

# Catalyst 9000 스위치의 대기열 버퍼 할당 이해

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[관련 제품](#)

[배경 정보](#)

[용어](#)

[버퍼](#)

[스위치 대기열 버퍼 할당 알고리즘](#)

[구성](#)

[시나리오 1: 지정된 포트 속도의 기본 버퍼 값 결정](#)

[시나리오 2: 다중 대기열 및 명시적 비율에 대한 소프트 및 하드 버퍼 결정](#)

[시나리오 2.1: 시나리오 2에서 사용자 구성 승수 수정](#)

[시나리오 3: 여러 대기열에 대한 소프트 및 하드 버퍼 및 암시적 비율 결정](#)

[시나리오 3.1: 다중 대기열 및 다중 암시적 비율에 대한 소프트 및 하드 버퍼 결정](#)

[시나리오 4: 대기열 간 대기열 버퍼 비율 한계가 100보다 작음](#)

[시나리오 5: 우선순위 레벨 2 클래스의 소프트 및 하드 버퍼 결정](#)

[큐 제한](#)

[관련 정보](#)

---

## 소개

이 문서에서는 Catalyst 9000 Series 스위치의 트래픽 대기열에 대한 대기열 버퍼 할당을 예측하는 방법에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- QoS(Quality of Service) 개념 및 패킷 표시, 대기 및 예약 방식 프로세스
- Cisco MQC QoS 컨피그레이션

## 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 관련 제품

이 문서는 다음 하드웨어 및 소프트웨어 버전에서도 사용할 수 있습니다.

- Catalyst 9200 - 9600 Series 스위치
- Catalyst 9300X & 9400X
- 모든 Cisco IOS XE 소프트웨어 버전



참고: 9500X 또는 9600X는 서로 다른 ASIC 및 QoS 아키텍처를 사용하므로 이 문서는 9500X에 적용되지 않습니다.

## 배경 정보

Catalyst 9000 Series 스위치의 QoS에 대한 기술 개요는 다음을 참조하십시오. [Catalyst 9000 QoS 및 대기열 처리 백서](#)

종종 버퍼 할당은 특정 트래픽 클래스에 대한 원하지 않는 출력 삭제에 대한 응답으로 조정되어야 합니다. Catalyst 9000 Series 스위치의 출력 삭제를 진단하고 문제를 해결하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오. [Catalyst 9000 스위치의 출력 삭제 문제 해결](#)

## 용어

QoS	QoS(Quality of Service)	네트워크 디바이스 안팎의 트래픽을 분류, 표시, 대기열 지정 및 스케줄링 하는 것과 관련된 개념/관련 기능 그룹
DSCP	Differentiated Services Code Point	패킷의 IP 헤더에 포함된 트래픽 분류 메커니즘
CoS	서비스 등급	패킷의 이더넷 프레임 헤더에 포함된 트래픽 분류 메커니즘
에이스	액세스 제어 항목	ACL(Access Control List) 내의 단일 규칙 또는 라인
ACL	액세스 제어 목록	트래픽을 일치시키고 작업을 수행하기 위해 다양한 기능에서 사용하는 ACE(Access Control Entry) 그룹
ASIC	주문형 집적 회로	특정 작업 또는 작업의 집합을 고효율로 수행하도록 설계된 컴퓨터 칩입니다.
UADP	Unified Access 데이터 플레인	Cisco ASIC는 Catalyst 9000 Series 스위치에서 많은 네트워크 패킷 처리 작업을 수행하는 데 사용됩니다.
PBC	패킷 버퍼 컴플렉스	패킷을 처리, 대기열 처리 및 스케줄링하는 중앙 패킷 버퍼 역할을 하는 Cisco UADP ASIC 하위 시스템.
AQM	활성 대기열 관리	네트워크 포트에 대한 트래픽 대기열 및 예약 작업을 관리하는 Cisco UADP ASIC 하위 시스템입니다.
DTS	동적 임계값 및 확장	포트 간에 버퍼를 동적으로 조정하고 확장하여 하드웨어 활용을 최적화하는 Cisco UADP ASIC 기술

## 버퍼

개념상, 버퍼는 포트로 스위칭되거나 라우팅되는 데이터가 유선에 데이터를 배치하는 포트 성능을 초과하는 경우, 데이터의 일시적인 버스트를 흡수하는 데 사용되는 메모리입니다. 포트는 데이터를 전송하고 대기열에서 제거하는 고정된 속도를 가집니다. 버퍼는 개념상 인터페이스 외부로 전송될 때까지 데이터를 저장하거나 대기시키는 장소입니다.

