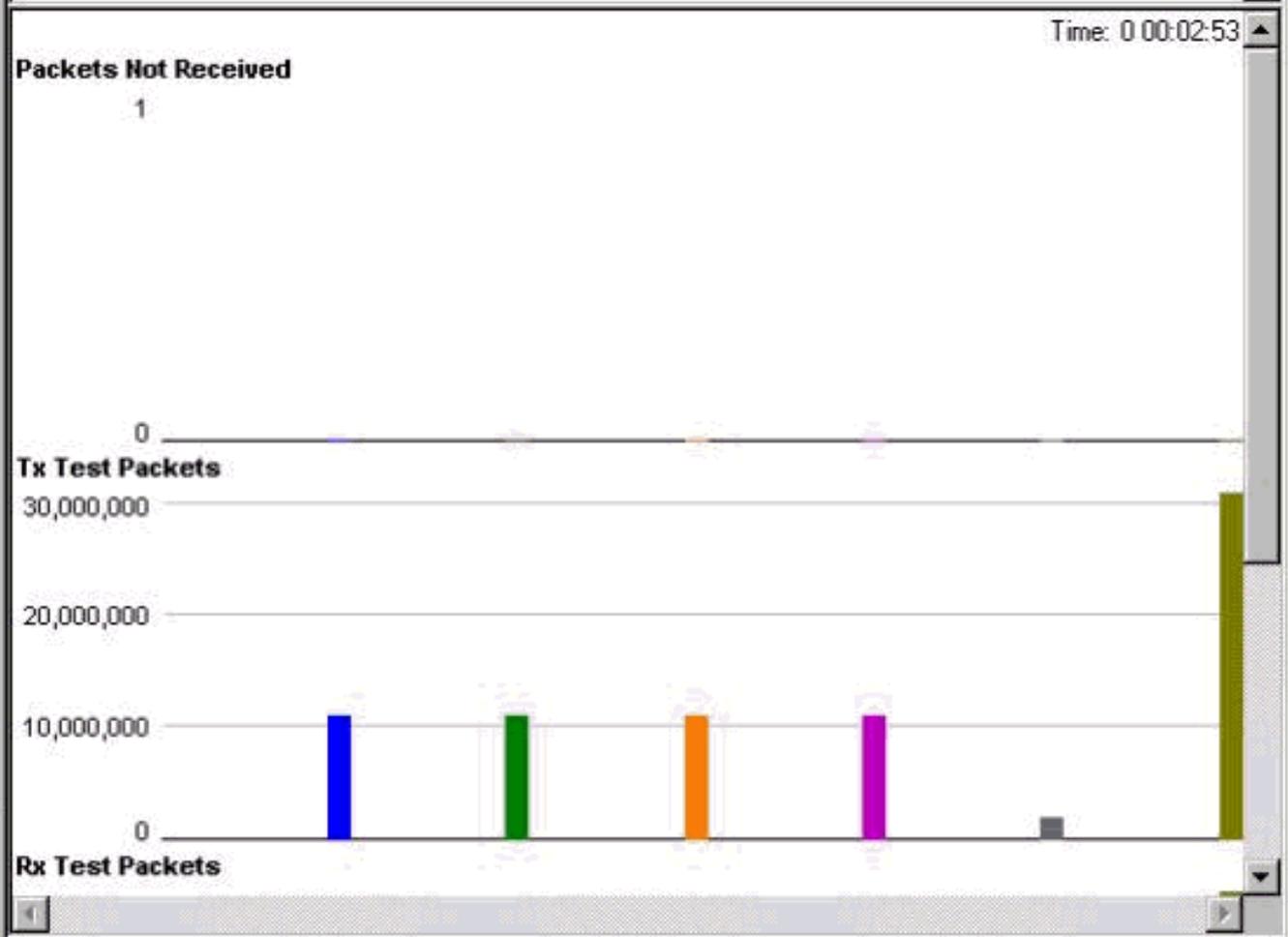
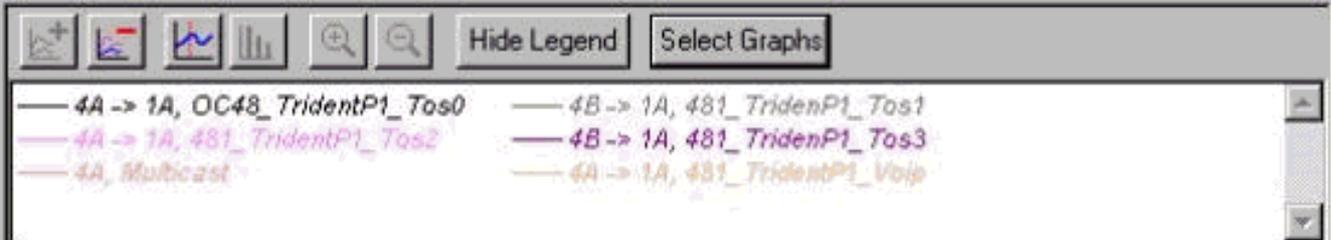
这将产生240Mb (VoIP和组播) 的后台流。

