

Inzicht in wachtrijbuffertoewijzing op Catalyst 9000-Switches

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Verwante producten](#)

[Achtergrondinformatie](#)

[Terminologie](#)

[buffers](#)

[Switch Queue Buffer Algorithm](#)

[Configureren](#)

[Scenario 1: Bepaal de basisbufferwaarde voor een bepaalde poortsnelheid](#)

[Scenario 2: Bepaal de zachte en harde buffer voor meerdere wachtrijen en expliciete ratio's](#)

[Scenario 2.1: Door gebruiker geconfigureerde multiplier wijzigen in scenario 2](#)

[Scenario 3: Bepaal een zachte en harde buffer voor meerdere wachtrijen en een impliciete verhouding](#)

[Scenario 3.1: Bepaal de zachte en harde buffer voor meerdere wachtrijen en meerdere impliciete ratio's](#)

[Scenario 4: Som van wachtrijbuffers De verhouding tussen wachtrijen is minder dan 100](#)

[Scenario 5: Bepaal de zachte en harde buffer van een prioriteitsniveau 2-klasse](#)

[wachtrijlimiet](#)

[Gerelateerde informatie](#)

Inleiding

In dit document wordt beschreven hoe u de toewijzing van de wachtrijbuffer aan verkeerswachtrijen op Switches uit de Catalyst 9000-reeks kunt voorspellen.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Quality of Service (QoS) concepten en processen voor hoe pakketten worden gemarkeerd, in de wachtrij staan en gepland
- Cisco MQC QoS-configuratie

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

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

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Verwante producten

Dit document kan ook worden gebruikt voor de volgende hardware- en softwareversies:

- Catalyst 9200 - 9600 reeks Switches
- Katalysator 9300X & 9400X
- Alle Cisco IOS XE-softwareversies



Opmerking: Dit document is niet van toepassing op de 9500X of 9600X omdat deze een andere ASIC- en QoS-architectuur gebruiken.

Achtergrondinformatie

Voor een technisch overzicht van QoS op Catalyst 9000 Series Switches, zie: [Catalyst 9000 QoS en Queueing White Paper](#).

Vaak moet de toewijzing van de buffer worden afgestemd als reactie op ongewenste outputdalingen voor een bepaalde verkeersklasse. Meer informatie over het diagnosticeren en oplossen van problemen met uitvoerdruppels op Switches uit de Catalyst 9000-reeks is te vinden

in dit artikel: [Problemen oplossen met uitvoerdruppels op Catalyst 9000-Switches](#)

Terminologie

QoS	Quality-of-Service	Een concept / groep van gerelateerde functies met betrekking tot het classificeren, markeren, in de wachtrij plaatsen en plannen van verkeer in en uit een netwerkapparaat
DSCP	Differentiated Services Code Point	Een verkeersclassificatiemechanisme in de IP-header van een pakket
CoS	Klasse van service	Een verkeersclassificatiemechanisme in de Ethernet-frameheader van een pakket
AAS	toegangscontrole-ingang	Eén regel of regel in een toegangscontrolelijst (ACL)
ACL	Toegangscontrolelijst	Een groep Access Control Entries (ACE's) die door verschillende functies worden gebruikt om verkeer te matchen en actie te ondernemen
ASIC	toepassings specifieke geïntegreerde schakeling	Een computerchip die is ontworpen om een specifieke taak of een reeks taken met hoge efficiëntie uit te voeren.
UADP	gegevensvlak voor Unified Access	Cisco ASIC wordt gebruikt in Catalyst 9000 Series-Switches om veel netwerkpakketverwerkingstaken uit te voeren.
PBC	pakketbuffercomplex	Cisco UADP ASIC-subsysteem dat dient als centrale pakketbuffer voor het verwerken, in de wachtrij plaatsen en plannen van pakketten.
AQM	Active Queue Management	Cisco UADP ASIC-subsysteem dat verkeerswachtrij beheert en acties voor netwerkpoorten plant.
DTS	Dynamische drempel en schaal	Cisco UADP ASIC-technologie waarmee buffers dynamisch worden aangepast en geschaald over poorten om het hardwaregebruik te optimaliseren

buffers

Als concept worden buffers gebruikt om voorbijgaande uitbarstingen van gegevens te absorberen, wanneer de gegevens worden geschakeld of gerouteerd naar een poort die de mogelijkheid van poorten om gegevens op de draad te plaatsen overschrijdt. Een poort heeft een vaste snelheid waarmee gegevens worden verzonden en uit de wachtrij worden verwijderd. Een buffer, conceptueel, is gewoon een plek om op te slaan, of in de wachtrij, gegevens totdat het wordt verzonden uit de interface.

