

# 執行Cisco IOS系統軟體的Catalyst 6500/6000系列交換器上的QoS輸出排程

## 目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[輸出佇列捨棄](#)

[Catalyst 6500/6000上不同線卡的輸出佇列功能](#)

[瞭解連線埠的佇列功能](#)

[Catalyst 6500/6000上的配置、監控和輸出排程示例](#)

[組態](#)

[監控輸出計畫和驗證配置](#)

[輸出排程示例](#)

[使用輸出排程減少延遲和抖動](#)

[減少延遲](#)

[減少抖動](#)

[結論](#)

[相關資訊](#)

## 簡介

使用輸出排程可以確保在嚴重超訂用情況下不會丟棄重要流量。本檔案將討論Catalyst 6500/6000交換器上輸出排程所涉及的所有技巧和演演算法。本檔案也說明如何在執行Cisco IOS®軟體的Catalyst 6500/6000上設定和驗證輸出排程的作業。

如需加權隨機早期偵測(WRED)、加權循環配置資源(WRR)和尾部丟棄的詳細資訊，請參閱執行CatOS系統軟體的Catalyst 6500/6000系列交換器上的[QoS輸出排程](#)。

## 必要條件

### 需求

本文件沒有特定需求。

### 採用元件

本文件所述內容不限於特定軟體和硬體版本。

## 慣例

如需文件慣例的詳細資訊，請參閱[思科技術提示慣例](#)。

## 背景資訊

### 輸出佇列捨棄

輸出丟棄是由介面擁塞引起的。此問題的一個常見原因可能是來自高頻寬鏈路的流量被交換到低頻寬鏈路，或者來自多個入站鏈路的流量被交換到單個出站鏈路。

例如，如果大量突發流量從千兆介面傳入並切換到100Mbps介面，則這可能導致100Mbps介面上的輸出丟棄增加。這是因為由於傳入和傳出頻寬之間的速度不匹配，該介面上的輸出隊列被超額流量壓倒。傳出介面的流量速率無法接受所有應傳出的封包。

為了解決這個問題，最好的解決辦法是提高線路速度。但是，如果您不想增加線路速度，則有方法可以防止、減少或控制輸出下降。僅當輸出丟棄是短時間突發資料造成的結果時，才能防止輸出丟棄。如果輸出丟棄是由恆定高速流引起的，則無法阻止丟棄。但是，您可以控制它們。

## Catalyst 6500/6000上不同線卡的輸出佇列功能

如果您不確定連線埠的佇列功能，請發出**show queueing interface {gigabitethernet | fastethernet} mod/port** 命令。此處顯示的是**show queueing interface**命令輸出的第一行。此連線埠位於Supervisor Engine 1A線路卡上：

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
```

```
QoS is disabled globally
Trust state: trust DSCP
Default COS is 0
Transmit group-buffers feature is enabled
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          WRR low           2
      2          WRR high          2
      3          Priority          1
```

*!--- Output suppressed.*

輸出顯示，此埠的輸出隊列型別稱為1p2q2t。

檢視特定連線埠上可用的佇列型別的另一種方法是發出**show interface capabilities**命令：

```
la-orion#show interface gigabitethernet 6/2 capabilities
GigabitEthernet6/2
Model:                WS-SUP720-BASE
Type:                  No GBIC
Speed:                 1000
Duplex:                full
```