然后，我们添加四个大小相同但优先级为0-3 (每个数据流一个优先级值) 的数据流。

四个每路 151Mb 的数据流

此配置提供的总带宽为844Mb。下图显示有0个丢包和极低的延迟(每个流 (包括语音) 大约50微秒)。

Ports/Streams	Seq Errors	Packets Not R...	Avera...	Max Latency...	T: ▲
- 4B -> 1A, 481_TridenP1_Tos1	0		30.39	44.36	
- 4B -> 1A, 481_TridenP1_Tos3	0		35.29	48.94	
- 4A -> 1A, 481_TridenP1_Tos2	0		33.73	48.98	
- 4A -> 1A, 0C48_TridenP1_Tos0	0		28.90	44.50	
- 4A -> 1A, 481_TridenP1_Voip	0		32.99	52.84	
+ Port 1B			51.04	56.58	
+ Port 1C			45.12	49.80	
Port 1D		0			



四个每路 160Mb 的数据流

此配置提供的总带宽为880Mb。下图显示数据包开始从优先级0类丢弃，并且语音延迟已增加到6毫秒 — 毫秒。

Ports/Streams	Seq Errors	Packets Not R...	Avera...	Max Latency...	T: ▲
- 4B -> 1A, 481_TridentP1_Tos1	0		1566.55	13117.44	
- 4B -> 1A, 481_TridentP1_Tos3	0		1571.17	13118.92	
- 4A -> 1A, 481_TridentP1_Tos2	0		1570.53	13124.36	
- 4A -> 1A, DC48_TridentP1_Tos0	114207	162429	1408.37	13117.96	
- 4A -> 1A, 481_TridentP1_Voip	0		815.62	6337.22	
+ Port 1B			847.10	6474.38	
+ Port 1C			834.87	6395.62	
+ Port 1D		0			

Hide Legend Select Graphs

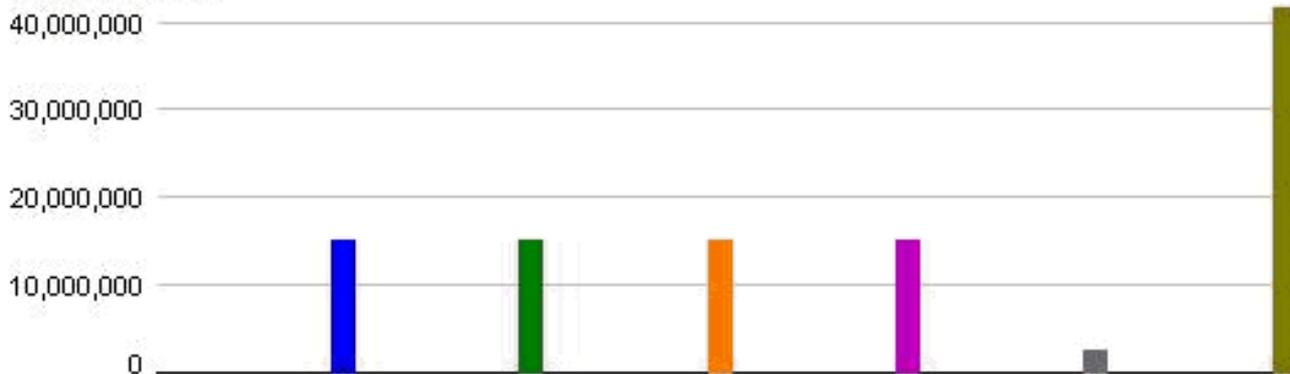
— 4A -> 1A, OC48_TridentP1_Tos0 — 4B -> 1A, 481_TridentP1_Tos1
 — 4A -> 1A, 481_TridentP1_Tos2 — 4B -> 1A, 481_TridentP1_Tos3
 — 4A, Multicast — 4A -> 1A, 481_TridentP1_Voip