Op Catalyst 9000 Series Switches, het woord buffer heeft twee toepassingen. De systeembuffer als geheel wordt ook wel het packet buffer complex (PBC) van de ASIC genoemd. Het woord buffers kan ook verwijzen naar kleine eenheid van de PBC. Een buffer wordt toegewezen aan poorten op basis van een wachtrij. Met andere woorden, aan een poortwachtrij wordt een hoeveelheid kleine individuele buffers toegewezen uit de totale systeembuffer.

Op Cisco UADP ASIC-gebaseerde platforms bevat één buffer maximaal 256 bytes aan gegevens en worden buffers aan elkaar gekoppeld om frames groter dan 256 bytes weer te geven.

Switch Queue Buffer Algorithm

- Aan een poort is een hard gecodeerde basisbufferwaarde toegewezen die door de gebruiker of het systeem kan worden geschaald (maar het basisnummer kan niet worden gewijzigd, wat belangrijk is om te begrijpen hoe u de resultaten kunt voorspellen)
- De basisbufferwaarde voor een bepaald poorttype is afhankelijk van de specifieke combinatie van hardware (hardwaremodel), softwareversie en poortsnelheid. Een Gigabit Ethernet-poort op Catalyst C9500-25Y4C begint bijvoorbeeld met een andere basisbufferwaarde dan een Gigabit Ethernet-poort op Catalyst 9200L
- Deze basiswaarde wordt verdeeld over N aantal wachtrijen - op basis van de configuratie.

De uiteindelijke berekening van de beschikbare buffer per wachtrij wordt beïnvloed door deze factoren:

- Aantal geconfigureerde wachtrijen
- De wachtrijbufferverhouding, een opdracht waarmee u kunt aanpassen hoe de basisbuffer wordt verdeeld over geconfigureerde poortwachtrijen.
- Een door de gebruiker geconfigureerde multiplier voor de basisbuffer (geconfigureerd als queue-softmax-multiplier <percent>)
- Een door het systeem gedefinieerde basisbuffermultiplier (niet door de gebruiker configureerbaar, 4x of 400%) - hoewel dit standaard wordt toegepast, zijn er enkele

omstandigheden waarin het is verwijderd, dus het is belangrijk om deze 4x-waarde te begrijpen om het toewijzingsalgoritme te voorspellen

- Beide multipliers hebben alleen invloed op zogenaamde zachte buffers

Zachte buffers zijn buffers die over poorten worden gedeeld. Deze buffers worden zacht genoemd omdat ze niet gegarandeerd zijn voor de haven.

Het systeem wijst opzettelijk zachte buffers toe. Hierdoor kan elke poort een groot aantal buffers gebruiken als dat nodig is, maar omdat meer poorten buffers nodig hebben, worden alle poorten en wachtrijen dynamisch en eerlijk verkleind als onderdeel van het Cisco UADP ASIC DTS-proces.

Kortom, zachte buffers - in uitgangen aangeduid als softmax, is een opportunistische maximale waarde. Een poort gebruikt alleen de volledige softmax als die hoeveelheid buffer beschikbaar is uit de totale systeembuffer. Aangezien de vraag naar bufferruimte stijgt in andere poorten en wachtrijen, is de maximale buffer die beschikbaar is voor de poort minder.

Harde buffers zijn buffers die expliciet zijn gereserveerd voor een poort en worden niet beïnvloed door het DTS-proces. Omdat harde buffers gegarandeerde buffers zijn, is de som van de totale harde buffers die aan poorten zijn toegewezen nooit groter dan het PBC-segment dat aan deze harde buffers is toegewezen.