```

Trunk encap. type:      802.1Q,ISL
Trunk mode:            on,off,desirable,nonegotiate
Channel:              yes
Broadcast suppression: percentage(0-100)
Flowcontrol:          rx-(off,on,desired),tx-(off,on,desired)
Membership:           static
Fast Start:           yes
QoS scheduling:      rx-(1p1q4t), tx-(1p2q2t)
CoS rewrite:          yes
ToS rewrite:          yes
Inline power:         no
SPAN:                 source/destination
UDLD                  yes
Link Debounce:        yes
Link Debounce Time:   yes
Ports on ASIC:        1-2

```

## 瞭解連線埠的佇列功能

Catalyst 6500/6000交換器上提供多種型別的佇列。下表說明連線埠QoS架構的表示法：

傳輸 (Tx)/接收 (Rx)	佇列表示法	佇列數	優先順序佇列	WRR佇列數	WRR佇列的閾值數量和型別
Tx	2q2t	2	—	2	2個可配置的尾部丟棄
Tx	1p2q2t	3	1	2	2個可配置的WRED
Tx	1p3q1t	4	1	3	1個可配置的WRED
Tx	1p2q1t	3	1	2	1個可配置的WRED
Rx	1q4t	1	—	—	4個可配置的尾部丟棄
Rx	1p1q4t	2	1	1	4個可配置的尾部丟棄
Rx	1p1q0t	2	1	1	不可配置
Rx	1p1q8t	2	1	1	8個可配置的WRED
Tx	1p3q8t	4	1	3	8個可配置的WRED或尾部丟棄
Tx	1p7q8t	8	1	7	8個可配置的WRED或尾部丟棄
Rx	1q2t	1	—	—	1個可配置尾部丟棄= 1個不可配置
Rx	1q8t	1	—	—	8個可配置的尾部丟棄
Rx	2q8t	2	—	2	8個可配置的尾部丟棄

下表列出介面或埠的Rx和Tx端的一些模組和隊列型別。如果您的模組未在此處列出，請使用**show interface capabilities**命令確定可用的隊列功能。[Catalyst 6500/6000上不同線卡的輸出佇列功能一節](#)中詳述**show interface capabilities**命令。

模組	Rx隊列	Tx隊列
WS-X6K-S2-PFC2	1p1q4t	1p2q2t
WS-X6K-SUP1A-2GE	1p1q4t	1p2q2t
WS-X6K-SUP1-2GE	1q4t	2q2t
X6501-10GEX4	1p1q8t	1p2q1t
X6502-10GE	1p1q8t	1p2q1t
X6516-GBIC	1p1q4t	1p2q2t
X6516-GE-TX	1p1q4t	1p2q2t
X6416-GBIC	1p1q4t	1p2q2t
WS-X6416-GE-MT	1p1q4t	1p2q2t
X6316-GE-TX	1p1q4t	1p2q2t
X6408A-GBIC	1p1q4t	1p2q2t
X6408-GBIC	1q4t	2q2t
X6524-100FX-MM	1p1q0t	1p3q1t
X6324-100FX-SM	1q4t	2q2t
X6324-100FX-MM	1q4t	2q2t
WS-X6224-100FX-MT	1q4t	2q2t
X6548-RJ-21	1p1q0t	1p3q1t
X6548-RJ-45	1p1q0t	1p3q1t
X6348-RJ-21	1q4t	2q2t
X6348-RJ21V	1q4t	2q2t
X6348-RJ-45	1q4t	2q2t
X6348-RJ-45V	1q4t	2q2t
X6148-RJ-45V	1q4t	2q2t
X6148-RJ21V	1q4t	2q2t
X6248-RJ-45	1q4t	2q2t
WS-X6248A-TEL	1q4t	2q2t
X6248-TEL	1q4t	2q2t
X6024-10FL-MT	1q4t	2q2t

## [Catalyst 6500/6000上的配置、監控和輸出排程示例](#)

### [組態](#)

本節介紹在執行Cisco IOS軟體的Catalyst 6500/6000上設定輸出排程所需的所有步驟。有關Catalyst 6500/6000預設組態，請參閱[案例1:本檔案的啟用了QoS並使用了預設引數一節](#)。

Catalyst 6500/6000的組態包含以下五個步驟：

1. [啟用QoS](#)

2. [將每個可能的服務類別\(CoS\)值對映到隊列和閾值](#) (可選)
3. [配置WRR重量](#) (可選)
4. [配置分配給每個隊列的緩衝區](#) (可選)
5. [為每個隊列配置閾值級別](#) (可選)

註：除步驟1外，以上每個步驟都是可選的。您可以決定保留一個或多個引數的預設值。

## [第1步：啟用QoS](#)

首先，啟用QoS。請記住，預設情況下禁用QoS。禁用QoS時，您配置的CoS對映不會影響結果。有一個隊列以先進先出(FIFO)方式提供服務，所有資料包都在此丟棄。

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#mls qos
```

### **QoS is enabled globally**

Microflow policing is enabled globally

### QoS global counters:

```
Total packets: 552638
IP shortcut packets: 0
Packets dropped by policing: 0
IP packets with TOS changed by policing: 0
IP packets with COS changed by policing: 0
Non-IP packets with CoS changed by policing: 0
```

