

Catalyst 9000スイッチのキューバッファ割り当ての理解

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[関連製品](#)

[バックグラウンド情報](#)

[用語](#)

[バッファ](#)

[スイッチキューバッファ割り当てアルゴリズム](#)

[設定](#)

[シナリオ1: 特定のポート速度のベースバッファ値の判別](#)

[シナリオ2: 複数のキューのソフトバッファとハードバッファおよび明示的な比率の判別](#)

[シナリオ2.1: シナリオ2からユーザ設定の割増率を変更する](#)

[シナリオ3: 複数のキューのソフトおよびハードバッファと暗黙的な比率の判別](#)

[シナリオ3.1: 複数のキューと複数の暗黙的な比率に対するソフトバッファとハードバッファの決定](#)

[シナリオ4: キュー間のキューバッファ比率の合計が100未満](#)

[シナリオ5: プライオリティレベル2クラスのソフトバッファとハードバッファの判別](#)

[キュー制限](#)

[関連情報](#)

はじめに

このドキュメントでは、Catalyst 9000シリーズスイッチのトラフィックキューへのキューバッファ割り当てを予測する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- パケットのマーキング、キューイング、およびスケジューリングの方法に関するQuality of Service(QoS)の概念とプロセス
- Cisco MQC QoS設定

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco Catalyst 9500-25Y4C
- Cisco IOS® XE 17.4.1

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

関連製品

このドキュメントは、次のバージョンのハードウェアとソフトウェアにも使用できます。

- Catalyst 9200 - 9600シリーズスイッチ
- Catalyst 9300Xおよび9400X
- すべてのCisco IOS XEソフトウェアバージョン



注:9500Xまたは9600Xは異なるASICおよびQoSアーキテクチャを使用しているため、このドキュメントの内容は適用されません。

バックグラウンド情報

Catalyst 9000シリーズスイッチのQoSの技術概要については、『[Catalyst 9000 QoSとキューイング](#)』 [ホワイトペーパー](#)を参照してください。

多くの場合、バッファ割り当ては、特定のトラフィッククラスに対する望ましくない出力廃棄への応答として調整する必要があります。Catalyst 9000シリーズスイッチでの出力ドロップの診断とトラブルシューティングの詳細については、『[Catalyst 9000スイッチの出力ドロップのトラブル](#)』

[ルシューティング](#)」を参照してください。

用語

QoS	Quality of Service	ネットワークデバイスで送受信されるトラフィックの分類、マーキング、キューイング、およびスケジューリングに関連する関連機能の概念/グループ
DSCP	Differentiated Services Code Point (DiffServ コード ポイント)	パケットのIPヘッダーに含まれるトラフィック分類メカニズム
CoS	[Class of Service] >	パケットのイーサネットフレームヘッダーに含まれるトラフィック分類メカニズム
ACE	アクセスコントロールエントリ	アクセスコントロールリスト(ACL)内の単一のルールまたは行
ACL	アクセスコントロールリスト	トラフィックを照合してアクションを実行するために、さまざまな機能で使用されるアクセスコントロールエントリ(ACE)のグループ
ASIC	特定用途向け集積回路	特定のタスクまたはタスクのセットを高効率で実行するように設計されたコンピュータチップ。
UADP	ユニファイドアクセスデータプレーン	Catalyst 9000シリーズスイッチで多数のネットワークパケット処理タスクを実行するために使用されるCisco ASIC。
PBC	パケットバッファ複合体	Cisco UADP ASICサブシステム：パケットの処理、キューイング、スケジューリングを行うための中央パケットバッファとして機能します。
AQM	アクティブキューの管理	Cisco UADP ASICサブシステム：トラフィックキューを管理し、ネットワークポートのアクションをスケジューリングします。
DTS	動的しきい値とスケール	Cisco UADP ASICテクノロジー：ポート全体でバッファの動的な調整と拡張を行い、ハードウェア使用率を最適化

バッファ

概念として、バッファはデータの一時的なバーストを吸収するために使用されるメモリです。ポートにスイッチまたはルーティングされるデータが、データをネットワークに送出するポートの能力を超えた場合に使用されます。ポートは、データを送信およびキューから削除する一定のレートを持ちます。バッファは、概念的には、インターフェイスから送信されるまでデータを格納（キューイング）する場所です。

Catalyst 9000シリーズスイッチでは、バッファという単語には2つの用途があります。システムバッファ全体は、ASICの packets buffer complex (PBC) とも呼ばれます。バッファという単語は、PBCの小さな単位を指すこともあります。バッファはキューごとにポートに割り当てられます。つまり、ポートキューには、システムバッファ全体から少量の個別バッファが割り当てられません。

Cisco UADP ASICベースのプラットフォームでは、1つのバッファに最大256バイトのデータが格納され、バッファはリンクされて256バイトを超えるフレームを表します。

スイッチキューバッファ割り当てアルゴリズム

- ポートにはハードコードされたベースバッファ値が割り当てられており、ユーザまたはシステムによってスケールリングできます（ただし、ベース番号は変更できません。これは結果の予測方法を理解するために重要です）
- 特定のポートタイプの基本バッファ値は、ハードウェア（スイッチモデル）、ソフトウェアバージョン、およびポート速度の特定の組み合わせによって異なります。たとえば、Catalyst C9500-25Y4Cのギガビットイーサネットポートは、Catalyst 9200Lのギガビットイーサネットポートとは異なるベースバッファ値で始まります
- この基準値は、設定に基づいて、N個のキューに分割されます。