De mechanismen die de actieve schaal van zachte buffer regelen, staan bekend als DTS (Dynamic Threshold and Scale), beschreven in het [witboek](#) van de [Catalyst 9000-serie](#).

De grootte van de PBC-segmenten die zijn gewijd aan harde en zachte buffers in de wachtrij verandert dynamisch terwijl u het systeem configureert en kan worden gezien als AQM GlobalSoftLimit en GlobalHardLimit in deze uitvoer:

```
<#root>
```

```
C9500#
```

```
show platform hardware fed active qos queue stats interface twel/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 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:4 - 1800 BufferSharing:Disabled
DTS

Hardmax

Softmax

```
PortSMin GblSMin 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 twel1/0/1

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



Opmerking: Let op de toename van GlobalHardLimit en proportionele afname van GlobalSoftLimit.

Verder, wanneer u prioriteitsniveau 1 configureert, wordt de softmax voor die wachtrij statisch ingesteld op exact gelijk aan de hardmax. U kunt de harde buffer alleen wijzigen voor een wachtrij van prioriteitsniveau 1.

De verandering in GlobalHardLimit en GlobalSoftLimit is gelijk aan 720. Dit is ook gelijk aan de verandering in hardmax na configuratie.

In de scenario's in dit document wordt uitgelegd hoe u softmax- en hardmax-toewijzingen kunt berekenen en voorspellen voor meerdere configuraties met beleidskaarten.

Configureren

Scenario 1: Bepaal de basisbufferwaarde voor een bepaalde poortsnelheid

Een laatste bufferwaarde van een wachtrij is gedeeltelijk een functie van een basiswaarde die voor het eerst wordt toegewezen aan een wachtrij. Dit wordt later vermenigvuldigd bij zachte buffers.

De vermenigvuldigingsfactoren, in combinatie met andere impliciete gedragingen, maken het bepalen van een uiteindelijke waarde voor een bepaalde wachtrij, met een bepaalde configuratie, een uitdaging.

De eerste stap om de resulterende wachtrijbuffertoewijzing te verduidelijken, is het bepalen van de basisbufferwaarde.

Maak hiervoor gebruik van een prioriteitswachtrij, die een harde buffer ontvangt die direct evenredig is met het aantal wachtrijen of de geconfigureerde wachtrijbufferverhouding.

Met een specifieke configuratie kunt u expliciet de hoeveelheid basisbuffer afleiden die aan een bepaalde poortsnelheid is toegewezen.

Alle buffers configureren en toewijzen aan één enkele, niet-vermenigvuldigde wachtrij (een wachtrij van prioriteitsniveau 1)

In dit voorbeeld wordt de klasse-default gebruikt om al het verkeer aan te passen, omdat er geen andere klassen zijn geconfigureerd.

```
<#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
```

De configuratie in het vorige voorbeeld voert de volgende acties uit:

- Het prioriteitsniveau 1 vertelt het systeem om een strikt schema uit te voeren voor deze

wachtrij / klasse en een harde buffer toe te wijzen aan deze wachtrij / klasse.

- Hoe pakketten worden gepland valt buiten het bereik van dit document, maar een strikte wachtrij wordt over het algemeen vóór alle anderen onderhouden.
- De queue-buffers ratio 100 wijst 100/100 of 100% van de beschikbare basisbuffer toe aan deze queue / klasse.

In een beleidskaart met meer dan één klasse kunt u 100% van de buffer niet aan één klasse toewijzen. Je bent verplicht om minimaal 1/100 of 1% toe te wijzen aan elke klasse.

In een beleid met slechts één klasse hebt u slechts één klasse en kunt u er alle buffer aan toewijzen.

Zoals eerder opgemerkt, krijgt een prioriteitswachtrij harde buffers die gelijk zijn aan de verdeling van de basisbuffer volgens de geconfigureerde wachtrijbufferverhouding. Een harde buffer is niet onderhevig aan een vermenigvuldigingsfactor.

Harde buffer wordt waargenomen in uitgangen onder een kolom met de titel Hardmax.