## [第2步：將每個可能的CoS值對映到隊列和閾值](#)

對於所有隊列型別，將CoS分配到隊列和閾值。為2q2t型別的埠定義的對映未應用於任何1p2q2t埠。此外，2q2t的對映應用於具有2q2t排隊機制的埠。在介面下發出以下**cos-map**命令：

```
wrr-queue cos-map Q_number_(1-2) threshold_number_(1-2) cos_value_1 cos_value_2
priority-queue cos-map Q_number_(always 1) cos_value_1 cos_value_2
```

注意：這些命令中的每一個應該在一行上。

您可以單獨配置WRR隊列。如果有優先順序隊列，可以使用**priority-queue**命令進行配置。

註：隊列總是從最低優先順序隊列開始編號，最後以可用的嚴格優先順序隊列結束。例如：

- 隊列1是低優先順序WRR隊列。
- 隊列2是高優先順序WRR隊列。
- 隊列3是嚴格優先順序隊列。

對所有型別的隊列重複此操作，否則將保留預設的CoS分配。以下是1p2q2t的配置示例：

```
cosmos#configure terminal
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#priority-queue cos-map 1 5
!--- Assign a CoS of 5 to priority queue. cos-map configured on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 1 1 0 1
!--- Assign CoS 0 and 1 to the first threshold of low-priority WRR queue. cos-map configured on:
```

```
Gil/1 Gil/2 cosmos(config-if)#wrr-queue cos-map 1 2 2 3
!--- Assign CoS 2 and 3 to the second threshold of low-priority WRR queue. cos-map configured
on: Gil/1 Gil/2 cosmos(config-if)#wrr-queue cos-map 2 1 4 6
!--- Assign CoS 4 and 6 to the first threshold of high-priority WRR queue. cos-map configured
on: Gil/1 Gil/2 cosmos(config-if)#wrr-queue cos-map 2 2 7
!--- Assign CoS 7 to the first threshold of high-priority WRR queue. cos-map configured on:
Gil/1 Gil/2
```

### 檢查設定：

```
cosmos#show queueing interface gigabitethernet 1/1
!--- Output suppressed. queue thresh cos-map ----- 1 1 0 1 1 2
2 3 2 1 4 6 2 2 7 3 1 5 !--- Output suppressed.
```

### 步驟3:配置WRR重量

為兩個WRR隊列配置WRR權重。發出以下介面命令：

```
wrr-queue bandwidth weight_for_Q1 weight_for_Q2
```

權重1與隊列1有關，該隊列應該是低優先順序WRR隊列。請始終將此重量保持低於重量2的一個級別。此重量可以採用介於1和255之間的任何值。使用以下公式分配百分比：

- 排隊1 -  $[\text{重量1} / (\text{重量1} + \text{重量2})]$
- 排隊2 -  $[\text{重量2} / (\text{重量1} + \text{重量2})]$

您必須為所有型別的隊列定義權重。這些權重型別不必相同。以下是2q2t的一個示例，其中20%的時間服務隊列1,80%的時間服務隊列2:

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue bandwidth ?
<1-255> enter bandwidth weight between 1 and 255
cosmos(config-if)#wrr-queue bandwidth 20 80
!--- Queue 1 is served 20% of the time, and queue 2 is served !--- 80% of the time.
cosmos(config-if)#
```

### 檢查設定：

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 1p2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low    2
2             WRR high   2
3             Priority   1

WRR bandwidth ratios:  20[queue 1]  80[queue 2]
queue-limit ratios:    90[queue 1]  5[queue 2]
!--- Output suppressed.
```

注意：如果無法使用CatOS軟體，您可以為每個介面配置不同的WRR權重。

## 第4步：配置分配給每個隊列的緩衝區

您必須定義傳輸隊列比率。這將確定緩衝區在不同隊列中的拆分方式。

```
wrr-queue queue-limit percentage_WRR_Q1 percentage_WRR_Q2
cosmos(config)#interface gigabitethernet 1/2
cosmos(config-if)#wrr-queue queue-limit 70 15
!--- Queue 1 has 70% of the buffers. !--- Queues 2 and 3 both have 15% of the buffers. queue-
limit configured on: Gi1/1 Gi1/2
```