キューごとの使用可能なバッファの最終的な計算には、次の要因が影響します。

- 設定されたキューの数
- queue-buffers ratio コマンド。このコマンドを使用すると、設定されたポートキュー間でベースバッファがどのように分割されるかを調整できます。
- ユーザ設定の基本バッファ乗数 (qos queue-softmax-multiplier <percent> として設定)
- システム定義の基本バッファ乗数（ユーザーによる構成は不可、4xまたは400%） – デフォルトで適用されますが、この4x値を理解することが割り当てアルゴリズムを予測するために重要であるため、これを削除する状況がいくつかあります
- どちらの乗数も、ソフトバッファと呼ばれるもののみ影響します

柔軟なバッファは、ポート間で共有されるバッファです。これらのバッファは、ポートに対して保証されていないため、ソフトと呼ばれます。

システムは意図的にソフトバッファを割り当てます。これにより、必要に応じて任意の1つのポートで大量のバッファを使用できますが、より多くのポートでバッファが必要になると、Cisco UADP ASIC DTSプロセスの一部としてすべてのポートとキューが動的かつ公平に縮小されます。

要約すると、ソフトバッファ（出力内ではsoftmaxと呼ばれる）は便宜上の最大値です。ポートは、システムバッファ全体からバッファの量を使用できる場合にのみ、完全なsoftmaxを使用します。バッファ要求が他のポートやキューで増加すると、ポートで使用可能な最大バッファは少なくなります。

厳格なバッファはポート用に明示的に予約されたバッファであり、DTSプロセスの影響を受けません。厳格なバッファは保証されたバッファであるため、ポートに割り当てられた厳格なバッファの合計がこれらの厳格なバッファ専用のPBCセグメントを超えることはありません。

ソフトバッファのアクティブなスケールを制御するメカニズムはDTS(Dynamic Threshold and Scale)と呼ばれ、[Catalyst 9000シリーズQoSホワイトペーパー](#)に記載されています。

キューハードバッファとソフトバッファ専用のPBCセグメントのサイズは、システムの設定時に動的に変更され、次の出力ではAQM GlobalSoftLimitおよびGlobalHardLimitとして表示されます。

```
<#root>
```

```
C9500#
```

```
show platform hardware fed active qos queue stats interface tw1/0/1
```

```
-----  
AQM Global counters  
GlobalHardLimit:
```

```
18072
```

```
| GlobalHardBufCount: 0  
GlobalSoftLimit:
```

```
37224
```

```
| GlobalSoftBufCount: 0
```

```
C9500#
```

```
show platform hardware fed active qos queue config interface tw1/0/1
```

Asic:0 Core:1 DATA Port:20 GPN:101 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 160 - 167
DrainFast:Disabled PortSoftStart:2 - 4320 BufferSharing:Disabled
DTS

Hardmax

Softmax

PortSMin	GlbISMin	PortStEnd	QEnable
0	1	2	
480			
3			
1920			
16	960	0	0
4	5760		En

<--- default configuration has a mix of hard buffer and soft buffer in queue 0

1	1	0	
0			
4			
2880			
16	1440	8	720
4	5760		En

<--- default configuration has two queues so some buffers are seen in queue 1

C9500(config)#

policy-map test

C9500(config-pmap)#

```
class class-default
```

```
C9500(config-pmap-c)#
```

```
priority level 1
```

```
<--- Priority level 1 queue configuration on first queue, which is queue 0 in the next output
```

```
C9500(config-pmap-c)#
```

```
exit
```

```
C9500(config-pmap)#
```

```
exit
```

```
C9500(config)#
```

```
int tw1/0/1
```

```
C9500(config-if)#
```

```
service-policy output test
```

```
C9500(config-if)#
```

```
end
```

```
C9500#
```

```
show platform hardware fed active qos queue config interface twe1/0/1
```

```
Asic:0 Core:1 DATA Port:20 GPN:101 LinkSpeed:0x12  
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 160 - 167  
  DrainFast:Disabled PortSoftStart:4 - 1800 BufferSharing:Disabled  
  DTS
```

```
Hardmax
```

Softmax

```
PortSMin GblSMin PortStEnd QEnable
-----
0  1  4
1200
7
1200
0  0  0  0  3  2400  En
```

<--- Hardmax increased to 1200 from 480 in queue 0, softmax reduced to 1200 from 1920

```
1  1  0  0  0  0  0  0  0  3  2400  En
```

<--- queue 1 now no longer has any values, as no second queue is configured

C9500#

show platform hardware fed active qos queue stats interface twe1/0/1

```
-----
AQM Global counters
GlobalHardLimit:
```

18792

| GlobalHardBufCount: 0

<--- GlobalHardLimit increased to 18792 from 18072, or by 720

GlobalSoftLimit:

36504

| GlobalSoftBufCount: 0

<---

GlobalSoftLimit decreased from 37224 to 36504, or by 720



注:GlobalHardLimitが増加し、GlobalSoftLimitが比例的に減少していることに注意してください。

さらに、プライオリティレベル1を設定すると、そのキューのsoftmaxがhardmaxと完全に等しくなるように静的に設定されます。変更できるのは、プライオリティレベル1キューのハードバッファだけです。