Catalyst 9000 Series 스위치에서 버퍼라는 단어는 두 가지 용도로 사용됩니다. 시스템 버퍼 전체를 ASIC의 PBC(Packet Buffer Complex)라고 합니다. 워드 버퍼들은 또한 PBC의 작은 유닛을 참조할 수 있다. 버퍼는 대기열별로 포트에 할당됩니다. 즉, 포트 대기열은 전체 시스템 버퍼에서 소량의 개별 버퍼가 할당됩니다.

Cisco UADP ASIC 기반 플랫폼에서 하나의 버퍼에는 최대 256바이트의 데이터가 포함되어 있으며 버퍼는 서로 연결되어 256바이트보다 큰 프레임을 나타냅니다.

## 스위치 대기열 버퍼 할당 알고리즘

- 포트에는 하드 코딩된 기본 버퍼 값이 할당되어 있으며 사용자 또는 시스템에서 크기를 조정할 수 있습니다. 그러나 기본 번호는 변경할 수 없으며, 이는 결과를 예측하는 방법을 이해하는 데 중요합니다
- 지정된 포트 유형에 대한 기본 버퍼 값은 하드웨어(스위치 모델), 소프트웨어 버전 및 포트 속도의 특정 조합에 따라 달라집니다. 예를 들어, Catalyst C9500-25Y4C의 기가비트 이더넷 포트는 Catalyst 9200L의 기가비트 이더넷 포트와 다른 기본 버퍼 값으로 시작합니다
- 이 기본 값은 컨피그레이션을 기준으로 N개의 대기열로 분할됩니다.

대기열당 사용 가능한 버퍼의 최종 계산은 다음 요인의 영향을 받습니다.

- 구성된 대기열 수
- 대기열-버퍼 비율. 구성된 포트 대기열 간에 기본 버퍼가 분할되는 방식을 조정할 수 있는 명령입니다.
- 사용자가 구성한 기본 버퍼 멀티플라이어(qos queue-softmax-multiplier <percent>로 구성됨)
- 시스템 정의 기본 버퍼 승수(사용자가 구성할 수 없음, 4x 또는 400%) - 이 값이 기본적으로 적용되지만, 할당 알고리즘을 예측하기 위해 이 4x 값을 이해하는 것이 중요하기 때문에 이 값이 제거되는 경우가 있습니다
- 두 승수는 모두 소프트 버퍼라고 하는 것에만 영향을 미칩니다

소프트 버퍼는 포트 간에 공유되는 버퍼입니다. 이러한 버퍼는 포트에 보장되지 않으므로 소프트 버퍼라고 합니다.

시스템이 소프트 버퍼를 의도적으로 할당합니다. 이렇게 하면 필요한 경우 어느 한 포트에서 많은 수의 버퍼를 사용할 수 있지만, 더 많은 포트에서 버퍼가 필요하기 때문에 Cisco UADP ASIC DTS 프로세스의 일부로 모든 포트 및 큐가 동적이고 균등하게 축소됩니다.

요약하면, 소프트 버퍼는(출력에서 softmax라고 함) 기회주의적인 최대값입니다. 포트는 전체 시스템 버퍼에서 사용할 수 있는 버퍼 양이 있는 경우에만 전체 softmax를 사용합니다. 다른 포트 및 대기열에서 버퍼 수요가 증가하면 해당 포트에서 사용할 수 있는 최대 버퍼가 줄어듭니다.

하드 버퍼는 포트에 명시적으로 예약된 버퍼이며 DTS 프로세스의 영향을 받지 않습니다. 하드 버퍼는 보장된 버퍼이므로 포트에 할당된 총 하드 버퍼의 합계가 이러한 하드 버퍼 전용 PBC 세그먼트를 초과하지 않습니다.

소프트 버퍼의 액티브 스케일을 제어하는 메커니즘은 DTS(Dynamic Threshold and Scale)라고 하며, 이는 [Catalyst 9000 Series QoS 백서](#)에 설명되어 있습니다.

대기열 하드 및 소프트 버퍼 전용 PBC 세그먼트의 크기는 시스템을 구성할 때 동적으로 변경되며 이 출력에서는 AQM GlobalSoftLimit 및 GlobalHardLimit로 볼 수 있습니다.