**注意：**如果Gigabit埠的排隊能力是1p1q2t，則對於嚴格優先順序隊列和高優先順序WRR隊列，需要使用相同級別。由於硬體原因，這些級別不能不同。僅配置兩個WRR隊列的頻寬。如果存在高優先順序WRR隊列和嚴格優先順序隊列，則自動使用相同的值。

某些隊列型別沒有可調的隊列大小。例如1p3q1t，可在WS-X6548RJ45上使用。這些隊列型別是固定的，您無法對其進行修改。

### 檢查設定：

```
cosmos#show queueing interface gigabitethernet 1/2
Interface GigabitEthernet1/2 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1              WRR low     2
  2              WRR high    2
  3              Priority    1

WRR bandwidth ratios:      5[queue 1] 255[queue 2]
queue-limit ratios:       70[queue 1] 15[queue 2]
```

**注意：**最好為低優先順序WRR隊列保留緩衝區的最大部分。這是需要啟用其他緩衝的隊列。其他隊列的優先順序更高。

## 第5步：為每個隊列配置閾值級別

最後一步是為WRED隊列或尾部丟棄隊列配置閾值級別。此清單提供以下命令：

- 對於使用WRED作為閾值丟棄機制的隊列，發出以下命令：

```
wrr-queue random-detect min-threshold Q_number threshold_1_value threshold_2_value
wrr-queue random-detect max-threshold Q_number threshold_1_value threshold_2_value
```

**注意：**這些命令中的每一個應該在一行上。

- 對於使用尾部丟棄作為丟棄機制的隊列，請發出以下命令：

```
wrr-queue threshold Q_number threshold_1_value threshold_2_value
```

**注意：**此命令應該在一行上。

WRED隊列的配置：

```

cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue random-detect min-threshold 1 20 50
!--- This sets the threshold of queue 1 to 20 and 50% minimum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect min-threshold 2 20 50
!--- This sets the threshold of queue 2 to 20 and 50% minimum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 1 50 80
!--- This sets the threshold of queue 1 to 50 and 80% maximum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 2 40 60
!--- This sets the threshold of queue 2 to 49 and 60% maximum threshold !--- configured on Gi1/1
Gi1/2.

```

## 尾部丟棄隊列的配置：

```

cosmos(config)#interface fastethernet 3/1
cosmos(config-if)#wrr-queue threshold ?
<1-2> enter threshold queue id (1-2)
cosmos(config-if)#wrr-queue threshold 1 ?
<1-100> enter percent of queue size between 1 and 100
cosmos(config-if)#wrr-queue threshold 1 50 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 1 (low-priority)
to 50 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)# cosmos(config-if)# cosmos(config-if)#wrr-
queue threshold 2 40 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 2 (high-priority)
to 40 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)#

```

## 檢查設定：

```

cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = lp2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low     2
2             WRR high    2
3             Priority    1

WRR bandwidth ratios: 20[queue 1] 80[queue 2]
queue-limit ratios: 70[queue 1] 15[queue 2]

queue random-detect-min-thresholds
-----
1    20[1] 50[2]
2    20[1] 50[2]

queue random-detect-max-thresholds
-----
1    50[1] 80[2]
2    40[1] 60[2]

```

```

cosmos#show queueing interface fastethernet 3/1
Interface FastEthernet3/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 2q2t]:

```



```

Queue Id      Scheduling  Num of thresholds
-----
    1          WRR low           2
    2          WRR high           2

WRR bandwidth ratios: 100[queue 1] 255[queue 2]
queue-limit ratios:   90[queue 1] 10[queue 2]

queue tail-drop-thresholds
-----
 1      50[1] 100[2]
 2      40[1] 100[2]

```

您不能配置閾值並為每個埠的隊列分配CoS。所有變更都會套用到一組連續的連線埠：

- 千兆線卡的四個埠 — 埠1至4在一起，埠5至8在一起。
- 12個埠，用於10/100埠或100個光纖埠（基於1q4t/2q2t隊列） — 1到12、13到24、25到36和36到48。
- 若要判斷屬於同一ASIC的確切連線埠，請使用**show interface capabilities**命令。