Time: 0 00:03:54 ▲

Packets Not Received



Tx Test Packets



Rx Test Packets

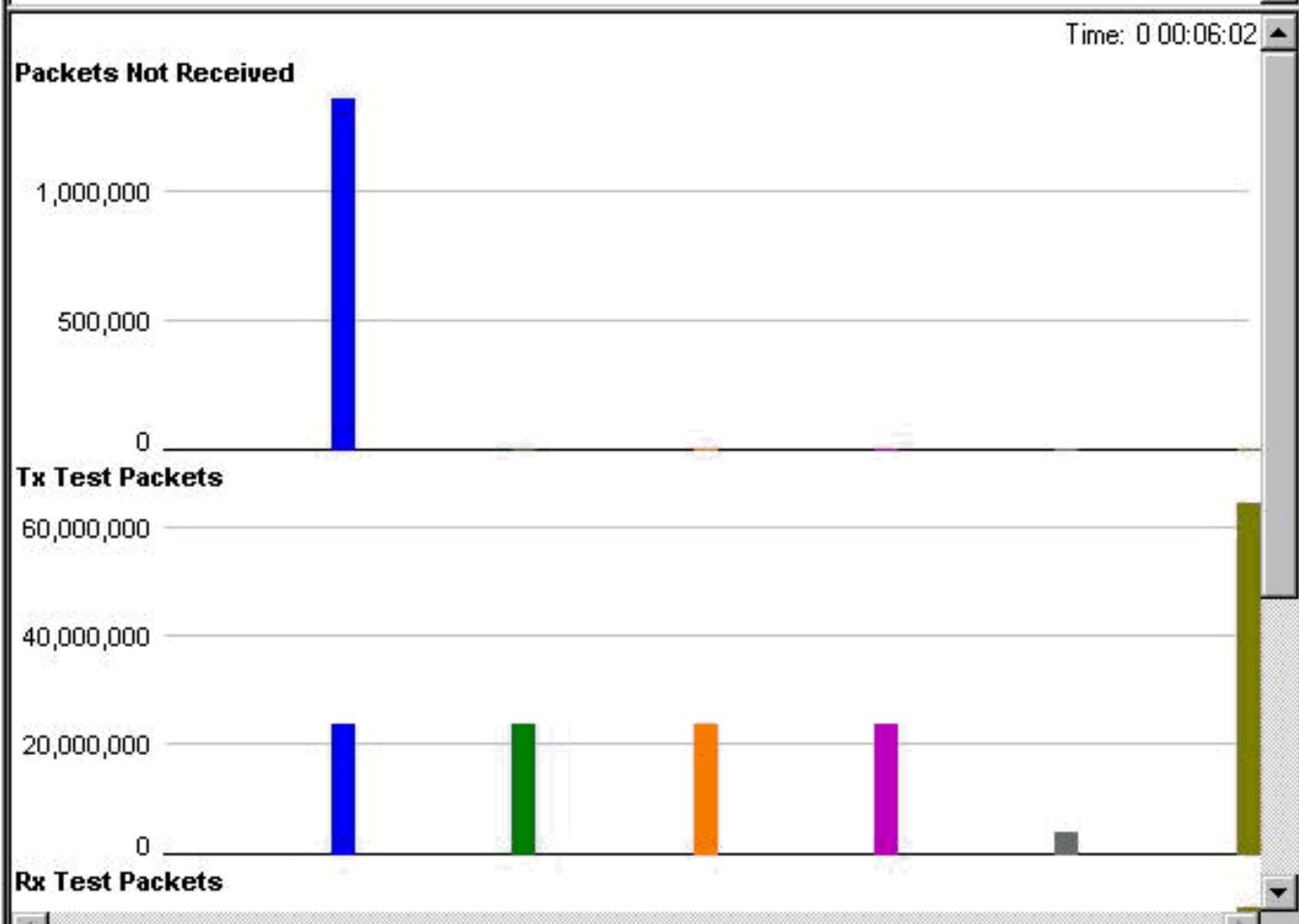
四个每路 167Mb 的数据流

此配置提供的总带宽为908Mb。现在，优先级1类也开始丢弃。语音流量的延迟仍然相同。

注意：流在增加之前未停止，因此流0和1中的丢弃数之间的差是累积的。

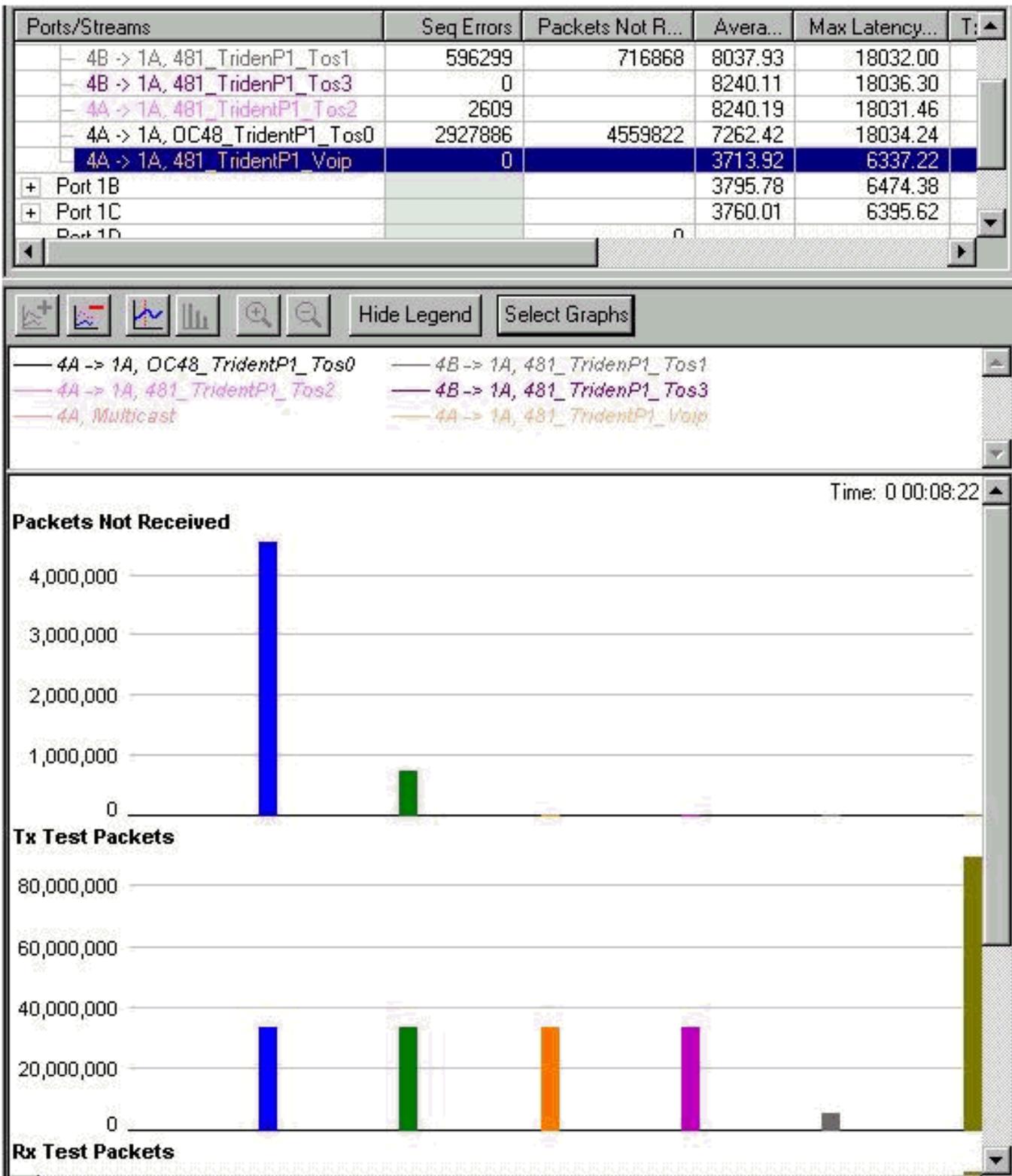
Ports/Streams	Seq Errors	Packets Not R...	Avera...	Max Latency...	T: ▲
- 4B -> 1A, 481_TridentP1_Tos1	12192		5387.09	14537.54	
- 4B -> 1A, 481_TridentP1_Tos3	0		5396.54	14535.50	
- 4A -> 1A, 481_TridentP1_Tos2	0		5397.81	14538.58	
- 4A -> 1A, OC48_TridentP1_Tos0	774949	1357422	4956.01	14539.28	
- 4A -> 1A, 481_TridentP1_Voip	0		2735.27	6337.22	
+ Port 1B			2800.05	6474.38	
+ Port 1C			2772.22	6395.62	
+ Port 1D		0			