<#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 GblSMin 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 twl/0/1

C9500(config-if)#

service-policy output test

C9500(config-if)#

end

C9500#

show platform hardware fed active qos queue config interface twel/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	GlbISMin	PortStEnd	QEnable
0	1	4	
1200			
7			
1200			
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
```



참고: GlobalHardLimit의 증가와 GlobalSoftLimit의 비례 감소를 확인합니다.

---

또한 우선순위 레벨 1을 구성할 때 해당 대기열의 softmax는 hardmax와 정확히 동일하게 정적으로 설정됩니다. 우선순위 레벨 1 대기열에 대한 하드 버퍼만 수정할 수 있습니다.

GlobalHardLimit 및 GlobalSoftLimit의 변경 사항은 720과 같습니다. 또한 구성 후 hardmax의 변경 사항과 같습니다.

이 문서의 시나리오는 여러 정책 맵 컨피그레이션에서 softmax 및 hardmax 할당을 계산하고 예측하는 방법을 설명합니다.

# 구성

## 시나리오 1: 지정된 포트 속도의 기본 버퍼 값 결정

대기열 최종 버퍼 값은 대기열 간에 먼저 할당되는 기본 값의 함수이기도 합니다. 그런 다음 소프트 버퍼의 경우 이 값이 나중에 곱해집니다.

곱하기 인자는 다른 암시적 동작과 결합하여 주어진 큐의 최종 값을 결정하고, 주어진 컨피그레이션과 함께 챌린지를 결정합니다.

결과 대기열 버퍼 할당을 명확히 하는 첫 번째 단계는 기본 버퍼 값을 결정하는 것입니다.

이렇게 하려면 우선 순위 큐를 활용하십시오. 우선 순위 큐는 큐 수 또는 구성된 큐 버퍼 비율에 직접 비례하는 하드 버퍼를 수신합니다.

특정 컨피그레이션을 사용하면 지정된 포트 속도에 할당된 기본 버퍼의 양을 명시적으로 파생할 수 있습니다.

모든 버퍼를 구성하여 곱하지 않은 단일 대기열(우선순위 레벨 1 대기열)에 할당

이 예에서는 다른 클래스가 구성되어 있지 않으므로 class-default 클래스를 사용하여 모든 트래픽을 확인합니다.

```
<#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
```

이전 예의 컨피그레이션에서는 다음 작업을 수행합니다.

- 우선순위 레벨 1은 이 대기열/클래스에 대해 엄격한 스케줄을 수행하고 이 대기열/클래스에 하드 버퍼를 할당하도록 시스템에 지시합니다.
  - 패킷 예약 방식은 이 문서의 범위를 벗어나지만 일반적으로 엄격한 대기열이 다른 대기열보다 먼저 서비스됩니다.
- 대기열-버퍼 비율 100은 이 대기열/클래스에 사용 가능한 기본 버퍼의 100/100 또는 100%를 할당합니다.

둘 이상의 클래스가 있는 정책 맵에서는 버퍼의 100%를 단일 클래스에 할당할 수 없습니다. 모든 클래스에 최소 1/100 또는 1%를 할당해야 합니다.

클래스가 하나만 있는 정책에서는 클래스가 하나만 있으며 모든 버퍼를 할당할 수 있습니다.

앞에서 설명한 것처럼, 우선 순위 큐는 구성된 큐 버퍼 비율에 따라 기본 버퍼의 분포와 동일한 하드 버퍼를 가져옵니다. 하드 버퍼에는 승수가 적용되지 않습니다.

하드 버퍼는 Hardmax라는 열 아래의 출력에서 관찰됩니다.

이제 멀티플라이어가 적용되지 않는 버퍼가 있는 단일 트래픽 클래스가 제공됩니다. 이 기능을 사용하면 기본 버퍼와 hardmax가 같으므로 이 포트 속도에 대한 기본 버퍼 할당을 명시적으로 파생할 수 있습니다(그리고 이 플랫폼에서는 이 포트 속도만 다르지만).

기본 버퍼 = ?

대기열 비율 1 =  $100/100 = 1$

이 대기열에 대한 Hardmax = 기본 버퍼 x 대기열 비율 1

$$X = Y \times 1$$

$$X / 1 = Y$$

$$X = Y$$

X = Y = Hardmax = 기본 버퍼 = 1200(예 2 참조).

이 예에서 policy-map test1은 인터페이스에 출력 service-policy로 적용됩니다

