

Cisco IOS システム ソフトウェアが稼働している Catalyst 6500/6000 シリーズ スイッチの QoS 出カスケジューリング

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[出力キューでの廃棄](#)

[Catalyst 6500/6000 のさまざまなラインカードの出力キューイング機能](#)

[ポートのキューの機能について](#)

[Catalyst 6500/6000 の設定、監視、および出カスケジューリング サンプル](#)

[コンフィギュレーション](#)

[出カスケジューリングの監視と設定の確認](#)

[出カスケジューリングの例](#)

[出カスケジューリングを使用した遅延とジッタの削減](#)

[遅延の削減](#)

[ジッタの削減](#)

[結論](#)

[関連情報](#)

概要

出カスケジューリングを使用すると、加入超過が過剰に発生した場合に重要なトラフィックが廃棄されなくなります。このドキュメントは、Catalyst 6500/6000 スイッチでの出カスケジューリングに関連するすべてのテクニックとアルゴリズムについて包括的に説明します。また、Cisco IOS® ソフトウェアが稼働している Catalyst 6500/6000 での出カスケジューリングの動作を設定して検証する方法についても説明します。

Weighted Random Early Detection (WRED; 重み付けランダム早期検出)、Weighted Round Robin (WRR; 重み付けラウンドロビン)、テールドロップの詳細は、『[CatOS システム ソフトウェアが稼働している Catalyst 6500/6000 シリーズ スイッチの QoS 出カスケジューリング](#)』を参照してください。

前提条件

要件

このドキュメントに特有の要件はありません。

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

背景説明

出力キューでの廃棄

出力キューでの廃棄はインターフェイスの輻輳によって発生します。通常、この原因としては、高帯域幅リンクからのトラフィックが低帯域幅リンクにスイッチングされるか、あるいは複数の着信リンクからのトラフィックが単一の発信リンクにスイッチングされることが考えられます。

たとえば、バーストトラフィックが大量にギガビットインターフェイスに流れ込み、それが 100 Mbps インターフェイスにスイッチングされると、100 Mbps インターフェイスの出力廃棄が増加する場合があります。これは、着信側の帯域幅と発信側の帯域幅との間で速度のミスマッチがあるため、インターフェイスの出力キューが過度のトラフィックによってあふれるためです。発信インターフェイスのトラフィックレートでは、発信する必要があるすべてのパケットを受け入れられません。

この問題の最適の解決方法は、回線速度を上げることです。ただし、回線速度を上げたくない場合に、出力の廃棄を防止、減少、または制御する方法があります。出力廃棄の防止は、短時間のデータバーストの結果として出力廃棄が発生している場合にだけ可能です。恒常的な高レートのフローによって出力の廃棄が引き起こされている場合、廃棄を防ぐことはできません。ただし、出力廃棄を制御することはできます。

Catalyst 6500/6000 のさまざまなラインカードの出力キューイング機能

ポートのキューイング機能がはっきりわからない場合は、`show queueing interface {gigabitethernet | fastethernet} mod/port` コマンドを発行します。次に示すのは、`show queueing interface` コマンド出力の始めの部分です。ポートは Supervisor Engine 1A ラインカード上にあります。

```
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 = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
-----
```

```

1          WRR low          2
2          WRR high        2
3          Priority         1

```

!--- Output suppressed.

出力は、このポートに 1p2q2t という出力キューイング タイプがあることを示しています。

特定のポートで利用可能なキューイングのタイプを確認するもう 1 つの方法は、**show interface capabilities** コマンドを発行することです。

```

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

```

ポートのキューの機能について

Catalyst 6500/6000 スイッチでは、いくつかのタイプのキューを利用できます。次の表には、ポート QoS アーキテクチャの表記法が説明されています。

送信 (Tx)/受信 (Rx)	キュー表記	キューの数	優先キュー	WRR キューの数	WRR キューのしきい値のタイプと数
Tx の後に引き継ぐ	2q2t	0	—	0	2 つの設定可能なテールドロップ
Tx の後に引き継ぐ	1p2q2t	3	1	0	2 つの設定可能な WRED
Tx の後に引き継ぐ	1p3q1t	4	1	3	1 つの設定可能な WRED
Tx の後に引き継ぐ	1p2q1t	3	1	0	1 つの設定可能な WRED
Rx	1q4t	1	—	—	4 つの設定可能なテールドロップ
Rx	1p1q4t	0	1	1	4 つの設定可能なテールドロップ