— 4A -> 1A, OC48_TridentP1_Tos0 — 4B -> 1A, 481_TridentP1_Tos1
 — 4A -> 1A, 481_TridentP1_Tos2 — 4B -> 1A, 481_TridentP1_Tos3
 — 4A, Multicast — 4A -> 1A, 481_TridentP1_Voip



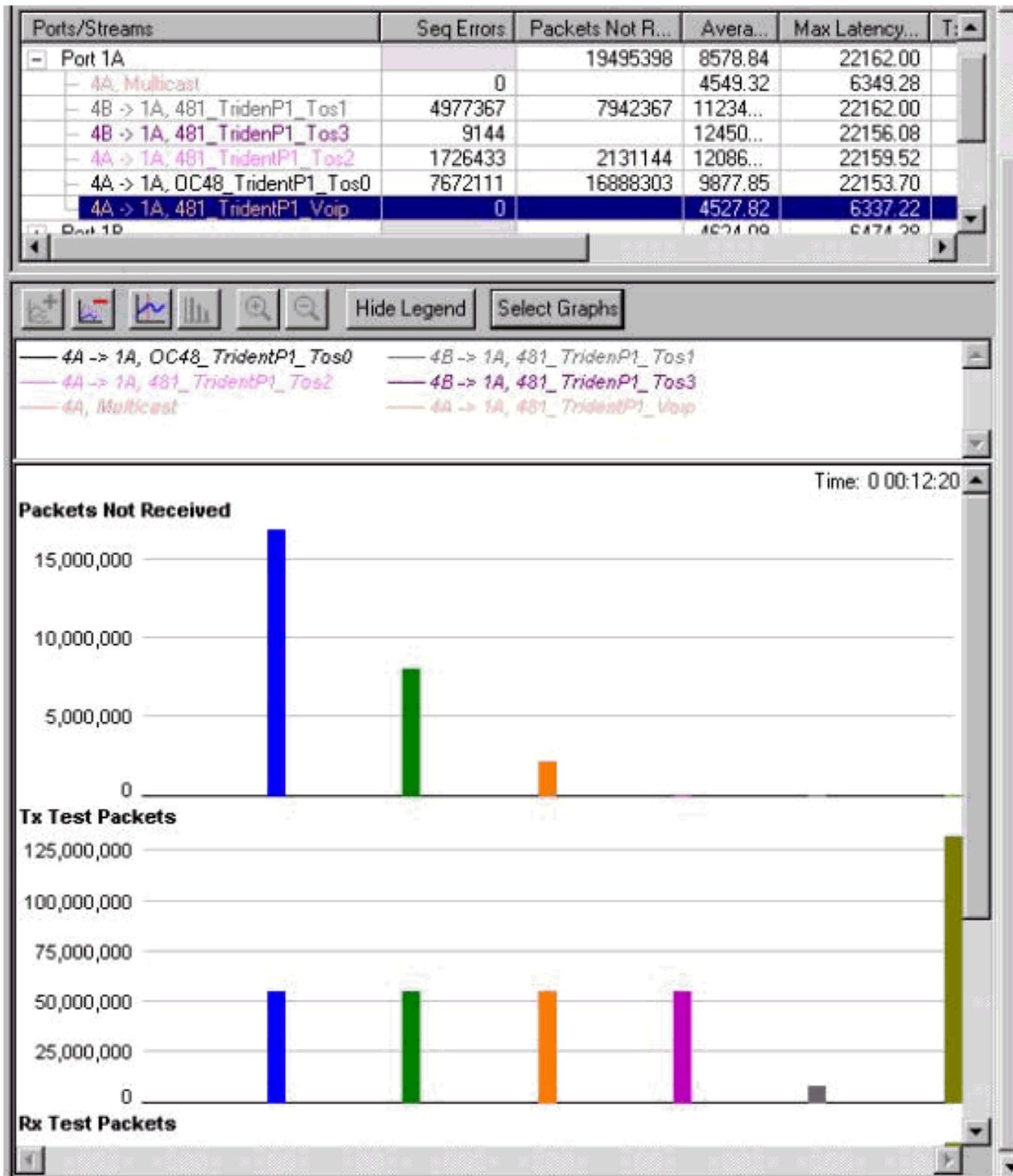
四个每路 191Mb 的数据流

当总带宽增加时，数据包也开始从优先级2队列中丢弃。我们现在尝试为千兆以太网接口提供的总带宽为1004Mb。这在下图的序列错误计数器中说明。



四个每路 244Mb 的数据流

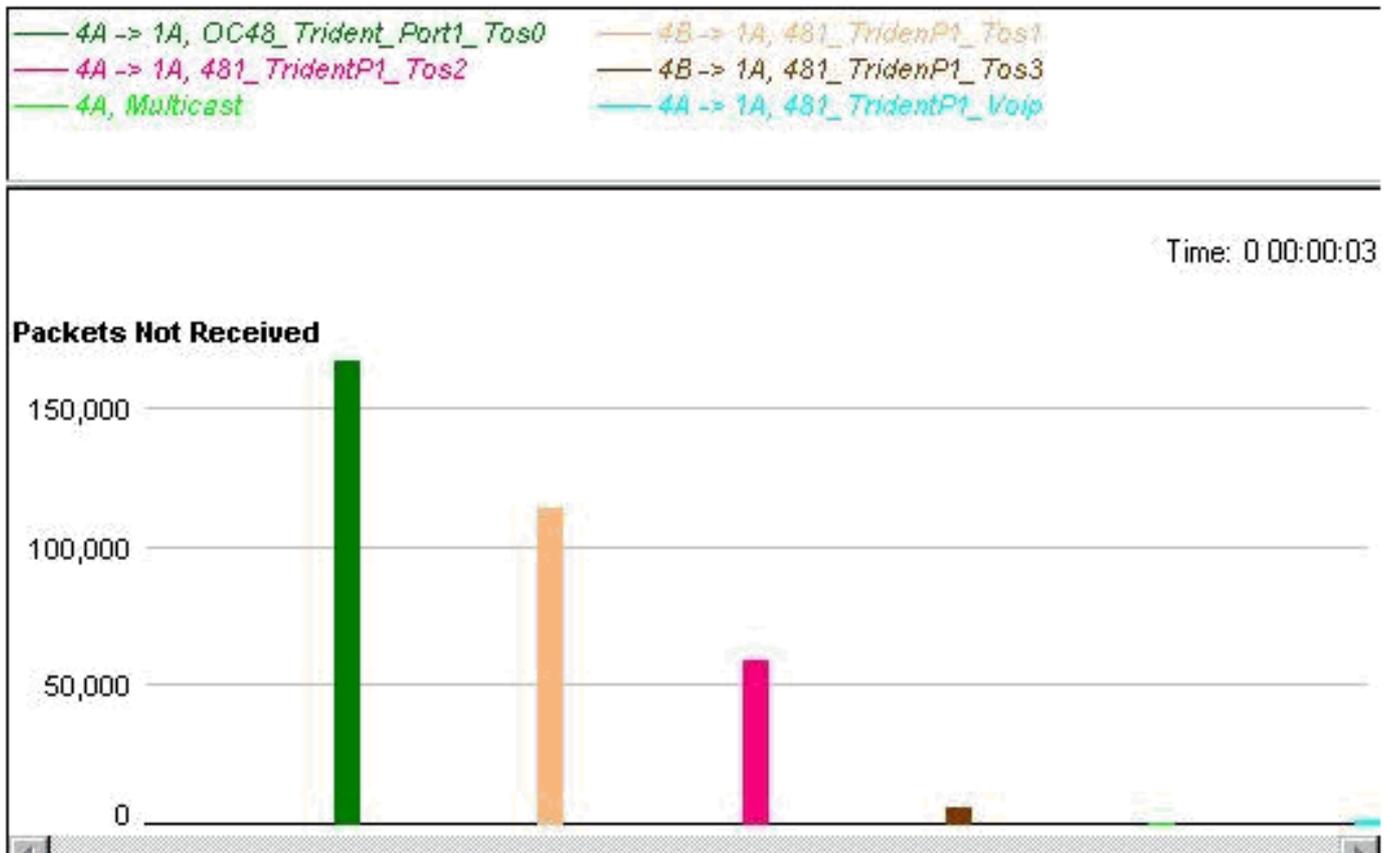
优先级3的序列错误也开始增加。这是丢弃将从该队列开始的第一个迹象。我们尝试从GbE接口发送的总带宽量现在为1216 Mb。请注意，组播(MC)和语音队列上的丢包数仍然为零，并且语音队列的延迟没有增加。



停止和启动

所有流都已停止并开始生成已清除计数器的图形。这显示了在严重拥塞时，它的外观。如下所示，行为是理想行为。

Ports/Streams	Packets Not Received
All Ports	244885
[-] Port 1A	259941
[-] 4A -> 1A, OC48_Trident_Port1_Tos0	133621
[-] 4A -> 1A, 481_TridentP1_Tos2	45024
[-] 4A -> 1A, 481_TridentP1_Voip	
[-] 4A, Multicast	
[-] 4B -> 1A, 481_TridentP1_Tos1	89282
[-] 4B -> 1A, 481_TridentP1_Tos3	624