## [監控輸出計畫和驗證配置](#)

若要驗證連線埠與輸出排程相關的目前運行時組態，發出的最簡單命令是**show queueing interface {gigabitethernet | fastethernet} slot/port**命令。此命令顯示埠上的隊列型別、CoS到不同隊列和閾值的對映、緩衝區共用和WRR權重。在這裡，隊列1的WRR為20%，隊列2的WRR為80%。該命令還顯示輸出排程的所有配置資訊以及每個閾值在每個隊列中丟棄的資料包數：

```

cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default COS is 0
Transmit queues [type = 1p2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
    1          WRR low           2
    2          WRR high           2
    3          Priority           1
WRR bandwidth ratios: 20[queue 1] 80[queue 2]
queue-limit ratios:  70[queue 1] 15[queue 2]

queue random-detect-max-thresholds
-----
 1      50[1] 80[2]
 2      40[1] 60[2]

queue thresh cos-map
-----
 1      1      0 1
 1      2      2 3
 2      1      4 6
 2      2      7
 3      1      5

Receive queues [type = 1plq4t]:
Queue Id      Scheduling  Num of thresholds
-----
    1          Standard           4

```

2 Priority 1

```
queue tail-drop-thresholds
-----
1    100[1] 100[2] 100[3] 100[4]

queue thresh cos-map
-----
1    1      0 1
1    2      2 3
1    3      4
1    4      6 7
2    1      5
```

Packets dropped on Transmit:

BPDU packets: 0

```
queue thresh    dropped [cos-map]
-----
1    1          0 [0 1 ]
1    2          0 [2 3 ]
2    1          0 [4 6 ]
2    2          0 [7 ]
3    1          0 [5 ]
```

Packets dropped on Receive:

BPDU packets: 0

```
queue thresh    dropped [cos-map]
-----
1    1          0 [0 1 ]
1    2          0 [2 3 ]
1    3          0 [4 ]
1    4          0 [6 7 ]
2    1          0 [5 ]
```

## 輸出排程示例

此流量將注入Catalyst 6500/6000上：

- 在埠gigabit 1/2中：1 gigabit的流量，優先順序為零
- 在埠gigabit 5/2中：133MB流量，優先順序為7133MB流量，優先順序為6133MB流量，優先順序為5133MB流量，優先順序為4133MB流量，優先順序為3133MB流量，優先順序為2133MB流量，優先順序為1

所有單點傳播流量都會以每個連線埠gigabit 1/1離開交換器，這是非常超額訂閱的流量。

## 案例1:已啟用QoS，但使用預設引數

**show queueing interface gigabitethernet 1/1**命令會設定此範例中的所有輸出。該命令提供有關輸入排程的其他資訊。但是，由於本文檔僅涉及輸出排程，因此會抑制該輸出。

當全域性啟用QoS且所有預設引數都在使用中時，幾分鐘後就會出現以下輸出：

```
nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
```

```

Trust state: trust DSCP
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
    1          WRR low           2
    2          WRR high          2
    3          Priority          1

WRR bandwidth ratios: 100[queue 1] 255[queue 2]
queue-limit ratios:   90[queue 1]   5[queue 2]

queue random-detect-max-thresholds
-----
  1    40[1] 100[2]
  2    40[1] 100[2]

queue thresh cos-map
-----
  1    1    0 1
  1    2    2 3
  2    1    4
  2    2    6 7
  3    1    5

Packets dropped on Transmit:
  BPDU packets: 0

queue thresh      dropped  [cos-map]
-----
  1    1          149606424  [0 1 ]
  1    2              0  [2 3 ]
  2    1          16551394  [4 ]
  2    2          4254446   [6 7 ]
  3    1              0  [5 ]

```

在此輸出中，預設值為：

- 隊列1的WRR權重 —  $100 / (100 + 255) = 28\%$
- 隊列2的WRR權重 —  $255 / (255 + 100) = 72\%$
- 緩衝區共用：— 對於隊列1,90%；對於隊列2,5%；對於嚴格優先順序隊列，5%

低優先順序WRR隊列中的大多數資料包被丟棄，但兩個閾值的高優先順序WRR隊列中的某些資料包仍被丟棄。一共有170,412,264個丟包(149,606,424 + 16,551,394 + 4,254,446)。這些丟棄按如下方式拆分：