					プ
Rx	1p1q0t	0	1	1	設定不可
Rx	1p1q8t	0	1	1	8つの設定可能な WRED
Tx の後に引き継ぐ	1p3q8t	4	1	3	8つの設定可能な WRED または テールドロップ
Tx の後に引き継ぐ	1p7q8t	8	1	7	8つの設定可能な WRED または テールドロップ
Rx	1q2t	1	—	—	1つの設定可能な テールドロップ = 1つの設定不可
Rx	1q8t	1	—	—	8つの設定可能な テールドロップ
Rx	2q8t	0	—	0	8つの設定可能な テールドロップ

次の表には、いくつかのモジュールと、インターフェイスやポートの Rx 側と Tx 側のキュータイプがリストされています。使用しているモジュールがこのリストにない場合、**show interface capabilities** コマンドを使用して利用可能なキューの機能を確認してください。**show interface capabilities** コマンドの詳細は、「[Catalyst 6500/6000 のさまざまなラインカードの出力キューイング機能](#)」セクションを参照してください。

モジュール	Rx キュー	Tx キュー
WS-X6K-S2-PFC2	1p1q4t	1p2q2t
WS-X6K-SUP1A-2GE	1p1q4t	1p2q2t
WS-X6K-SUP1-2GE	1q4t	2q2t
WS-X6501-10GEX4	1p1q8t	1p2q1t
WS-X6502-10GE	1p1q8t	1p2q1t
WS-X6516-GBIC	1p1q4t	1p2q2t
WS-X6516-GE-TX	1p1q4t	1p2q2t
WS-X6416-GBIC	1p1q4t	1p2q2t
WS-X6416-GE-MT	1p1q4t	1p2q2t
WS-X6316-GE-TX	1p1q4t	1p2q2t
WS-X6408A-GBIC	1p1q4t	1p2q2t
WS-X6408-GBIC	1q4t	2q2t
WS-X6524-100FX-MM	1p1q0t	1p3q1t
WS-X6324-100FX-SM	1q4t	2q2t
WS-X6324-100FX-MM	1q4t	2q2t
WS-X6224-100FX-MT	1q4t	2q2t

WS-X6548-RJ-21	1p1q0t	1p3q1t
WS-X6548-RJ-45	1p1q0t	1p3q1t
WS-X6348-RJ-21	1q4t	2q2t
WS-X6348-RJ21V	1q4t	2q2t
WS-X6348-RJ-45	1q4t	2q2t
WS-X6348-RJ-45V	1q4t	2q2t
WS-X6148-RJ-45V	1q4t	2q2t
WS-X6148-RJ21V	1q4t	2q2t
WS-X6248-RJ-45	1q4t	2q2t
WS-X6248A-TEL	1q4t	2q2t
WS-X6248-TEL	1q4t	2q2t
WS-X6024-10FL-MT	1q4t	2q2t

Catalyst 6500/6000 の設定、監視、および出カスケジューリングサンプル

コンフィギュレーション

このセクションでは、Cisco IOS ソフトウェアが稼働している Catalyst 6500/6000 で出カスケジューリングを設定するために必要なすべての手順について説明します。Catalyst 6500/6000 のデフォルト設定については、このドキュメントの「[ケース 1: QoS がイネーブルになり、デフォルトパラメータが使用される](#)」セクションを参照してください。

Catalyst 6500/6000 の設定には、次の 5 つのステップがあります。

1. [qos を有効にします。](#)
2. [それぞれの可能な Class of Service \(CoS; サービス クラス \) 値をキューとしきい値にマップする \(オプション \)](#)
3. [WRR の重み付けを設定する \(オプション \)](#)
4. [それぞれのキューに割り当てられるバッファを設定する \(オプション \)](#)
5. [それぞれのキューにしきい値レベルを設定する \(オプション \)](#)

注意: これらの各ステップは、ステップ1を除いてオプションです。1つ以上のパラメータにデフォルト値を残すことができます。

ステップ 1: qos を有効にします。

まず、QoS をイネーブルにします。QoS はデフォルトでは無効になっていることに注意してください。QoS がディセーブルのときは、設定済みの CoS マッピングは結果に影響しません。First In First Out (FIFO; 先入れ先出し) 方式のキューが 1 つあり、すべてのパケットはそこで廃棄されます。

```
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
```