卸除所有 QoS

为了证明QoS确实在拥塞期间提高了性能，QoS现在已被删除，接口已拥塞。结果如下（除非另有说明，否则生成的流使用以下值）。

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

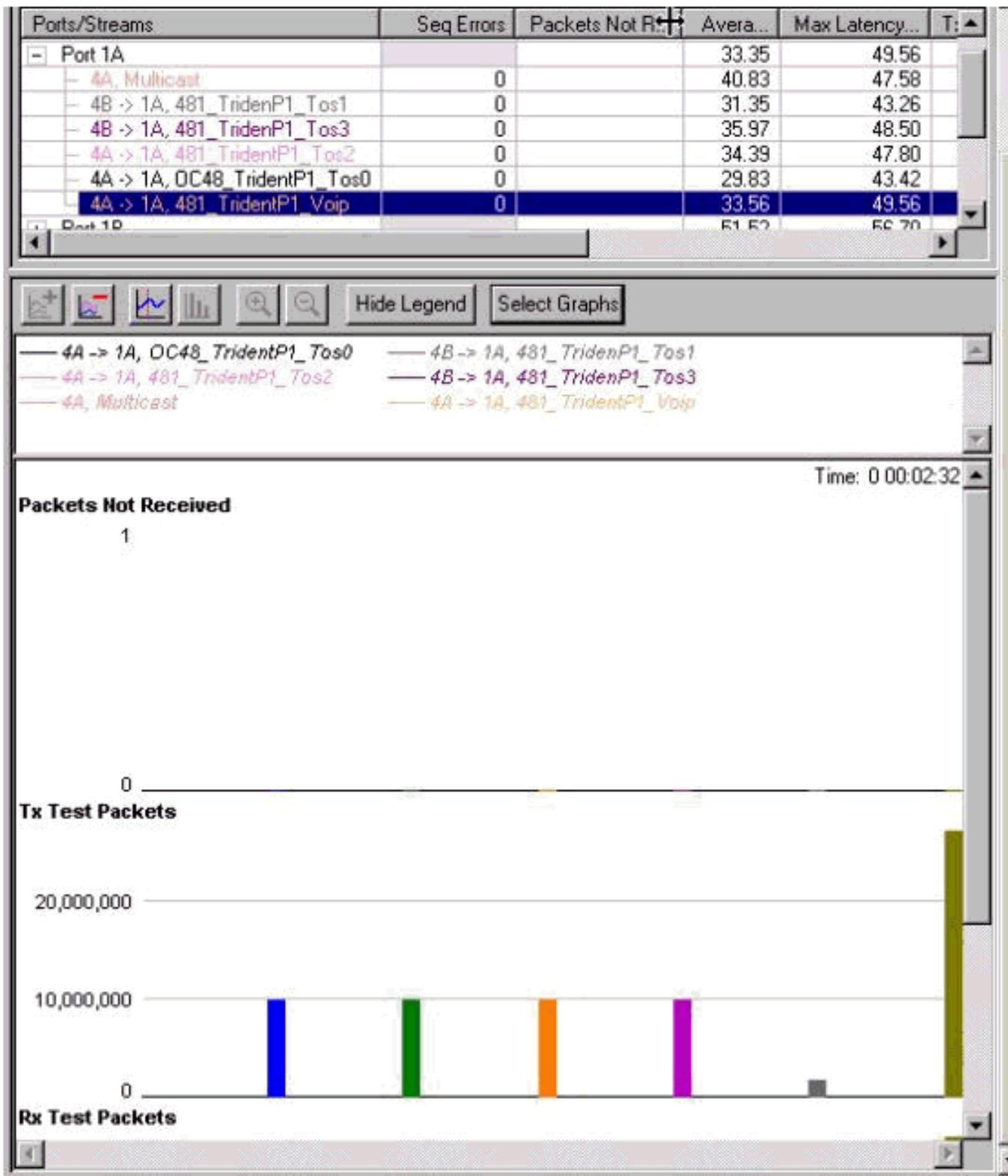
126Mb MC, 114Mb VoIP

这将产生240Mb（VoIP和组播）的后台流。

然后，我们添加四个大小相同但优先级为0-3（每个数据流一个优先级值）的数据流。

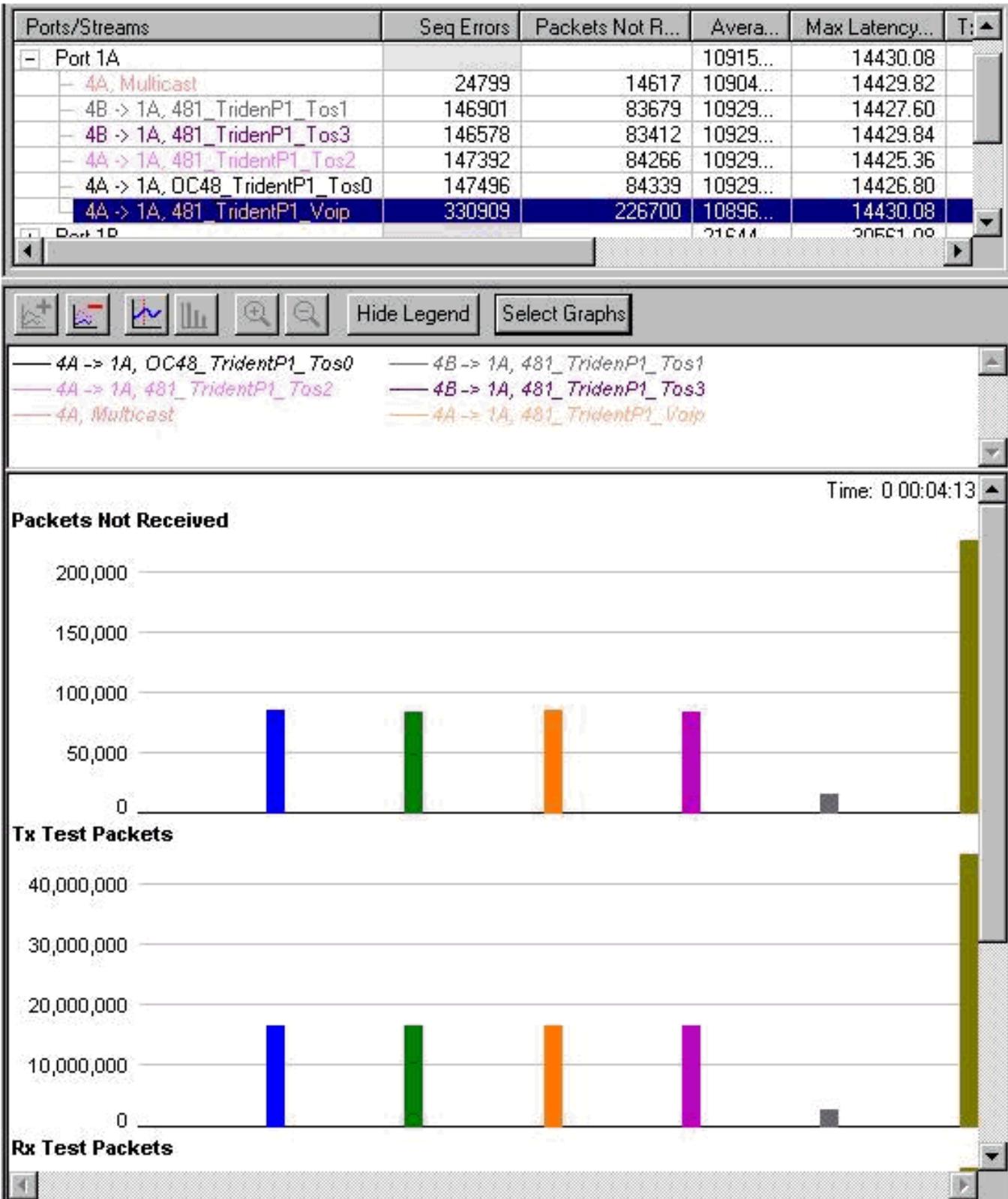
四个每路 153Mb 的数据流

这总计为852Mb。有0个丢包，延迟小于50。



四个每路 158Mb 的数据流

我们开始降低与应用WRED时(872Mb)的利用率大致相同的利用率。现在的区别在于，语音数据包的延迟为14毫秒（比WRED测试的延迟多一倍），并且丢弃在所有类别（包括VoIP和组播）中都同样发生。



添加入口负载

到目前为止，所有测试仅包括通过千兆以太网接口进行传输。为了验证接口在另一方向也导致接口拥塞的情况下如何反应，我们进行了以下测试。

在本测试中，我们加载了总量为1056 Mb的千兆以太网接口。这会导致优先级0-2上的丢弃，优先级3流量上的丢弃。（MC和VOIP保持不变，即无丢包）。然后，我们向另一个方向添加流量，即数据包生成器能够通过千兆以太网接口发送的流量。结果令人印象深刻：接收拥塞完全不干扰发送队列，并且接收流量的延迟极低，语音的延迟小于13微秒。

-	Port 1A	21976403	12050.70	20237.92
	- 4A, Multicast		6277.06	6402.88
	- 4B -> 1A, 481_TridenP1_Tos1	7729260	16763.90	20232.04
	- 4B -> 1A, 481_TridenP1_Tos3		17287.73	20237.92
	- 4A -> 1A, 481_TridenP1_Tos2	2090730	17139.98	20233.44
	- 4A -> 1A, OC48_TridenP1_Tos0	17210053	16519.08	20236.76
	- 4A -> 1A, 481_TridenP1_Voip		6245.21	6376.02