- $149,606,424 / 170,412,264 =$  隊列1中丟棄數的88% ( 具有CoS 0和1的第一個閾值資料包 )
- $16,551,394 / 170,412,264 =$  隊列2中丟棄數的10% ( 具有CoS 4的第一個閾值資料包 )
- $4,254,446 / 170,412,264 =$  隊列2中丟棄數的2% ( CoS為6或7的第二個閾值資料包 )

注意：在嚴格優先順序隊列中看不到任何丟棄。

## 案例2:修改WRR權重

如案例1所述：「[QoS已啟用](#)」和「[使用預設引數](#)」部分，隊列2中的資料包仍被丟棄。修改WRR權重，為隊列2提供更多的頻寬。現在，隊列1被清空的時間為4%，隊列2被清空的時間為96%：

```

show run interface gigabitethernet 1/1
interface GigabitEthernet1/1

```

```

no ip address
wrr-queue bandwidth 10 255
mls qos trust dscp
switchport
switchport mode access
end

```

```

nelix#show queueing interface gigabitethernet 1/1

```

```

Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin

```

```

Port QoS is enabled

```

```

Trust state: trust DSCP

```

```

Default cos is 0

```

```

Transmit queues [type = lp2q2t]:

```

Queue Id	Scheduling	Num of thresholds
1	WRR low	2
2	WRR high	2
3	Priority	1

```

WRR bandwidth ratios: 10[queue 1] 255[queue 2]
queue-limit ratios: 90[queue 1] 5[queue 2]

```

```

queue random-detect-max-thresholds

```

1	40[1]	100[2]
2	40[1]	100[2]

```

queue thresh cos-map

```

1	1	0 1
1	2	2 3
2	1	4
2	2	6 7
3	1	5

```

Packets dropped on Transmit:

```

```

BPDU packets: 0

```

```

queue thresh dropped [cos-map]

```

1	1	2786205	[0 1 ]
1	2	0	[2 3 ]
2	1	11363	[4 ]
2	2	69	[6 7 ]
3	1	0	[5 ]

如本輸出所示，隊列2中的丟棄百分比現在低得多。總共2,797,637個丟棄以這種方式分割：

- $2,786,205 / 2,797,637 = 99.591\%$  丟棄在隊列1中 ( 資料包為CoS 0和1 )
- $11,363 / 2,797,637 =$  隊列2中丟棄數的0.408% ( 使用資料包CoS 4的第一個閾值 )
- $69 / 2,797,637 =$  隊列2中丟棄數的0.001% ( 具有CoS 6和7的資料包的第二個閾值 )

如果您使用各種WRR權重，它可確保隊列2中有更多的QoS。

### 案例3:其他WRR權重修改

使用WRR權重可以更加積極。在此示例輸出中，只有0.39%的權重賦予隊列1:

```

show run interface gigabitethernet 1/1
interface GigabitEthernet1/1

```

```

no ip address
wrr-queue bandwidth 1 255
mls qos trust dscp
switchport
switchport mode access
end

```

```

nelix#show queueing interface gigabitethernet 1/1

```

```

Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin

```

```

Port QoS is enabled

```

```

Trust state: trust DSCP

```

```

Default cos is 0

```

```

Transmit queues [type = lp2q2t]:

```

Queue Id	Scheduling	Num of thresholds
1	WRR low	2
2	WRR high	2
3	Priority	1

```

WRR bandwidth ratios: 1[queue 1] 255[queue 2]
queue-limit ratios: 90[queue 1] 5[queue 2]

```

```

queue random-detect-max-thresholds

```

1	40[1]	100[2]
2	40[1]	100[2]

```

queue thresh cos-map

```

1	1	0 1
1	2	2 3
2	1	4
2	2	6 7
3	1	5

```

Packets dropped on Transmit:

```

```

BPDU packets: 0

```

```

queue thresh dropped [cos-map]

```

1	1	2535315	[0 1]
1	2	0	[2 3]
2	1	705	[4]
2	2	73	[6 7]
3	1	0	[5]