GlobalHardLimitとGlobalSoftLimitの変更は720です。これは、設定後のhardmaxの変更と同じです。

このドキュメントのシナリオでは、複数のポリシーマップ設定でsoftmaxとhardmaxの割り当てを計算および予測する方法について説明します。

設定

シナリオ1：特定のポート速度のベースバッファ値の判別

キューの最終バッファ値は、最初にキューに割り当てられる基本値の関数の一部です。この値は、後でソフトバッファの場合に乗算されます。

乗算係数は、他の暗黙的な動作と組み合わせて、特定のキューの最終値を特定の設定で決定する際の課題となります。

結果のキューバッファ割り当てを明確にするための最初の手順は、ベースバッファ値を決定することです。

これを行うには、プライオリティキューを利用します。プライオリティキューは、キュー数または設定されたキューバッファ率に正比例する厳格なバッファを受け取ります。

特定の設定を使用すると、特定のポート速度に割り当てられた基本バッファの量を明示的に取得できます。

すべてのバッファを設定し、単一の非多重化キュー(優先度レベル1キュー)に割り当てる

この例では、他のクラスは設定されていないため、すべてのトラフィックの照合にclass-defaultクラスが使用されています。

```
<#root>
Switch(config)#

policy-map test1

Switch(config-pmap)#

class class-default

Switch(config-pmap-c)#

priority level 1

<--- Assign hard buffer to the port, which is not affected by multipliers

Switch(config-pmap-c)#

queue-buffers ratio 100

<--- Assign all buffers to this queue only
```

前の例の設定では、次のアクションを実行します。

- プライオリティレベル1では、このキュー/クラスに対して厳密なスケジュールを実行し、このキュー/クラスに厳格なバッファを割り当てるようにシステムに指示します。
 - パケットのスケジュール設定については、このドキュメントでは説明しませんが、通常は他のすべてのキューよりも先にストリクトキューが処理されます。
- queue-buffers ratio 100は、100/100、つまり使用可能な基本バッファの100 %をこのキュー/クラスに割り当てます。

複数のクラスを含むpolicy-mapでは、バッファの100 %を単一のクラスに割り当てることはできません。どのクラスにも、最低でも1/100または1 %を割り当てる必要があります。

クラスが1つしかないポリシーでは、クラスは1つしかなく、すべてのバッファをそのクラスに割り当てることができます。

前述したように、プライオリティキューは、設定されたqueue-buffers ratioに従って、ベースバッファの分配と等しい厳格なバッファを取得します。ハードバッファは乗数の影響を受けません。

「Hardmax」という列の出力に厳格なバッファが示されます。

これで、マルチプライヤの影響を受けないバッファを持つ単一のトラフィッククラスが作成されました。これにより、ベースバッファとhardmaxが等しいため、このポート速度に対するベースバッファの割り当てを明示的に取得できます（このプラットフォームでは、このポート速度のみが異なります）。

ベースバッファ=?

キュー比率1 = 100/100 = 1

このキューのHardmax =ベースバッファxキュー比1

$X = Y \times 1$

$X / 1 = Y$

$X = Y$

$X = Y = \text{Hardmax} = \text{基本バッファ} = 1200$ （例2を参照）。

この例では、ポリシーマップtest1が出力サービスポリシーとしてインターフェイスに適用されています

```
<#root>
```

```
9500H(config)#
```

```
int tw1/0/3
```

```
9500H(config-if)#
```

```
service-policy output test1
```

```
<--- service policy that assigns all buffer to the first
```

9500H#

```
show platform hardware fed active qos queue config interface twl/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:4 - 1800 BufferSharing:Disabled
  DTS
```

Hardmax

```
  Softmax  PortSMin  GblSMin  PortStEnd  QEnable
-----
0  1  4
```

1200

```
  7 1200  0  0  0  0  3 2400  En
```

<--- hardmax 1200 - the maximum amount of buffer this port can use without multiplication

```
  1  1  0  0  0  0  0  0  0  0  3 2400  En
  2  1  0  0  0  0  0  0  0  0  3 2400  En
  3  1  0  0  0  0  0  0  0  0  3 2400  En
  4  1  0  0  0  0  0  0  0  0  3 2400  En
  5  1  0  0  0  0  0  0  0  0  3 2400  En
  6  1  0  0  0  0  0  0  0  0  3 2400  En
  7  1  0  0  0  0  0  0  0  0  3 2400  En
```

<snip>

ここに示すように、バッファの100 %が割り当てられたこのプライオリティキューのハードマックスは1200です。

hardmaxは完全に乗算されない/スケールされない値であり、このキューに対してバッファの100 %が設定されているため、スイッチ、ソフトウェアバージョン、および特定のポート速度のこの特定のモデルに対する基本バッファ割り当ては1200です。

この同じスイッチ上の他のポート速度、および同じポート速度の他のモデルのスイッチは、異なるベースバッファ割り当てを受け取ります。この基本割り当てはユーザーが設定することはできず、観察を介して導出する必要があります。

このドキュメントの以降のシナリオでは、すべて同じスイッチ、ソフトウェア、およびポート速度を使用します。したがって、これらはすべて、最終的なバッファ割り当てを決定する計算に1200の基本割り当てを想定しています。



注：前の例のsoftmaxも1200であることに注意してください。

設計上、優先度レベル1のキューのsoftmaxは、そのhardmaxと完全に等しくなっています。これは意図された動作であり、ユーザによる設定はできません。

さらに、このsoftmax割り当ての特定のケースは、後で示すsoftmax乗数の影響を受けません。softmaxのこの動作は、プライオリティレベル1のキューだけが意図されています。