```
<#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 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 - 1800 BufferSharing:Disabled
  DTS
```

Hardmax

Softmax	PortSMin	GlbSMin	PortStEnd	QEnable
0	1	4		
7	1200	0	0	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  0  3  2400  En
4  1  0  0  0  0  0  0  0  0  0  3  2400  En
5  1  0  0  0  0  0  0  0  0  0  3  2400  En
6  1  0  0  0  0  0  0  0  0  0  3  2400  En
7  1  0  0  0  0  0  0  0  0  0  3  2400  En
<snip>

```

표시된 대로, 100%의 버퍼가 할당된 이 우선순위 대기열의 하드맥스는 1200입니다.

hardmax는 곱하지 않거나 스케일링되지 않은 값이며, 버퍼의 100%가 이 대기열에 구성되어 있으므로, 이 특정 스위치 모델, 소프트웨어 버전 및 특정 포트 속도에 대한 기본 버퍼 할당은 1200입니다.

동일한 스위치의 다른 포트 속도와 동일한 포트 속도의 다른 스위치 모델은 서로 다른 기본 버퍼 할당을 받습니다. 이 기본 할당은 사용자가 구성할 수 없으며 관찰을 통해 파생되어야 합니다.

이 문서의 추가 시나리오에서는 모두 동일한 스위치, 소프트웨어 및 포트 속도를 사용합니다. 따라서 이들은 모두 최종 버퍼 할당을 결정하기 위해 계산에 1200의 기본 할당을 가정합니다.



참고: 이전 예에서 softmax도 1200입니다.

우선 순위 수준 1 큐의 softmax는 hardmax와 정확히 같습니다. 이는 사용자가 구성할 수 있는 것이 아닙니다.

또한, 이러한 특정 softmax 할당 사례는 이후 보여진 softmax 승수의 영향을 받지 않는다. 우선순위 레벨 1 대기열에 대해서만 이 동작이 소프트맥스에 대해 수행되며, 이는 의도한 것입니다.

## 시나리오 2: 다중 대기열 및 명시적 비율에 대한 소프트 및 하드 버퍼 결정

이 시나리오에서는 추가 대기열이 추가됩니다. 이 대기열은 우선 순위 수준 1을 사용하지 않으므로 softmax는 승수로 확장됩니다.

한 승수는 사용자가 구성하고 다른 승수는 숨겨진/구성 불가능한 승수입니다.

이 승수를 이 포트에 대해 파생된 기본 버퍼와 결합합니다(이 경우 시나리오 1에 따라 1200).

알고리즘적:

현재 대기열 비율 = 예측할 대기열/클래스에 대한 대기열-버퍼 비율  
숨겨진 승수 = 400%

User Multiplier = qos 큐에서 구성한 퍼센트 값 softmax-multiplier <percent>입니다. 기본값은 100%입니다

Softmax = ( 기본 버퍼 x (현재 대기열 비율 / 100)) x 숨겨진 승수 x (사용자 승수 / 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 this
```

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

```
queue-buffers ratio 50
```

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

값의 요약:

- 기본 버퍼 = 1200 - 이 숫자를 찾는 방법은 시나리오 1에 나와 있습니다.
- 사용자 승수 = 100% - 기본값, 아직 수정되지 않음
- 숨겨진 승수 = 4 - 기본값 및 사용자 구성 불가
- Current Queue Ratio(class1) = 50% - 정책 맵에서 대기열-버퍼 비율 50
- 현재 대기열 비율(class-default) = 50% - 대기열 버퍼 비율 50으로 정책 맵에 구성됨

Class1 버퍼 할당을 결정합니다.

class1은 우선 순위 큐이므로 hardmax(하드 버퍼)와 multiplier의 영향을 받지 않는 softmax의 특수한 경우를 받습니다.

Class1 hardmax = (기본 버퍼 x 현재 대기열 비율(class1) / 100)

Class1 hardmax = 1200 x (50/100) = 600 - 우선순위 대기열의 특수한 경우 모든 계산을 중지하고 결과를 hardmax에 할당합니다. Softmax는 우선순위 레벨 1에 대한 규칙으로 Hardmax와 같습니다.

클래스 기본 버퍼 할당을 결정합니다.

클래스 클래스 클래스 기본값 = (기본 버퍼 x (현재 대기열 비율(클래스 기본값 / 100)) x 숨겨진 승수 x (사용자 승수 / 100)

클래스 class-default = [

[기본 버퍼] 1200 x [현재 대기열 비율] (50/100) = 600

[이전 결과] 600 x [숨겨진 배수] 4 x [사용자 배수] (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 Glb1SMin 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>
```