+	Port 1B		6379.18	6512.30
+	Port 1C		6323.52	6432.94
	Port 1D	0		
-	Port 4A	7	13.09	15.02
	- 1A -> 4A, 481_Triden_P3_tos3		13.37	15.02
	- 1A -> 4A, 481_TridenP3_Voip		12.67	14.30
	- 1A -> 4A, 48_TridenP3_tos2		13.23	14.94
-	Port 4B	6	13.11	14.64
	- 1A -> 4B, 48_Triden_P3_Tos0		13.11	14.62
	- 1A -> 4B, 481_TridenP3_Tos1		13.11	14.64

Ports/Streams	Tx Test Throughput (Mb/s)	Rx Test Throughput (Mb/s)
All Ports		
- Port 1A	747.47	858.75
- 4A, Multicast	126.24	126.24
- 4B -> 1A, 481_TridenP1_Tos1	189.55	142.03
- 4B -> 1A, 481_TridenP1_Tos3	189.54	189.54
- 4A -> 1A, 481_TridenP1_Tos2	189.54	175.70
- 4A -> 1A, OC48_TridenP1_Tos0	189.54	95.24
- 4A -> 1A, 481_TridenP1_Voip	130.00	130.00
+ Port 1B	0.00	126.24
+ Port 1C	0.00	126.24
Port 1D	0.00	0.00
- Port 4A	635.32	424.24
- 1A -> 4A, 481_Triden_P3_tos3	161.62	161.62
- 1A -> 4A, 481_TridenP3_Voip	101.01	101.01
- 1A -> 4A, 48_TridenP3_tos2	161.62	161.62
- Port 4B	379.09	323.23
- 1A -> 4B, 48_Triden_P3_Tos0	161.62	161.62
- 1A -> 4B, 481_TridenP3_Tos1	161.62	161.62