即使具有主動的WRR權重，隊列2中的資料包仍然被丟棄。但是，相比較而言，資料包並不多。現在，隊列2中只有0.03%的資料包丟棄。

#### 案例4:修改隊列限制緩衝區分配

如案例2所示：[修改WRR重量](#)和[案例3:在其他WRR權重修改部分](#)，資料包仍在隊列2中丟棄，儘管WRR百分比可以確保丟棄的數量是最小的。但是，當隊列2達到第二個閾值（設定為100%）時，某些資料包會繼續被丟棄。

為了改善這一點，請更改隊列限制（分配給每個隊列的緩衝區大小）。在本示例中，將隊列1的隊列限制設定為70%，隊列2的隊列限制設定為15%，嚴格優先順序隊列的隊列限制設定為15%：

```

show run gigabitethernet 1/1

```

```
interface GigabitEthernet1/1
no ip address
wrr-queue bandwidth 1 255
wrr-queue queue-limit 70 15
mls qos trust dscp
switchport
switchport mode access
end
```

nelix#show queueing interface gigabitethernet 1/1

Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin

Port QoS is enabled

Trust state: trust DSCP

Default cos is 0

Transmit queues [type = lp2q2t]:

Queue Id	Scheduling	Num of thresholds
1	WRR low	2
2	WRR high	2
3	Priority	1

WRR bandwidth ratios: 1[queue 1] 255[queue 2]  
queue-limit ratios: 70[queue 1] 15[queue 2]

queue random-detect-max-thresholds

1	40[1]	100[2]
2	40[1]	100[2]

queue thresh cos-map

1	1	0 1
1	2	2 3
2	1	4
2	2	6 7
3	1	5

Receive queues [type = lp1q4t]:

Queue Id	Scheduling	Num of thresholds
1	Standard	4
2	Priority	1

queue tail-drop-thresholds

1	100[1]	100[2]	100[3]	100[4]
---	--------	--------	--------	--------

queue thresh cos-map

1	1	0 1
1	2	2 3
1	3	4
1	4	6 7
2	1	5

Packets dropped on Transmit:

BPDU packets: 0

queue thresh dropped [cos-map]

1	1	154253046	[0 1]
1	2	0	[2 3]
2	1	0	[4]
2	2	0	[6 7]

現在，丟棄僅發生在隊列1中。

## 使用輸出排程減少延遲和抖動

**輸出排程示例**一節中的案例研究演示了實施輸出排程的好處，以便在輸出埠超訂用時避免VoIP或任務關鍵型流量的下降。超訂用在正常網路中（尤其是千兆鏈路）不會非常頻繁。超訂用只能在流量高峰期或在極短時間內發生流量突發期間發生。

即使沒有任何超訂用，在端到端實施QoS的網路中，輸出排程也可以有很大的幫助。本節提供輸出排程如何幫助減少延遲和抖動的示例。

### 減少延遲

封包延遲會增加，因為每個交換器的緩衝區在等待傳輸時「遺失」時間。例如，在大型備份或檔案傳輸期間，會將CoS為5的小型語音資料包從埠傳送出去。假設輸出埠沒有QoS，並且小型語音資料包在10個大型1500位元組資料包後排隊。在這種情況下，您可以輕鬆計算傳輸10個大型資料包所需的千兆速度時間為：