최종 결과: Q0 - 하드웨어: 600 소프트맥스: 600. Q1 - 소프트맥스: 2400

## 시나리오 2.1: 시나리오 2에서 사용자 구성 승수 수정

이 시나리오는 시나리오 2와 동일하게 시작합니다. 단, 이제 qos queue-softmax-multiplier 1200을 구성합니다.

이렇게 하면 현재 컨피그레이션의 softmax 버퍼와 1200% 또는 12배수가 곱해집니다.

값의 요약:

- 기본 버퍼 = 1200 - 이 숫자를 찾는 방법은 시나리오 1에 나와 있습니다.
- 사용자 멀티플라이어 = 1200% - 이 시나리오에서 qos queue-softmax-multiplier 1200을 통해 변경된 값입니다.
- 숨겨진 승수 = 4 - 기본값 및 사용자 구성 불가
- Current Queue Ratio(class1) = 50% - 정책 맵에서 대기열-버퍼 비율 50
- Current Queue Ratio(class-default) = 50% - 정책 맵에서 대기열-버퍼 비율 50

Class1 버퍼 할당을 결정합니다.

class1은 우선 순위 큐이므로 hardmax(하드 버퍼)와 multiplier의 영향을 받지 않는 softmax의 특수한 경우를 받습니다.

Class1 hardmax = (기본 버퍼 x 현재 대기열 비율(class1) / 100)

Class1 hardmax = 1200 x (50/100) = 600 - 우선순위 대기열의 특수한 경우 모든 계산을 중지하고 결과를 hardmax에 할당합니다. Softmax는 우선순위 레벨 1에 대한 규칙으로 Hardmax와 같습니다.

클래스 기본 버퍼 할당을 결정합니다.

클래스 클래스 클래스 기본값 = (기본 버퍼 x (현재 대기열 비율(클래스 기본값 / 100))) x 숨겨진 승수 x (사용자 승수 / 100)

클래스 class-default =[

[기본 버퍼] 1200 x [현재 대기열 비율] (50/100) = 600

[이전 결과] 600 x [숨겨진 배수] 4 x [사용자 배수] (1200/100) = 28800

]

qos queue-softmax-multiplier 1200을 구성하고 softmax에 대한 변경 사항을 관찰합니다(softmax는 해당 대기열에 대한 최대 버퍼 값이며 현재 전체 버퍼 사용량을 기반으로 동적으로 확장됨).

<#root>

9500H(config)#

```
qos queue-softmax-multiplier 1200
```

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:3 - 31500 BufferSharing:Disabled
  DTS   Hardmax   Softmax   PortSMin   G1b1SMin   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>

최종 결과: Q1 - 하드웨어 600, 소프트맥스: 600. Q2 - 소프트맥스: 28800

### 시나리오 3: 여러 대기열에 대한 소프트 및 하드 버퍼 및 암시적 비율 결정

이 시나리오에서는 5개의 대기열이 구성되지만 대기열 버퍼 비율이 명시적으로 정의된 대기열은 4개뿐입니다. 이러한 대기열에 할당된 버퍼는 이전 예와 동일합니다.

구성되지 않은 대기열은 구성된 모든 대기열 버퍼의 합계와 100 사이의 차이를 수신합니다.

명시적으로 구성된 비율의 합계 = (Q0 버퍼 비율) + (Q1 버퍼 비율) ... (최종 버퍼 비율) - 최대 8개의 대기열이 Catalyst 9000 Series 스위치에서 지원되므로 최대 8개의 비율을 추가할 수 있습니다.

암시적 비율 왼쪽 = (100 - 명시적으로 구성된 비율의 합계).

Implicit Ratio Left Tover는 큐 버퍼 비율이 구성되지 않은 큐에 할당된 값입니다.