Nu heb je een enkele verkeersklasse met buffers die niet onderhevig zijn aan een multiplier. Hiermee kunt u de toewijzing van de basisbuffer voor deze poortsnelheid expliciet afleiden (en alleen deze poortsnelheid op dit platform, anderen verschillen), omdat de basisbuffer en hardmax gelijk zijn.

Basisbuffer = ?

Wachtrijverhouding 1 = 100/100 = 1

Hardmax voor deze wachtrij = basisbuffer x wachtrijverhouding 1

$X = Y \times 1$

$X / 1 = Y$

$X = Y$

$X = Y = \text{Hardmax} = \text{basisbuffer} = 1200$ (zie voorbeeld 2).

In dit voorbeeld wordt policy-map test1 toegepast op een interface als een output service-policy

```
<#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  
  
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>
```

Zoals getoond, is de hardmax voor deze prioriteitswachtrij met 100% buffer toegewezen aan het, is 1200.

Aangezien hardmax een volledig ongevermenigvuldigde / niet-geschaalde waarde is en 100% van de buffer is geconfigureerd voor deze wachtrij, is de basisbuffertoewijzing voor dit specifieke model van switch, softwareversie en specifieke poortsnelheid 1200.

Andere poortsnelheden op dezelfde switch en andere modellen switches voor dezelfde poortsnelheid ontvangen verschillende basisbuffertoewijzingen. Deze basistoewijzing kan niet door de gebruiker worden geconfigureerd en moet worden afgeleid via observatie.

Verdere scenario's in dit document beginnen allemaal met dezelfde switch, software en poortsnelheid. Ze gaan dus allemaal uit van een basistoewijzing van 1200 voor berekeningen om de definitieve buffertoewijzing te bepalen.



Opmerking: Merk op dat de softmax in het vorige voorbeeld ook 1200 is.

Een prioriteitsniveau 1 wachtrij heeft een softmax die exact gelijk is aan de hardmax. Dit is bedoeld en kan niet door de gebruiker worden geconfigureerd.

Verder wordt dit specifieke geval van softmax-toewijzing niet beïnvloed door softmax-multipliers die later worden weergegeven. Alleen een prioriteitsniveau 1 wachtrij heeft dit gedrag voor softmax, wat bedoeld is.

Scenario 2: Bepaal de zachte en harde buffer voor meerdere wachtrijen en expliciete ratio's

In dit scenario wordt een extra wachtrij toegevoegd. Deze wachtrij maakt geen gebruik van prioriteitsniveau 1, en dus softmax schalen met multipliers.

De ene multiplier is door de gebruiker geconfigureerd en de andere een verborgen / niet configureerbare multiplier.

Combineer deze multipliers met de basisbuffer die is afgeleid voor deze poort, in dit geval 1200 volgens scenario 1.

Algoritmisch:

Huidige wachtrijverhouding = wachtrijbufferverhouding voor de te voorspellen wachtrij / klasse
Verborgen multiplier = 400%

User Multiplier = Percentage waarde dat u configureert in queue-softmax-multiplier <percentage>.
Standaard is 100%

$\text{Softmax} = (\text{Basisbuffer} \times (\text{huidige wachtrijverhouding} / 100)) \times \text{verborgen multiplier} \times (\text{gebruikersmultiplier} / 100)$

```
<#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
```

Overzicht van waarden:

- Basisbuffer = 1200 - Hoe dit getal te vinden wordt getoond in Scenario 1
- User Multiplier = 100% - Standaardwaarde, deze is nog niet gewijzigd
- Verborgene multiplier = 4 - standaard en niet door de gebruiker configureerbaar
- Huidige Queue Ratio(class1) = 50% - Geconfigureerd in de beleidskaart met queue-buffers ratio 50
- Huidige Queue Ratio (class-default) = 50% - Geconfigureerd in de policy map met queue-buffers ratio 50

Klasse 1-buffertoewijzing bepalen:

Omdat klasse 1 een prioriteitswachtrij is, ontvangt het hardmax (harde buffer) en een speciaal geval van softmax dat niet wordt beïnvloed door multipliers.

Klasse1 hardmax = (basisbuffer x verhouding huidige wachtrij(klasse1) / 100)

Class1 hardmax = 1200 x (50/100) = 600 - vanwege een speciaal geval van een prioriteitswachtrij, stop alle wiskunde, wijs uw resultaat toe aan hardmax. Softmax is gelijk aan Hardmax als regel voor prioriteitsniveau 1.