シナリオ2：複数のキューのソフトバッファとハードバッファおよび明示的な比率の判別

このシナリオでは、追加のキューが追加されます。このキューはプライオリティレベル1を使用しないため、softmaxは乗数でスケールします。

一方のマルチプライヤはユーザ設定され、もう一方は非表示または設定不可能なマルチプライヤです。

これらの乗数を、このポート用に導出された基本バッファ（この場合はシナリオ1に従って1200）と組み合わせます。

アルゴリズム：

現在のキュー率=予測されるキュー/クラスのキューバッファ率
非表示の乗数= 400%

ユーザ乗数= qos queue-softmax-multiplier <percent>で設定するパーセント値。デフォルトは100%

$$\text{Softmax} = (\text{Base Buffer} \times (\text{Current Queue Ratio} / 100)) \times \text{Hidden Multiplier} \times (\text{User Multiplier} / 100)$$

<#root>

9500H(config)#

policy-map test2

```
9500H(config-pmap)#
```

```
class class1
```

```
9500H(config-pmap-c)#
```

```
priority level 1
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 50
```

```
<-- class 1 / first queue gets 50% of base buffer
```

```
9500H(config-pmap-c)#
```

```
class class-default
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 100 <-- required configuration due to priority queue, can be ignored for thi
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 50
```

```
<-- class 2 / first queue gets 50% of base buffer
```

値の要約:

- 基本バッファ= 1200 : この番号の検索方法は、シナリオ1に示されています。
- User Multiplier = 100% – デフォルト値。これは変更されていません。
- 非表示の乗数= 4 – デフォルトで、ユーザによる設定はできません。
- Current Queue Ratio(class1) = 50% : ポリシーマップ内で キューバッファ比50
- Current Queue Ratio(class-default) = 50% : ポリシーマップ内でqueue-buffers ratio 50で設定されます。

クラス1バッファ割り当ての決定

class1はプライオリティキューであるため、hardmax (ハードバッファ) を受け取り、乗数の影響を受けないsoftmaxの特殊なケースを受け取ります。

クラス1ハードマックス = (ベースバッファ x 現在のキュー比 (クラス1) / 100)

クラス1 hardmax = 1200 x (50/100) = 600 – プライオリティキューの特殊なケースにより、すべての計算を停止し、結果をhardmaxに割り当てます。Softmaxは、プライオリティレベル1のルールとしてHardmaxと等しくなります。

class-default/バッファ割り当ての決定

class class-default = (ベースバッファ x (現在のキュー比率(class-default / 100)) x 隠し乗数 x (ユーザ乗数 / 100)

class class-default = [

[ベースバッファ] 1200 x [現在のキューの比率] (50/100) = 600

[前の結果] 600 x [隠し乗数] 4 x [ユーザ乗数] (100/100) = 2400

]

<#root>

9500H(config)#

int tw1/0/3

9500H(config-if)#

service-policy output test2

<-- apply the policy

9500H#

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 3600 BufferSharing:Disabled
  DTS Hardmax Softmax PortSMin GlbSMin PortStEnd QEnable
  -----
0 1 5
```

600

9

600

0 0 0 0 1 4800 En

<-- Hardmax is 600 as predicted, Softmax is set equal to Hardmax due to priority level 1

1 1 0 0 10

2400

16 1200 8 600 1 4800 En

<--

Softmax is 2400 as predicted

<snip>

最終結果 : Q0 - ハードマックス : 600 ソフトマックス : 600 Q1 - Softmax:2400

シナリオ2.1 : シナリオ2からユーザ設定の割増率を変更する

このシナリオは、qos queue-softmax-multiplier 1200を設定する点を除き、シナリオ2と同じように開始します。

これにより、現在の設定のsoftmaxバッファが1200 %、つまり係数12倍になります。

値の要約:

- 基本バッファ = 1200 : この番号の検索方法は、シナリオ1に示されています。
- User Multiplier = 1200% : これは、qos queue-softmax-multiplier 1200を介してこのシナリオ

で変更された値です。

- 非表示の乗数= 4 – デフォルトで、ユーザによる設定はできません。
- Current Queue Ratio(class1) = 50% : ポリシーマップ内で キューバッファ比50
- 現在のキュー比率(class-default) = 50% – ポリシーマップ内で キューバッファ比50

クラス1バッファの割り当てを決定します。

class1はプライオリティキューであるため、hardmax (ハードバッファ)を受け取り、乗数の影響を受けないsoftmaxの特殊なケースを受け取ります。

クラス1ハードマックス= (ベースバッファx現在のキュー比 (クラス1) / 100)

クラス1 hardmax = 1200 x (50/100) = 600 – プライオリティキューの特殊なケースにより、すべての計算を停止し、結果をhardmaxに割り当てます。Softmaxは、プライオリティレベル1のルールとしてHardmaxと等しくなります。

class-default/バッファ割り当ての決定:

class class-default = (ベースバッファ x (現在のキュー比率(class-default / 100)) x 隠し乗数 x (ユーザ乗数 / 100)

class class-default =[

[ベースバッファ] 1200 x [現在のキューの比率] (50/100) = 600

[前の結果] 600 x [隠し乗数] 4 x [ユーザ乗数] (1200/100) = 28800

]

qos queue-softmax-multiplier 1200を設定し、softmaxへの変更を確認します (softmaxはそのキューの最大バッファ値で、現在の全体的なバッファ使用状況に基づいて動的に調整されます)。

<#root>

9500H(config)#

qos queue-softmax-multiplier 1200

9500H#