이 시나리오에 사용 중인 정책 맵:

```
<#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
```

값의 요약:

- 기본 버퍼 = 1200 - 이 숫자를 찾는 방법은 시나리오 1에 나와 있습니다.
- 사용자 승수 = 100% - 기본값, 아직 수정되지 않음
- 숨겨진 승수 = 4 - 기본값 및 사용자 구성 불가
- Current Queue Ratio(class1) = 20% - 정책 맵에서 대기열-버퍼 비율 20
- 현재 대기열 비율(class2) = 알 수 없음. 이 문제를 해결해야 합니다. 대기열-버퍼 비율이 이 클래스에 대해 구성되지 않았습니니다.
- Current Queue Ratio(class3) = 10% - 정책 맵에서 대기열-버퍼 비율 10
- 현재 대기열 비율(class4) = 10% - 대기열 버퍼 비율이 10인 정책 맵에 구성됨
- Current Queue Ratio(class-default) = 40% - 정책 맵에서 대기열-버퍼 비율 40

남은 큐 버퍼 비율 계산:

암시적 비율 왼쪽 = (100 - 명시적으로 구성된 비율의 합계).

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

현재 대기열 비율(class2) = 20

최종 대기열 버퍼 할당 계산

Class1 = [기본 버퍼] 1200 x [현재 대기열 비율(class1)] (20/100) = 240 - 우선순위 대기열, 추가 계산 없음

클래스2 = [

[Base Buffer] 1200 x [Current Queue Ratio(class2)] (10/100) = 240 - 이 대기열에 대한 기본 버퍼 할당이지만, 우선 순위가 아닌 대기열에 대한 softmax를 얻으려면 이를 곱해야 합니다.

[이 대기열에 대한 기본 버퍼 할당] 120 x [숨겨진 승수] 4 x [사용자 승수] (100/100) = 960

]

남아 있는 대기열에 대해 반복:

클래스 3 = [

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

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

]

클래스 4 = [

$1200 \times (10/100) = 120$   
 $120 \times 4 \times (100/100) = 480$   
]

클래스 class-default = [

$1200 \times (40/100) = 480$   
 $600 \times 4 \times (100/100) = 1920$   
]

예측과 비교하여 적용된 test3 policy-map의 결과:

<#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 G1b1SMin 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>

최종 결과: Q0 - 하드웨어: 240, 소프트맥스: 240. Q1 - 소프트맥스: 960, Q2 - 소프트맥스: 480, Q3 - 소프트맥스: 480, Q4 - 소프트맥스: 480

### 시나리오 3.1: 다중 대기열 및 다중 암시적 비율에 대한 소프트 및 하드 버퍼 결정

이 시나리오에서는 5개의 대기열이 구성되고 2개의 대기열에 대기열-버퍼 비율이 구성되지 않습니다.

버퍼 할당을 결정하려면 시나리오 2와 동일한 논리가 계속 적용되지만, 암시적 비율 잔차를 대기열-버퍼 비율이 없는 총 암시적 대기열/대기열 수로 나누어야 합니다

이 시나리오에 사용 중인 정책 맵:

```
<#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
```

값의 요약:

- 기본 버퍼 = 1200 - 이 숫자를 찾는 방법은 시나리오 1에 나와 있습니다.
- 사용자 승수 = 100% - 기본값
- 숨겨진 승수 = 4 - 기본값 및 사용자 구성 불가
- Current Queue Ratio(class1) = 20% - 정책 맵에서 대기열-버퍼 비율 20
- 현재 대기열 비율(class2) = 알 수 없음. 이 문제를 해결해야 합니다. 대기열-버퍼 비율이 이 클래스에 대해 구성되지 않았습니다.
- 현재 대기열 비율(class3) = 알 수 없음. 이 문제를 해결해야 합니다. 대기열-버퍼 비율이 이 클래스에 대해 구성되지 않았습니다.
- 현재 대기열 비율(class4) = 10% - 대기열 버퍼 비율이 10인 정책 맵에 구성됨

- Current Queue Ratio(class-default) = 40% - 정책 맵에서 큐 버퍼 비율 40으로 구성됨

남은 큐 버퍼 비율 계산:

암시적 비율 왼쪽 = (100 - 명시적으로 구성된 비율의 합계).