Standaardbuffertoewijzing voor klasse bepalen:

Class-default = (Basisbuffer x (Huidige wachtrijverhouding (class-default / 100)) x Verborgene multiplier x (Gebruikersmultiplier / 100)

Class-default = [

[basisbuffer] 1200 x [Huidige wachtrijverhouding] (50/100) = 600

[previous result] 600 x [Verborgene vermenigvuldiger] 4 x [Vermenigvuldiger gebruiker] (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>
```

```
Eindresultaat: Q0 - Hardmax: 600 Softmax: 600. Q1 - Softmax: 2400
```

Scenario 2.1: Door gebruiker geconfigureerde multiplier wijzigen in scenario 2

Dit scenario begint hetzelfde als scenario 2, behalve dat u nu queue-softmax-multiplier 1200 configureert.

Dit vermenigvuldigt de softmax buffers in de huidige configuratie met 1200%, oftewel een factor 12.

Overzicht van waarden:

- Basisbuffer = 1200 - Hoe dit getal te vinden wordt getoond in Scenario 1
- User Multiplier = 1200% - Dit is de waarde die in dit scenario is gewijzigd, via qos queue-softmax-multiplier 1200
- Verborgen multiplier = 4 - standaard en niet door de gebruiker configureerbaar
- Huidige Queue Ratio(class1) = 50% - Geconfigureerd in de beleidskaart met queue-buffers ratio 50
- Huidige Queue Ratio(class-default) = 50% - Geconfigureerd in de beleidskaart met queue-buffers ratio 50

Klasse 1-buffertoewijzing bepalen:

Omdat klasse 1 een prioriteitswachtrij is, ontvangt het hardmax (harde buffer) en een speciaal geval van softmax dat niet wordt beïnvloed door multipliers.

Klasse1 hardmax = (basisbuffer x verhouding huidige wachtrij(klasse1) / 100)

Class1 hardmax = $1200 \times (50/100) = 600$ - vanwege een speciaal geval van een prioriteitswachtrij, stop alle wiskunde, wijs uw resultaat toe aan hardmax. Softmax is gelijk aan Hardmax als regel voor prioriteitsniveau 1.

Standaardbuffertoewijzing voor klasse bepalen:

Class-default = (Basisbuffer x (Huidige wachtrijverhouding (class-default / 100)) x Verborgen multiplier x (Gebruikersmultiplier / 100)

Class-default = [

[basisbuffer] 1200 x [Huidige wachtrijverhouding] (50/100) = 600
[previous result] 600 x [Verborgen vermenigvuldiger] 4 x [Vermenigvuldiger gebruiker] (1200/100)
= 28800

]

Configureer qos queue-softmax-multiplier 1200 en bekijk wijzigingen in softmax (softmax is een maximale bufferwaarde voor die wachtrij, dynamisch geschaald op basis van het huidige algehele buffergebruik):

```
<#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   GlbISMin   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>

Eindresultaat: Q1 - Hardmax: 600, Softmax: 600. Q2 - Softmax: 28800

Scenario 3: Bepaal een zachte en harde buffer voor meerdere wachtrijen en een impliciete verhouding

In dit scenario worden vijf wachtrijen geconfigureerd, maar slechts vier hebben een expliciet gedefinieerde verhouding tussen wachtrijbuffers. De buffer die aan die wachtrijen is toegewezen, is dezelfde als de vorige voorbeelden.

De wachtrij die niet is geconfigureerd, ontvangt het verschil tussen de som van alle geconfigureerde wachtrijbuffers en 100.

Som van expliciet geconfigureerde ratio's = (Q0-bufferverhouding) + (Q1-bufferverhouding) ... (uiteindelijke bufferverhouding) - maximaal 8 wachtrijen worden ondersteund op Switches uit de Catalyst 9000-reeks, zodat u maximaal 8 ratio's kunt toevoegen

Impliciete ratio = (100 - Som van expliciet geconfigureerde ratio's).