ステップ 2 : それぞれの可能な Class of Service (CoS; サービス クラス) 値をキューとしきい値にマップする

すべてのキュー タイプに対して、キューとしきい値に CoS を割り当てます。2q2t タイプのポートに対して定義されるマッピングは、1p2q2t ポートには適用されません。また、2q2t 用のマッピングは、2q2t のキューイング メカニズムを備えたすべてのポートに適用されます。次の `cos-map` コマンドをインターフェイスの下で発行します。

```
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 : これらの各コマンドは1行で記述する必要があります。

WRR キューは独立して設定できます。優先キューが存在する場合は、`priority-queue` コマンドで設定できます。

注 : キューには、可能な最も低いプライオリティキューから始まり、使用可能な完全優先キューまで、常に番号が付けられます。以下に、いくつかの例を示します。

- queue 1 は、低優先度 WRR キューです。
- queue 2 は、高優先度 WRR キューです。
- queue 3 は、完全優先キューです。

この操作をすべてのタイプのキューで繰り返します。操作を行わない場合は、デフォルトの CoS 割り当てのままになります。次に示すのは、1p2q2t の設定例です。

```
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: Gi1/1 Gi1/2 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: Gi1/1 Gi1/2 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: Gi1/1 Gi1/2 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: Gi1/1 Gi1/2
```

設定の確認 :

```
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.
```

ステップ 3 : WRR の重み付けを設定する

WRRの重み付けを2つのWRRキューに設定します。次のインターフェイスコマンドを発行します。

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

weight 1はqueue 1に関連します。このキューは低優先度のWRRキューである必要があります。この重量は、常に重量2より1レベル低く保ちます。重量には、1～255の任意の値を使用できます。次の式を使用して、パーセンテージを割り当てます。

- queue 1に対して： $[\text{weight } 1 / (\text{weight } 1 + \text{weight } 2)]$
- queue 2へ－ $[\text{weight } 2 / (\text{weight } 1 + \text{weight } 2)]$

すべてのタイプのキューに対して、重み付けを定義する必要があります。これらの重量のタイプは同じである必要はありません。これは2q2t用の例であり、この場合、queue 1には時間の20%が使用され、queue 2には時間の80%が使用されます。

```
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)#
```

設定の確認：

```
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:    90[queue 1]  5[queue 2]
!--- Output suppressed.
```

注：CatOSソフトウェアを使用できない場合は、インターフェイスごとに異なるWRR重み付けを設定できます。

ステップ4：それぞれのキューに割り当てられるバッファを設定する

転送キュー率を定義する必要があります。これによって、さまざまなキューでのバッファの分割方法が決定されます。

```
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：ギガビットポートのキューイング機能が1p1q2tの場合は、完全優先キューと高優先WRRキューに同じレベルを使用する必要があります。これらのレベルはハードウェアのために異なるものにはできません。2つのWRRキュー用の帯域幅だけが設定されます。高優先度WRRキューと完全優先キューが存在する場合、それらには自動的に同じ値を使用します。

調整可能なキューサイズのないキューイングタイプもあります。たとえば、1p3q1tはWS-X6548RJ45で使用できます。これらのキュータイプは固定されており、変更できません。

設定の確認：

```
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：低優先度WRRキューのバッファの最大部分を残しておくことをお勧めします。これが、追加のバッファリングをイネーブルにする必要のあるキューです。他のキューは高い優先順位で使用されます。

ステップ5：それぞれのキューにしきい値レベルを設定する

最後のステップとして、WRED キューまたはテール ドロップ キューにしきい値レベルを設定します。次にコマンドを示します。

- しきい値の廃棄メカニズムとして WRED を使用するキューに対しては、次のコマンドを発行します。

```
wrr-queue random-dtect min-threshold Q_number threshold_1_value threshold_2_value
```

```
wrr-queue random-dtect max-threshold Q_number threshold_1_value threshold_2_value
```

注：これらの各コマンドは1行で記述する必要があります。

- 廃棄メカニズムとしてテール ドロップを使用するキューに対しては、次のコマンドを発行します。

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

注：このコマンドは1行で記述する必要があります。

WRED キューの設定：