```
show platform hardware fed active qos queue config interface twl/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:3 - 31500 BufferSharing:Disabled
  DTS Hardmax Softmax PortSMin GlbSMin PortStEnd QEnable
  -----
0 1 5
```

600

5

600

```
0 0 0 0 6 42000 En
```

<-- Queue 0 does not change as its configured with priority level 1

```
1 1 0 0 6
```

28800

```
1 900 1 900 6 42000 En
```

<-- Softmax increases by 12x to 28800 from 1200 due to queue-softmax-multiplier 1200

<snip>

最終結果 : Q1 - ハードマックス : 600、ソフトマックス : 600 Q2 - Softmax:28800

シナリオ3 : 複数のキューのソフトおよびハードバッファと暗黙的な比率の判別

このシナリオでは、5つのキューが設定されていますが、queue-buffers ratio が明示的に定義されているのは4つだけです。これらのキューに割り当てられるバッファは、前述の例と同じです。

設定されていないキューは、設定されたすべてのキューバッファの合計と100の差を受け取りません。

明示的に設定された比率の合計 = (Q0バッファ比率) + (Q1バッファ比率) ... (最終バッファ

比率) - Catalyst 9000シリーズスイッチでは最大8つのキューがサポートされているため、最大8つの比率を追加できます

暗黙的な比率Leftover = (100 - 明示的に構成された比率の合計)。

Implicit Ratio Leftoverは、queue-buffers ratioが設定されていないキューに割り当てられる値です。

このシナリオで使用されているポリシーマップ :

```
<#root>
```

```
9500H(config)#
```

```
policy-map test3
```

```
9500H(config-pmap)#
```

```
class class1
```

```
9500H(config-pmap-c)#
```

```
priority level 1
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 20
```

```
9500H(config-pmap-c)#
```

```
class class2
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 10 <-- no queue-buffers ratio statement for this class
```

```
9500H(config-pmap-c)#
```

```
class class3
```

```
9500H(config-pmap-c)#  
  
bandwidth remaining percent 10  
  
9500H(config-pmap-c)#  
  
queue-buffers ratio 10  
  
<-- rest of queues have an explicit queue-buffers ratio  
  
9500H(config-pmap-c)#  
  
class class4  
  
9500H(config-pmap-c)#  
  
bandwidth remaining percent 10  
  
9500H(config-pmap-c)#  
  
queue-buffers ratio 10  
  
9500H(config-pmap-c)#  
  
class class-default  
  
9500H(config-pmap-c)#  
  
bandwidth remaining percent 70  
  
9500H(config-pmap-c)#  
  
queue-buffers ratio 40
```

値の要約:

- 基本バッファ= 1200 : この番号の検索方法は、シナリオ1に示されています。
- User Multiplier = 100% – デフォルト値。これは変更されていません。
- 非表示の乗数= 4 – デフォルトで、ユーザによる設定はできません。

- Current Queue Ratio(class1) = 20% : ポリシーマップ内で キューバッファ率20
- 現在のQueue Ratio(class2) = Unknownです。これを解決する必要があります。queue-buffers ratioはこのクラスに設定されていません。
- 現在のキュー率(class3) = 10% – ポリシーマップ内で queue-buffers ratio 10 (キューバッファ率10)
- Current Queue Ratio(class4) = 10% : ポリシーマップ内でqueue-buffers ratio 10で設定されます。
- 現在のキュー比率(class-default) = 40% – ポリシーマップ内で キューバッファ比40

残りのキューバッファ率を計算:

暗黙的な比率Leftover = (100 – 明示的に構成された比率の合計)。

$$100 - (20) - (10) - (10) - (50) = 20$$

現在のキュー比率(class2) = 20

最終キューバッファ割り当ての計算

Class1 = [Base Buffer] 1200 x [Current Queue Ratio(class1)] (20/100) = 240 – プライオリティキュー、それ以上の計算なし

クラス2 = [

[Base Buffer] 1200 x [Current Queue Ratio(class2)] (10/100) = 240 – このキューの基本バッファ割り当て。ただし、非プライオリティキューのsoftmaxを取得するには乗算する必要があります。
[このキューのベースバッファ割り当て] 120 x [隠し乗数] 4 x [ユーザ乗数] (100/100) = 960

]

残りのキューに対して繰り返します。

クラス3 = [

$$1200 \times (10/100) = 120$$

$$120 \times 4 \times (100/100) = 480$$

]

クラス4 = [

$$1200 \times (10/100) = 120$$

120 x 4 x (100/100) = 480

]

class class-default = [

1200 x (40/100) = 480

600 x 4 x (100/100) = 1920

]

予測と比較して適用されたtest3ポリシーマップの結果:

<#root>

9500H(config)#

int tw1/0/3

9500H(config-if)#

service-policy output test3

9500H#

show platform hardware fed active qos queue config interface tw1/0/3

Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12

AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183

DrainFast:Disabled PortSoftStart:4 - 2880 BufferSharing:Disabled

DTS Hardmax Softmax PortSMin GblSMin PortStEnd QEnable

	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
0	1	5	240	8	240	0	0	0	0	6	3840	En
1	1	0	0	9								

960

16	480	8	240	6	3840	En
----	-----	---	-----	---	------	----

<-- queue without queue buffers ratio configured receives any leftover ratio, as predicted

2	1	0	0	11	480	16	240	8	120	6	3840	En
3	1	0	0	11	480	16	240	8	120	6	3840	En
4	1	0	0	4	1920	16	960	8	480	6	3840	En

<snip>

最終結果 : Q0 - ハードマックス : 240、ソフトマックス : 240 Q1 - Softmax:960、 Q2 - Softmax:480、 Q3 - Softmax:480、 Q4 - Softmax:480

シナリオ3.1 : 複数のキューと複数の暗黙的な比率に対するソフトバッファとハードバッファの決定

このシナリオでは、5つのキューが設定されており、2つのキューにはqueue-buffers ratioが設定されていません。

バッファ割り当てを決定するには、シナリオ2と同じロジックに従いますが、Implicit Ratio Leftoverを、queue-buffers ratioを持たない暗黙的なキューまたはキューの合計数で除算する必要があります

このシナリオで使用されているポリシーマップ :