암시적 대기열 수 = 2(class2 및 class3에 대기열 버퍼 비율이 정의되지 않음)

구성된 비율의 합계 = 20+40+10 = 70

암시적 비율 남은 값 = 100 - 70 = 30

암시적 대기열 비율 할당 = [암시적 비율 왼쪽] 30 / [암시적 대기열 수] 2 = 15

최종 대기열 버퍼 할당 계산:

클래스1 =

[기본 버퍼] 1200 x [현재 대기열 비율(class1)] (20/100) = 240 - 우선순위 대기열, 추가 계산 없음

클래스2 =

[Base Buffer] 1200 x [Implicit Queue Ratio allocation] (15/100) = 180 - 클래스 2에는 정의된 대기열-버퍼 비율이 없으므로 명시적 대기열의 나머지 대기열-버퍼 비율은 암시적 대기열 간에 공유됩니다.

[이 대기열에 대한 기본 버퍼 할당] 180 x [숨겨진 승수] 4 x [사용자 승수] (100/100) = 720

남아 있는 대기열에 대해 반복:

클래스 3 = [

1200 x (15/100) = 180

120 x 4 x (100/100) = 720

]

클래스 4 = [

$1200 \times (10/100) = 120$   
 $120 \times 4 \times (100/100) = 480$   
]

클래스 class-default = [

$1200 \times (40/100) = 480$   
 $600 \times 4 \times (100/100) = 1920$   
]

예측과 비교하여 적용된 test4 정책 맵의 결과:

<#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>

```



참고: 암시적 대기열 비율 할당의 결과가 정수가 아니면 동일한 공유가 가능하지 않습니다. 결과는 정책 맵의 초기 대기열에 대해 반올림되고 이후 대기열에 대해 반올림됩니다. 할당된 대기열 버퍼 비율의 최종 합계는 100으로 남아 있지만 암시적 대기열은 방금 설명한 정수 결과 요구 사항으로 인해 항상 동일한 할당을 얻지 못합니다.

## 시나리오 4: 대기열 간 대기열 버퍼 비율 합계가 100보다 작음

이 시나리오에서는 5개의 대기열이 구성되는데, 모두 대기열-버퍼 비율입니다. 클래스 간 큐-버퍼 비율의 합계가 100보다 작습니다.

이 경우 할당되지 않은 버퍼 비율은 클래스 간에 균등하게 분산됩니다.

이전 시나리오와 마찬가지로, 나머지 대기열-버퍼 비율의 분할 결과가 정수가 아닌 경우 각 대기열에 대한 최종 할당은 반올림되거나 반올림되며 구성된 대기열 버퍼 비율에 추가됩니다.

이 시나리오에 사용 중인 정책 맵:

```

<#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
```

값의 요약:

- 기본 버퍼 = 1200 - 이 숫자를 찾는 방법은 시나리오 1에 나와 있습니다.
- 사용자 승수 = 100% - 기본값
- 숨겨진 승수 = 4 - 기본값 및 사용자 구성 불가
- Current Queue Ratio(class1) = 10% - 정책 맵에서 대기열-버퍼 비율 10
- Current Queue Ratio(class2) = 10% - 정책 맵에서 큐-버퍼 비율 10으로 구성됨
- 현재 대기열 비율(class3) = 10% - 대기열 버퍼 비율 10을 사용하여 정책 맵에 구성됨
- 현재 대기열 비율(class4) = 10% - 대기열 버퍼 비율이 10인 정책 맵에 구성됨
- Current Queue Ratio(class-default) = 12% - 정책 맵에서 다음과 같이 구성됨 대기열-버퍼 비율 12
- 총 대기열 수 = 5

구성된 비율의 합계 =  $10 + 10 + 10 + 12 = 52$

남은 버퍼 비율 =  $100\% - 52\% = 48\%$

[Buffer Ratio Left Tover] 48% / [Total Number of Queues] 5 = 대기열당 9.6% 추가됨 - 정수가 아니므로 대기열에 대한 최종 응용 프로그램은 대기열당 반올림하거나 축소해야 합니다.

시스템에서 사용하는 최종 대기열-버퍼 비율 번호를 가져오려면 이미 구성된 대기열 버퍼 비율에 9 또는 10을 추가해야 합니다.

정책 맵에서 더 높은 클래스는 반올림된 값 10을 받습니다. 정책 맵에서 더 낮은 클래스는 반올림된 값 9를 받습니다.

최종 대기열 버퍼 할당 계산

버퍼 비율 남은 메모리 = 48

Class1 = [Base Buffer] x ([Current Queue ratio(class1) + 남은 공유 버퍼 비율의 반올림된 값])

Class1 = 1200 x ((10% + 10%)/100) = 240 - 우선순위 대기열, 추가 계산 없음

남은 버퍼 비율 = (48 - 10) = 38

Class2 = [Base Buffer] x ([Current Queue ratio(class2) + 남은 공유 버퍼 비율의 반올림된 값])

Class2 = 1200 x ((10% + 10%)/100) = 240 - 우선순위 대기열이 아니므로 사용자 및 시스템 승수로 계속 곱합니다.

Class2 = [이 대기열에 대한 기본 버퍼 할당] 240 x [숨겨진 승수] 4 x [사용자 승수] (100/100) = 960  
- 이 대기열에 대한 소프트맥스 결과

버퍼 비율 남음 = (38 - 10) = 28

남아 있는 대기열에 대해 반복:

클래스 3 = [

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

120 x 4 x (100/100) = 960

]

남은 버퍼 비율 = (28 - 10) = 18

클래스 4 = [

1200 x ((10+9)/100) = 240

120 x 4 x (100/100) = 912

]

버퍼 비율 남음 = 9

클래스 class-default= [

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

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

]

버퍼 비율 남은 메모리 = 0

예측과 비교하여 적용된 test5 정책 맵의 결과:

<#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 G1b1SMin PortStEnd QEnable

	DTS	Hardmax	Softmax	PortSMin	G1b1SMin	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>

## 시나리오 5: 우선순위 레벨 2 클래스의 소프트 및 하드 버퍼 결정

이 시나리오에서 정책 맵의 클래스는 우선순위 레벨 2로 구성됩니다.

softmax가 승수의 영향을 받지 않고 hardmax로 설정된 우선순위 레벨 1과 달리 우선순위 레벨 2는 softmax가 곱해지는 동시에 하드 버퍼(hardmax) 할당을 가질 수 있도록 합니다.

이 시나리오에 사용 중인 정책 맵:

```
<#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
```

적용된 test6 정책 맵의 결과:

```
<#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 G1b1SMin 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>
```

이전에 표시된 출력에서 두 번째 대기열 softmax는 첫 번째 대기열의 softmax가 4입니다. 이는 우선 순위 수준 1 softmax가 특별히 시스템 softmax 승수의 영향을 받지 않지만 우선 순위 수준 2가 영향을 받기 때문입니다.

사용자 소프트맥스 멀티플라이어를 구성할 경우 우선순위 레벨 2 대기열만 영향을 받습니다.

```
<#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 GblSMin 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>
```

## 큐 제한

대기열 제한 컨피그레이션은 최종 대기열 버퍼 할당에 영향을 미칩니다

대기열 버퍼 할당에 영향을 주는 기본 메커니즘은 MQC 정책 맵에서 대기열당 추가된 대기열 버퍼 비율 컨피그레이션입니다.

그러나 대기열 버퍼 할당은 다른 컨피그레이션의 영향을 받습니다.

Queue-limit은 이 문서에서 다루지 않는 WTD(Weighted Tail Drop)를 통해 특정 트래픽 클래스를 삭제하는 임계값을 정의합니다.

특정 상황에서 queue-limit은 소프트 버퍼 대기열에 대한 시스템 숨겨진 승수를 수정합니다. 이는 queue-limit이 적용되는 해당 대기열의 전체 소프트 버퍼 할당에 영향을 미칩니다.

먼저, queue-limit은 클래스당 최대 3번 구성할 수 있음을 이해합니다. 이는 DSCP 또는 CoS 단위로 WTD에 대해 최대 3개의 임계값을 정의합니다.

다음 출력에서는 두 개의 임계값만 정의됩니다.

정책 맵에 두 개의 queue-limits 적용:

```
<#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
```

버퍼 할당 결과를 관찰합니다.

```
<#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 GblSMin 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>
```

다음 예에서는 세 번째 queue-limit 컨피그레이션이 class class-default에 추가됩니다.

버퍼 할당 결과를 관찰합니다.

```
<#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 GlbSMin 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>
```

세 번째 대기열 제한 컨피그레이션이 대기열에 추가되면 해당 대기열에 대해 시스템 숨겨진 소프트 버퍼 승수 400%가 비활성화됩니다. 그러나 이 대기열은 여전히 사용자가 구성한 qos 대기열-softmax-multiplier <percent>를 고려합니다.

## 관련 정보

- [Cisco Catalyst 9000 스위칭 플랫폼: QoS 및 큐잉 백서](#)
- [Cisco 기술 지원 및 다운로드](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.