```
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.

```

テールドロップキューの設定：

```

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)#

```

設定の確認：

```

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-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]
```

しきい値を設定して、ポートごとのキューに CoS を割り当てることはできません。すべての変更は一連の連続したポートに適用されます。

- ギガビット ラインカード用の 4 つのポート：ポート 1 から 4 までがまとめられ、ポート 5 から 8 までがまとめられます。
- 10/100 ポート用の 12 個のポートまたは 1q4t/2q2t キューイングをベースにした 100 ファイバポート：1 から 12 まで、13 から 24 まで、25 から 36 まで、36 から 48 まで。
- 同じ ASIC に属しているポートを判断するには、**show interface capabilities** コマンドを使用します。

出カスケジューリングの監視と設定の確認

出カスケジューリングを考慮してポートに対する現在のランタイム設定を確認するために発行する最も簡単なコマンドは、**show queueing interface {gigabitethernet | fastethernet} slot/port** コマンドを発行します。このコマンドによって、ポートのキューイングのタイプ、さまざまなキューやしきい値への CoS のマッピング、バッファ共有、WRR の重み付けが表示されます。このコマンドは、キュー1のWRRが20%、キュー2のWRRが80%です。また、出カスケジューリング用に設定されたすべての情報と、各しきい値の各キューで廃棄されたパケット数も表示します。

```
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 = 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:
BPDUs 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:
BPDUs 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 ]
```

出カスケジューリングの例

次のトラフィックは Catalyst 6500/6000 にインジェクトされます。

- ポート ギガビット 1/2 内：優先順位 0 の 1 ギガビット トラフィック
 - ポート ギガビット 5/2へ：優先順位 7 の 133MB トラフィック優先順位 6 の 133MB トラフィック優先順位 5 の 133MB トラフィック優先順位 4 の 133MB トラフィック優先順位 3 の 133MB トラフィック優先順位 2 の 133MB トラフィック優先順位 1 の 133MB トラフィック
- すべてのユニキャストトラフィックは、ポート ギガビット 1/1 ごとでスイッチに存在し、かなりの加入超過になります。

ケース 1：QoS ガイネーブルになり、デフォルト パラメータが使用される

`show queueing interface gigabitethernet 1/1` コマンドは、この例ですべての出力を設定しています。このコマンドは、入カスケジューリングについての追加情報を提供します。ただし、このドキュメントは出カスケジューリングだけを対象にしているため、その出力は示していません。

QoS がグローバルにイネーブルになり、すべてのデフォルト パラメータが使用中である場合、2、3 分後に次の出力が生成されます。

```
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 ]
```

この出力で、デフォルト値は次のようになります。

- queue 1 用の WRR の重み付け : $100 / (100 + 255) = 28\%$
- キュー2のWRRの重み付け : $255 / (255 + 100) = 72\%$
- バッファ共有 : queue 1 に対しては 90%、queue 2 に対しては 5%、完全優先キューに対しては 5%

低優先度 WRR キューのほとんどのパケットは廃棄されますが、両方のしきい値に対して高優先度 WRR キューでもいくつかは廃棄されます。合計で 170,412,264 の廃棄 (149,606,424 + 16,551,394 + 4,254,446) があります。これらの廃棄は次のように分割されます。

- queue 1 の $149,606,424 / 170,412,264 = 88\%$ のドロップ (CoS 0 および 1 の 1 番目のしきい値パケット)
- queue 2 の $16,551,394 / 170,412,264 = 10\%$ のドロップ (CoS 4 の 1 番目のしきい値パケット)
- queue 2 の $4,254,446 / 170,412,264 = 2\%$ のドロップ (CoS 6 または 7 の 2 番目のしきい値パケット)

注：完全優先キューにドロップは表示されません。

ケース 2：WRR の重み付けの変更

「[ケース 1：QoS がイネーブルになり、デフォルト パラメータが使用される](#)」セクションでも説明したように、queue 2 のパケットは引き続き廃棄されます。キュー2により多くの帯域幅を提供するようにWRRの重み付けを変更します。ここで、キュー1は時間の4%を空にし、キュー2は時間の96%を空にします。

```
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 = 1p2q2t]:
  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 ]
```

この出力でわかるように、queue 2 の廃棄の割合はかなり低くなっています。合計で 2,797,637 の廃棄が次のように分割されています。

- queue 1 の $2,786,205 / 2,797,637 = 99.591\%$ のドロップ (CoS 0 および 1 のパケットで)
- queue 2 の $11,363 / 2,797,637 = 0.408\%$ のドロップ (パケット CoS 4 の 1 番目のしきい値)
- queue 2 の $69 / 2,797,637 = 0.001\%$ のドロップ (CoS 6 および 7 のパケットの 2 番目のしきい値)

さまざまな WRR の重み付けを使用する場合、queue 2 の QoS はさらに多くなります。

ケース 3 : その他の WRR の重み付けの変更

さらにアグレッシブな WRR の重み付けをすることもできます。次の出力例では、重み付けの 0.39% だけが queue 1 に提供されています。