```
<#root>
```

```
9500H(config)#
```

```
policy-map test4
```

```
9500H(config-pmap)#
```

```
class class1
```

```
9500H(config-pmap-c)#
```

```
priority level 1
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 20
```

```
9500H(config-pmap-c)#
```

```
class class2
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 10 <-- no queue-buffers ratio statement for this class
```

```
9500H(config-pmap-c)#
```

```
class class3
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 10 <-- no queue-buffers ratio statement for this class
```

```
9500H(config-pmap-c)#
```

```
class class4
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 10
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 10
```

```
9500H(config-pmap-c)#
```

```
class class-default
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 70
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 40
```

値の要約:

- 基本バッファ= 1200 : この番号の検索方法は、シナリオ1に示されています。
- ユーザ乗数= 100 % – デフォルト値
- 非表示の乗数= 4 – デフォルトで、ユーザによる設定はできません。
- Current Queue Ratio(class1) = 20% : ポリシーマップ内で キューバッファ率20
- 現在のQueue Ratio(class2) = Unknownです。これを解決する必要があります。queue-buffers ratioはこのクラスに設定されていません。
- 現在のQueue Ratio(class3) = Unknownです。これを解決する必要があります。queue-

buffers ratioはこのクラスに設定されていません。

- Current Queue Ratio(class4) = 10% : ポリシーマップ内でqueue-buffers ratio 10で設定されます。
- Current Queue Ratio(class-default) = 40% : ポリシーマップ内でqueue-buffers ratio 40で設定されます。

残りのキューバッファ率を計算:

暗黙的な比率Leftover = (100 - 明示的に構成された比率の合計) 。

暗黙的キューの数= 2(class2およびclass3にqueue-buffers ratio が定義されていない)

構成された比率の合計= 20+40+10 = 70

暗黙の比残存率= 100 - 70 = 30

暗黙のキュー比割当て= [暗黙の比率の残り時間] 30 / [暗黙のキューの数] 2 = 15

最終的なキューバッファ割り当ての計算:

クラス1 =

[基本バッファ] 1200 x [現在のキュー比率 (クラス1)] (20/100) = 240 – 優先キュー、それ以上の計算なし

クラス2 =

[Base Buffer] 1200 x [Implicit queue ratio allocation] (15/100) = 180 – クラス2にはqueue-buffers ratioが定義されていないため、明示的なキューの残りのqueue-buffers ratioは暗黙的なキュー間で共有されます。

[このキューのベースバッファ割り当て] 180 x [隠し乗数] 4 x [ユーザ乗数] (100/100) = 720

残りのキューに対して繰り返します。

クラス3 = [

```
1200 x (15/100) = 180
120 x 4 x (100/100) = 720
]
```

クラス4 = [

```
1200 x (10/100) = 120
120 x 4 x (100/100) = 480
]
```

class class-default = [

```
1200 x (40/100) = 480
600 x 4 x (100/100) = 1920
]
```

予測と比較して適用されたtest4ポリシーマップの結果:

<#root>

9500H(config)#

interface tw1/0/3

9500H(config-if)#

service-policy output test4

9500H#

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:4 - 2880 BufferSharing:Disabled
  DTS   Hardmax   Softmax   PortSMin   G1b1SMin   PortStEnd   QEnable
  -----
0  1  5  240  8  240  0  0  0  0  6  3840  En
1  1  0   0  9
```

720

```
16  360  8  180  6  3840  En
```

```

<-- queue 1 and 2

were not configured with queue-buffers ratio

2 1 0 0 9

720

16 360 8 180 6 3840 En

<-- queue 1 and 2 get an equal share of leftover buffer ratio

3 1 0 0 11 480 16 240 8 120 6 3840 En
4 1 0 0 4 1920 16 960 8 480 6 3840 En
<snip>

```



注：暗黙的なキュー比割り当ての結果が整数でない場合、等しいシェアは不可能です。この結果は、ポリシーマップ内の前のキューに対しては切り上げられ、後のキューに対しては切り下げられます。割り当てられるキューバッファの比率の最終的な合計は100のままですが、暗黙のキューは、先ほど説明した整数の結果要件のために、常に同等の割り当てを得るわけではありません。

シナリオ4：キュー間のキューバッファ比率の合計が100未満

このシナリオでは、5つのキューが設定され、すべてqueue-buffers ratioで設定されます。クラス間のキューバッファ比率の合計は100未満です。

この場合、未割り当てのバッファ率はクラス間で均等に分散されます。

前のシナリオと同様に、残りのキューバッファ率の分割結果が整数でない場合、各キューへの最終的な割り当ては切り上げられるか切り捨てられ、設定したキューバッファ率に追加されます。