- $(10 \times 1500 \times 8) = 120,000$ 位，在120微秒內傳輸

如果此資料包在通過網路時需要經過八九台交換機，則可能導致大約1毫秒的延遲。這僅包括網路中交叉的交換機的輸出隊列中的延遲。

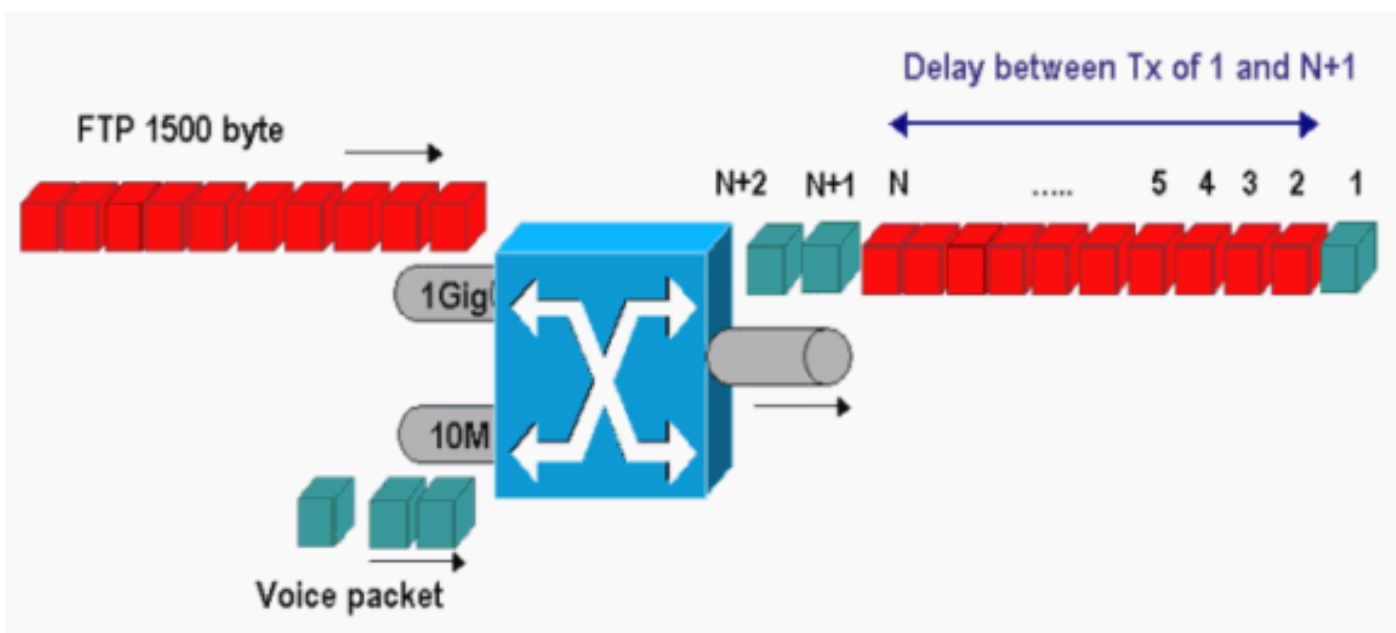
**注意：**如果需要在10MB介面（例如，連線到IP電話和PC）上對相同的10個大型資料包進行排隊，則延遲引入如下：

- $(10 \times 1500 \times 8) = 120,000$ 位，在12毫秒內傳輸

輸出排程的實現確保將CoS為5的語音資料包放入嚴格優先順序隊列中，並在CoS小於5的任何資料包之前傳送。這減少了延遲。

### 減少抖動

輸出排程的另一個重要優點是減少了抖動。抖動是同一資料流中資料包延遲的變化。此示例場景顯示輸出排程如何減少抖動：



在此案例中，同一個輸出連線埠需要傳送兩個流：

- 10MB乙太網埠上的一個傳入語音流。
- 1 Gigabit乙太網路上行鏈路連線埠上的一個傳入FTP流。

兩個流通過相同的輸出埠離開交換機。此示例顯示了不使用輸出排程時會發生的情況。所有大型資料包都可以在兩個語音資料包之間交錯。這會在接收來自同一流的語音資料包時產生抖動。當交換器傳輸大型資料封包時，封包1和封包 $n + 1$ 之間的接收延遲會更大。但是， $n + 1$ 和 $n + 2$ 之間的延遲可以忽略。這會導致語音流量流中的抖動。使用嚴格優先順序隊列可以很容易避免此問題。確保將語音資料包的CoS值對映到嚴格優先順序隊列。

## 結論

在本文中，您已經看到如何對執行Cisco IOS軟體的Catalyst 6500/6000上的輸出佇列排程進行設定和疑難排解的範例研究。在大多數具有語音流量的網路中，您還看到了輸出排程的優勢：

- 避免在輸出埠超訂用時丟棄關鍵流量。
- 減少延遲。
- 減少抖動。

## 相關資訊

- [執行CatOS系統軟體的Catalyst 6500/6000系列交換器上的QoS輸出排程](#)
- [瞭解Catalyst 6000系列交換器上的服務品質](#)
- [LAN 產品支援頁面](#)
- [LAN 交換支援頁面](#)
- [技術支援與文件 - Cisco Systems](#)