```
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
```

Queue Id	Threshold 1	Threshold 2
1	40[1]	100[2]
2	40[1]	100[2]

```
queue thresh cos-map
```

Queue Id	Thresh Id	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]
```

Queue Id	Thresh Id	Dropped	CoS-Map
1	1	2535315	[0 1]
1	2	0	[2 3]
2	1	705	[4]
2	2	73	[6 7]

アグレッシブWRRの重み付けでも、パケットはキュー2で引き続き廃棄されます。ただし、それに比べてパケットの数は多くありません。queue 2 でのパケット廃棄は 0.03% だけになっています。

ケース 4：キュー制限バッファ割り当ての変更

「[ケース 2：WRR の重み付けの変更](#)」セクションおよび「[ケース 3：その他の WRR の重み付けの変更](#)」セクションで説明しているように、パケットは queue 2 に引き続き廃棄されていますが、WRR の割合で廃棄を最小限にすることが保証されています。ただし、queue 2 で 2 番目のしきい値 (100% に設定されるもの) に到達すると、いくつかのパケットは引き続き廃棄されます。

これを改善するために、キュー制限 (各キューに割り当てられるバッファのサイズ) を変更します。次の例で、キュー制限は queue 1 に対しては 70%、queue 2 に対しては 15%、完全優先キューに対しては 15% に設定されています。

```
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 = 1p2q2t]:
```

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
```

Queue Id	Threshold 1	Threshold 2
1	40[1]	100[2]
2	40[1]	100[2]

```
queue thresh cos-map
```

Queue Id	Priority	WRR Low	WRR High
1	1	0	1
1	2	2	3
2	1	4	
2	2	6	7
3	1	5	

```
Receive queues [type = 1p1q4t]:
```

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 ]
3      1              0  [5 ]
```

これで、廃棄は queue 1 だけで発生します。

出力スケジューリングを使用した遅延とジッタの削減

「[出力スケジューリングの例](#)」セクションで示している事例では、出力ポートで加入超過が発生したときに VoIP やミッションクリティカルなトラフィックの廃棄を避けるために、出力スケジューリングを実装するメリットを示しています。加入超過はそれほど頻繁にはありませんが、正常なネットワーク（特にギガビットリンク）で発生します。加入超過が発生するのは、トラフィックのピーク時またはごく短時間内に発生するトラフィックのバーストの間だけです。

加入超過がない場合でも、出力スケジューリングは、QoS がエンド ツー エンドで実装されたネットワークではきわめて有効です。このセクションでは、出力スケジューリングが遅延およびジッタを減らす方法についての例を紹介しています。

遅延の削減

パケットの遅延は、送信を待機しているときに各スイッチのバッファで「消失した」時間によって増加します。たとえば、CoS 5 の小さい音声パケットが、大きいバックアップかファイル転送の間にポートから送られます。出力ポートに QoS がなく、小さい音声パケットが 10 個の大きい 1500 バイトパケットの後でキューに入ると仮定します。この場合、次のように、10 個の大きいパケットの送信に必要なギガビット速度の時間を簡単に計算できます。

- $(10 \times 1500 \times 8) = 120,000$ ビットが 120 マイクロ秒の間に転送されます。

このパケットがネットワークを通過する間に 8 つか 9 つのスイッチを経由する必要がある場合、約 1 ミリ秒の遅延が生じる可能性があります。これでカウントされているのは、ネットワーク内で経路するスイッチの出力キューでの遅延だけです。