このシナリオで使用されているポリシーマップ：

```

<#root>

9500H(config)#

policy-map test5

```

9500H(config-pmap)#

class class1

9500H(config-pmap-c)#

priority level 1

9500H(config-pmap-c)#

queue-buffers ratio 10

9500H(config-pmap-c)#

class class2

9500H(config-pmap-c)#

bandwidth remaining percent 10

9500H(config-pmap-c)#

queue-buffers ratio 10

9500H(config-pmap-c)#

class class3

9500H(config-pmap-c)#

bandwidth remaining percent 10

9500H(config-pmap-c)#

queue-buffers ratio 10

9500H(config-pmap-c)#

class class4

9500H(config-pmap-c)#

```
bandwidth remaining percent 10
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 10
```

```
9500H(config-pmap-c)#
```

```
class class-default
```

```
9500H(config-pmap-c)#
```

```
bandwidth remaining percent 70
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 12
```

値の要約:

- 基本バッファ = 1200 : この番号の検索方法は、シナリオ1に示されています。
- ユーザ乗数 = 100 % - デフォルト値
- 非表示の乗数 = 4 - デフォルトで、ユーザによる設定はできません。
- Current Queue Ratio(class1) = 10% : ポリシーマップ内で queue-buffers ratio 10 (キューバッファ率10)
- Current Queue Ratio(class2) = 10% : ポリシーマップ内でqueue-buffers ratio 10で設定されます。
- Current Queue Ratio(class3) = 10% : ポリシーマップ内でqueue-buffers ratio 10で設定されます。
- Current Queue Ratio(class4) = 10% : ポリシーマップ内でqueue-buffers ratio 10で設定されます。
- Current Queue Ratio(class-default) = 12 % : ポリシーマップ内で キューバッファ比12
- キューの総数 = 5

構成比率の合計 = $10 + 10 + 10 + 10 + 12 = 52$

バッファ比残存率 = $100\% - 52\% = 48\%$

[Buffer Ratio Leftover] 48% / [Total Number of Queues] 5 = 9.6% added per queue – これは整数ではないため、キューへの最終的な適用はキューごとに切り上げまたは切り下げる必要があります

システムが使用する最終的なキューバッファの比率を取得するには、設定済みのキューバッファの比率に9または10を追加する必要があります。

ポリシーマップで上位のクラスは、切り上げられた値10を受け取ります。ポリシーマップ内で下位のクラスは、丸められた値「9」を受け取ります。

最終キューバッファ割り当ての計算

バッファ率の残り = 48

Class1 = [基本バッファ] x ([現在のキュー比率(class1) + 共有バッファ比率の切り上げ値の残り値])

Class1 = 1200 x ((10% + 10%)/100) = 240 - プライオリティキュー、それ以上の計算なし

バッファ率の残り値 = (48 - 10) = 38

Class2 = [基本バッファ] x ([現在のキュー比率(class2) + 共有バッファ比率の切り上げ値の残り値])

クラス2 = 1200 x ((10% + 10%)/100) = 240 - プライオリティキューではないため、ユーザとシステムの乗数で乗算し続けます。

Class2 = [このキューのベースバッファ割り当て] 240 x [隠し乗数] 4 x [ユーザ乗数] (100/100) = 960 - このキューのsoftmax結果

バッファ率の残り値 = (38 - 10) = 28

残りのキューに対して繰り返します。

クラス3 = [

1200 x ((10+10)/100) = 240

120 x 4 x (100/100) = 960

]

バッファ率の残り値 = (28 - 10) = 18

クラス4 = [

$1200 \times ((10+9)/100) = 240$

$120 \times 4 \times (100/100) = 912$

]

バッファ率の残り=9

class class-default= [

$1200 \times ((12+9)/100) = 252$

$120 \times 4 \times (100/100) = 1008$

]

バッファ率の残り=0

予測と比較して適用されたtest5ポリシーマップの結果:

<#root>

9500H#

show platform hardware fed active qos queue config interface tw1/0/3

Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
DrainFast:Disabled PortSoftStart:5 - 1512 BufferSharing:Disabled
DTS Hardmax Softmax PortSMin GblSMin PortStEnd QEnable

0 1 5 240 9

240

0	0	0	0	6	2016	En
1	1	0	0	10		

960

16	480	8	240	6	2016	En
2	1	0	0	10		

960

16	480	8	240	6	2016	En
3	1	0	0	11		

912

16	456	8	228	6	2016	En
----	-----	---	-----	---	------	----

```
4 1 0 0 12
```

```
1008
```

```
16 504 8 252 6 2016 En  
<snip>
```

シナリオ5：プライオリティレベル2クラスのソフトバッファとハードバッファの判別

このシナリオでは、ポリシーマップのクラスはプライオリティレベル2で設定されます。

プライオリティレベル1では乗数の影響を受けず、hardmaxと同じ値に設定されていますが、プライオリティレベル2では、softmaxに乘算すると同時に厳格なバッファ(hardmax)が割り当てられます。

このシナリオで使用されているポリシーマップ：

```
<#root>
```

```
9500H(config)#
```

```
policy-map test6
```

```
9500H(config-pmap)#
```

```
class class1
```

```
9500H(config-pmap-c)#
```

```
priority level 1
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 50 <-- 50 / 50 split between both queues
```

```
9500H(config-pmap-c)#
```

```
class class-default
```

```
9500H(config-pmap-c)#
```

```
priority level 2
```

```
<-- Priority level 2 in use now
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 50 <-- 50 / 50 split between both queues
```