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

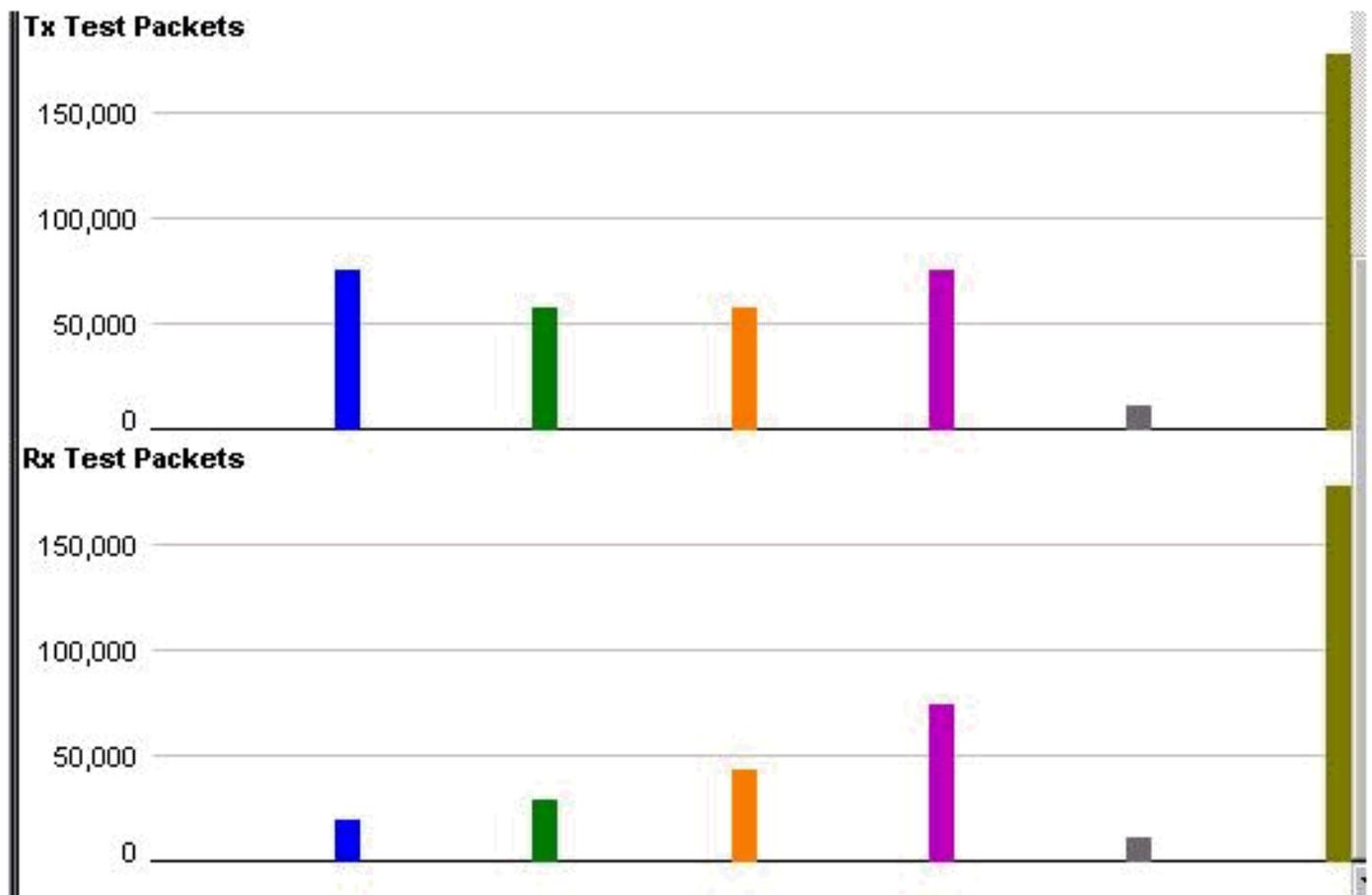
您可以使用show interfaces命令监控千兆链路上的负载：

```
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

为了验证测试结果是否不是由于所有数据流的带宽相同，我们更改了数据流，以便它们传输不同数量的数据。我们还尝试更改最大传输单位(MTU)，因此每个数据流的传输单位不同。对于配置的队列值，结果仍然相同，先丢弃优先级0，然后丢弃优先级1，然后丢弃优先级2，最后丢弃优先级3。



用 10 端口千兆以太网引擎 4 线路卡验证

由于测试中的VoIP队列（低延迟队列）的延迟相当高，因此我们使用10端口千兆以太网引擎4线卡执行了相同的测试。如预期，此线卡在低延迟队列(LLQ)中的延迟方面的结果要好得多。结果与下降发生方式相同。LLQ的延迟约为10us，是3端口千兆以太网引擎2线卡的延迟的1:1000。

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 | Cisco 12000系列互联网路由器上的tofab queue命令](#)
- [Internet协议簇中的服务类型，优先级\(RFC-1349\)](#)
- [加权 随机 早期 检测](#)
- [技术支持和文档 - Cisco Systems](#)