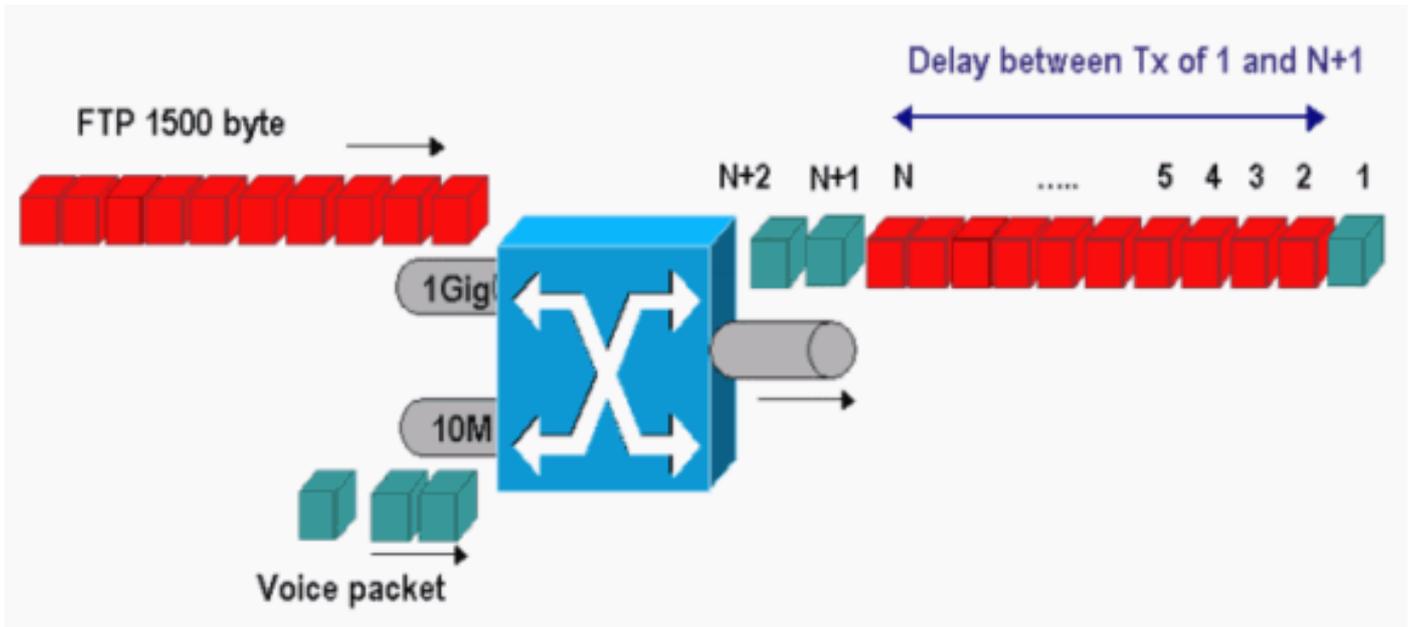
注：同じ 10 個の大きなパケットを 10MB のインターフェイス（たとえば、IP 電話や PC に接続されている）でキューイングする必要がある場合、遅延は次のようになります。

- $(10 \times 1500 \times 8) = 120,000$ ビットが 12 ミリ秒の間に転送されます。

出力スケジューリングの実装により、CoSが5の音声パッケージが完全優先キューに入れられ、CoSが5未満のパッケージの前に送信されます。これにより、遅延が減少します。

ジッタの削減

出力スケジューリングのもう 1 つの重要なメリットは、ジッタの削減です。ジッタとは、同じフロー内のパッケージの遅延がばらついていることです。次のサンプルのシナリオは、出力スケジューリングがジッタを削減できるしくみを示しています。



このシナリオでは、同じ出力ポートで 2 つのストリームを送信する必要があります。

- 1 つの入力音声ストリームは 10MB のイーサネット ポート上にあります。
 - 1 つの入力 FTP ストリームは 1 ギガビット イーサネット アップリンク ポート上にあります。
- どちらのストリームも、同じ出力ポートでスイッチから送信されます。この例では、出力スケジューリングを使用しない場合に何が発生するかが示されます。すべての大きなデータ パッケージは、2 つの音声パッケージの間でインターリーブされる可能性があります。これにより、同じストリームからの音声パッケージの受信でジッタが生成されます。スイッチから大きいデータ パッケージが送信されると、パッケージ 1 とパッケージ $n+1$ の受信の間には比較的大きな遅延が伴います。一方、 $n+1$ と $n+2$ の間の遅延は無視できる程度です。その結果、音声トラフィックストリームのジッタが発生します。完全優先キューを使用すると、この問題を容易に回避できます。音声パッケージの CoS 値を必ず完全優先キューにマップしてください。

結論

このドキュメントでは、Cisco IOS ソフトウェアが稼働している Catalyst 6500/6000 で出力キュースケジューリングを設定してトラブルシューティングする方法の事例を説明しました。また、音声トラフィックのあるほとんどのネットワークでの出力スケジューリングの次のメリットも確認しました。

- 出力ポートで加入超過が発生した場合、重要なトラフィックの廃棄を避ける。
- 遅延を削減する。
- ジッタを削減する。

関連情報

- [CatOS システム ソフトウェアが稼働している Catalyst 6500/6000 シリーズ スイッチの QoS 出カスケジューリング](#)
- [Catalyst 6000 ファミリ スイッチの Quality of Service \(QoS \) について](#)
- [LAN 製品に関するサポート ページ](#)
- [LAN スイッチングに関するサポート ページ](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)