適用されたtest6 policy-mapの結果:

```
<#root>
```

```
9500H#
```

```
show platform hardware fed active qos queue config interface tw1/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 3600 BufferSharing:Disabled
  DTS Hardmax Softmax PortSMin Glb1SMin PortStEnd QEnable
  ---- -
0 1 5 600 9
```

```
600
```

```
0 0 0 0 1 4800 En
```

```
<-- Softmax is equal to hardmax
```

```
1 1 5 600 10
```

```
2400
```

```
16 1200 0 0 1 4800 En
```

```
<-- Softmax is multiplied by Hidden Multiplier (400%) and User Multiplier (100% default)r
```

```
<snip>
```

上記の出力では、2番目のキューsoftmaxには最初のキューの4倍のsoftmaxがあります。これは、priority-level 1 softmaxは特にsystem softmax乗数の影響を受けませんが、priority-level 2は影響を受けるためです。

ユーザsoftmax乗数を設定する場合、priority-level 2キューだけが影響を受けます。

```
<#root>
```

```
9500H(config)#
```

```
qos queue-softmax-multiplier 200
```

```
<#root>
```

```
9500H#
```

```
show platform hardware fed active qos queue config interface tw1/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 7200 BufferSharing:Disabled
    DTS Hardmax Softmax PortSMin GlbSMin PortStEnd QEnable
    -----
    0 1 5 600 9 600 0 0 0 0 5 9600 En
```

```
<--- priority-level 1 queue unaffected by softmax multiplier
```

```
1 1 5 600 10
```

```
4800
```

```
8 1200 0 0 5 9600 En
```

```
<--- User multiplier increased to 200%, softmax for this queue doubles
```

```
<snip>
```

キュー制限

キュー制限の設定は、最終的なキューバッファの割り当てに影響します

キューバッファの割り当てに影響を与える主なメカニズムは、MQCポリシーマップでキューごとに追加されるqueue buffers-ratio設定です。

ただし、キューバッファの割り当ては他の設定の影響を受けます。

Queue-limitでは、(Weighted Tail Drop (WTD ; 重み付けテールドロップ) を介して)特定のトラフィッククラスをドロップするしきい値が定義されますが、このドキュメントでは説明しません。

特定の状況では、queue-limitはソフトバッファキューのシステムHidden Multiplier (隠し乗数) を変更します。これは、queue-limitが適用されるそのキューの全体的なソフトバッファの割り当てに影響します。

まず、キュー制限はクラスごとに最大3回設定できることを理解します。これは、DSCPまたはCoSごとにWTDに対して最大3つのしきい値を定義します。

次の出力では、2つのしきい値だけが定義されています。

ポリシーマップに2つのキュー制限を適用します。

```
<#root>
```

```
9500H(config)#
```

```
policy-map test7
```

```
9500H(config-pmap)#
```

```
class class1
```

```
9500H(config-pmap-c)#
```

```
priority level 1
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 50
```

```
9500H(config-pmap-c)#
```

```
class class-default
```

```
9500H(config-pmap-c)#
```

```
priority level 2
```

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 50
```

```
9500H(config-pmap-c)#
```

```
queue-limit dscp af11 percent 10 <-- Tells system to drop af11 traffic at 10% queue utilization
```

```
9500H(config-pmap-c)#
```

```
queue-limit dscp af12 percent 50 <-- Tells system to drop af12 traffic at 50% queue utilization
```

バッファ割り当ての結果を確認します。

```
<#root>
```

```
9500H(config-pmap-c)#
```

```
interface tw1/0/3
```

```
9500H(config-if)#
```

```
service-policy output test7
```

```
9500H#
```

```
show platform hardware fed active qos queue config interface tw1/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 7200 BufferSharing:Disabled
  DTS  Hardmax  Softmax  PortSMin  Glb1SMin  PortStEnd  QEnable
  -----
0  1  5  600  9  600  0  0  0  0  5  9600  En
1  1  5  600  10
```

```
4800
```

```
8 1200 0 0 5 9600 En
```

```
<--- final result for queue that contains 2 queue-limit statements is 4800
```

```
<snip>
```

次の例では、3番目のqueue-limit設定をclass class-defaultに追加します。

バッファ割り当ての結果を確認します。

```
<#root>
```

```
9500H(config)#
```

```
policy-map test7
```

```
9500H(config-pmap)#
```

```
class class-default
```

```
9500H(config-pmap-c)#
```

```
queue-limit dscp af13 percent 100
```

```
9500H#
```

```
show platform hardware fed active qos queue config interface tw1/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 1800 BufferSharing:Disabled
  DTS  Hardmax  Softmax  PortSMin  GlbISMin  PortStEnd  QEnable
  ----  -
0   1  5   600   9   600   0   0   0   0   5  2400   En
1   1  5   600  10
```

```
1200
```

```
32 1200 0 0 5 2400 En
```

```
<-- Softmax reduces by 400% from previous example
```

```
<snip>
```

3つ目のqueue-limit設定がキューに追加されると、そのキューに対してシステムの隠しソフトバッファ乗数である400 %が無効になります。ただし、そのキューは、ユーザ設定のqos queue-softmax-multiplier <percent>を引き続き尊重します。

関連情報

- [Cisco Catalyst 9000スイッチングプラットフォーム：QoSとキューイングに関するホワイトペーパー](#)
- [シスコのテクニカルサポートとダウンロード](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。