Impliciete ratio left-over is de waarde die wordt toegewezen aan een wachtrij waarvan de queue-buffers ratio niet is geconfigureerd.

Policy-map in gebruik voor dit scenario:

```
<#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
```

Overzicht van waarden:

- Basisbuffer = 1200 - Hoe dit getal te vinden wordt getoond in Scenario 1
- User Multiplier = 100% - Standaardwaarde, deze is nog niet gewijzigd
- Verborgen multiplier = 4 - standaard en niet door de gebruiker configureerbaar
- Huidige Queue Ratio(class1) = 20% - Geconfigureerd in de beleidskaart met queue-buffers ratio 20
- Huidige wachtrijverhouding(class2) = Onbekend, u moet dit oplossen - wachtrijbufferverhouding is niet geconfigureerd voor deze klasse
- Huidige Queue Ratio(class3) = 10% - Geconfigureerd in de beleidskaart met queue-buffers ratio 10
- Huidige Queue Ratio (klasse 4) = 10% - Geconfigureerd in de policy map met queue-buffers ratio 10
- Huidige Queue Ratio(class-default) = 40% - Geconfigureerd in de beleidskaart met queue-buffers ratio 40

Bereken de bufferverhouding van de wachtrij die overblijft:

Impliciete ratio = (100 - Som van expliciet geconfigureerde ratio's).

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

Huidige Queue Ratio(klasse2) = 20

Definitieve wachtrijbuffertoewijzing berekenen

Klasse1 = [Basisbuffer] 1200 x [Huidige wachtrijverhouding(klasse1)] (20/100) = 240 - Prioritaire wachtrij, geen verdere berekening

Klasse 2 = [

[Base Buffer] $1200 \times [\text{Current Queue Ratio(class2)}] (10/100) = 240$ - base buffer toewijzing voor deze wachtrij, maar deze moet worden vermenigvuldigd om softmax te krijgen voor een wachtrij zonder prioriteit

[Toewijzing basisbuffer voor deze wachtrij] $120 \times [\text{Verborgten vermenigvuldiger}] 4 \times [\text{Vermenigvuldiger gebruiker}] (100/100) = 960$

]

Herhaal dit voor wachtrijen die blijven staan:

Klasse 3 = [

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

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

]

Klasse 4 = [

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

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

]

Class-default = [

$1200 \times (40/100) = 480$

$600 \times 4 \times (100/100) = 1920$

]

Resultaat van de toegepaste test3-beleidskaart ten opzichte van de voorspelling:

```
<#root>
```

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

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

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

```
service-policy output test3
```

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 - 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>
```

Eindresultaat: Q0 - Hardmax: 240, Softmax: 240. Q1 - Softmax: 960, Q2 - Softmax: 480, Q3 - Softmax: 480, Q4 - Softmax: 480

Scenario 3.1: Bepaal de zachte en harde buffer voor meerdere wachtrijen en meerdere impliciete ratio's

In dit scenario worden vijf wachtrijen geconfigureerd en twee wachtrijen hebben geen wachtrijbufferverhouding geconfigureerd.

Om buffertoewijzing te bepalen, gaat dezelfde logica uit Scenario 2 verder, maar u moet ook de impliciete overgebleven ratio delen door het totale aantal impliciete wachtrijen / wachtrijen die geen wachtrijbufferverhouding hebben

Policy-map in gebruik voor dit scenario:

```
<#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
```

Overzicht van waarden:

- Basisbuffer = 1200 - Hoe dit getal te vinden wordt getoond in Scenario 1
- Gebruikersmultiplier = 100% - Standaardwaarde
- Verborggen multiplier = 4 - Standaard en niet door de gebruiker configureerbaar
- Huidige Queue Ratio(class1) = 20% - Geconfigureerd in de beleidskaart met queue-buffers ratio 20
- Huidige wachtrijverhouding(class2) = Onbekend, u moet dit oplossen - wachtrijbufferverhouding is niet geconfigureerd voor deze klasse
- Huidige wachtrijverhouding(class3) = Onbekend, u moet dit oplossen - wachtrijbufferverhouding is niet geconfigureerd voor deze klasse
- Huidige Queue Ratio (klasse 4) = 10% - Geconfigureerd in de policy map met queue-buffers ratio 10
- Huidige Queue Ratio (class-default) = 40% - Geconfigureerd in de policy map met queue-buffers ratio 40

Bereken de bufferverhouding van de wachtrij die overblijft:

Impliciete ratio = (100 - Som van expliciet geconfigureerde ratio's).

Aantal impliciete wachtrijen = 2 (klasse 2 en klasse 3 hebben geen wachtrijbufferverhouding gedefinieerd)

Som van geconfigureerde verhoudingen = 20 + 40 + 10 = 70

Impliciete ratio overgebleven = 100 - 70 = 30

Impliciete wachtrijverhouding toewijzing = [Impliciete overgebleven ratio] 30 / [Aantal impliciete wachtrijen] 2 = 15

Bereken de uiteindelijke toewijzing van de wachtrijbuffer:

Klasse1 =

[Base Buffer] $1200 \times [\text{Current Queue Ratio(class1)}] (20/100) = 240$ - Prioriteitswachtrij, geen verdere berekening

Klasse2 =

[Basisbuffer] $1200 \times [\text{toewijzing van impliciete wachtrijverhouding}] (15/100) = 180$ - Omdat klasse 2 geen gedefinieerde wachtrijbufferverhouding heeft, wordt de rest van de wachtrijbufferverhouding van expliciete wachtrijen gedeeld tussen impliciete wachtrijen.

[Toewijzing basisbuffer voor deze wachtrij] $180 \times [\text{Verborgen vermenigvuldiger}] 4 \times [\text{Vermenigvuldiger gebruiker}] (100/100) = 720$

Herhaal dit voor wachtrijen die blijven staan:

Klasse 3 = [

$1200 \times (15/100) = 180$
 $120 \times 4 \times (100/100) = 720$
]

Klasse 4 = [

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

Class-default = [

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

Resultaat van de toegepaste test4-beleidskaart ten opzichte van de voorspelling:

<#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 GlbSMin 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>
```



Opmerking: als het resultaat van de impliciete toewijzing van de wachtrijverhouding geen geheel getal is, is een gelijk aandeel niet mogelijk. Het resultaat rondt naar boven voor wachtrijen eerder in de policy-map, en naar beneden af voor latere wachtrijen. De uiteindelijke som van de toegewezen wachtrijbufferverhouding blijft 100, maar impliciete

wachtrijen krijgen niet altijd dezelfde toewijzing vanwege de zojuist beschreven vereiste van het gehele resultaat.

Scenario 4: Som van wachtrijbuffers De verhouding tussen wachtrijen is minder dan 100

In dit scenario worden vijf wachtrijen geconfigureerd, allemaal met queue-buffers ratio. De totale som van de queue-buffers ratio over de klassen is minder dan 100.

In dit geval wordt de niet-toegewezen bufferverhouding gelijkmatig over de klassen verdeeld.

Vergelijkbaar met het vorige scenario, als het verdeelde resultaat van de overgebleven wachtrijbufferverhouding geen geheel getal is, wordt de uiteindelijke toewijzing aan elke wachtrij naar boven of naar beneden afgerond en toegevoegd aan de wachtrijbufferverhouding die u hebt geconfigureerd.

Policy-map in gebruik voor dit scenario:

```
<#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

Overzicht van waarden:

- Basisbuffer = 1200 - Hoe dit getal te vinden wordt getoond in Scenario 1
- Gebruikersmultiplier = 100% - Standaardwaarde
- Verborgene multiplier = 4 - Standaard en niet door de gebruiker configureerbaar
- Huidige Queue Ratio(klasse1) = 10% - Geconfigureerd in de beleidskaart met queue-buffers ratio 10
- Huidige Queue Ratio(klasse2) = 10% - Geconfigureerd in de beleidskaart met queue-buffers ratio 10
- Huidige Queue Ratio (klasse 3) = 10% - Geconfigureerd in de policy map met queue-buffers ratio 10
- Huidige Queue Ratio (klasse 4) = 10% - Geconfigureerd in de policy map met queue-buffers ratio 10
- Huidige Queue Ratio(class-default) = 12% - Geconfigureerd in de beleidskaart met queue-buffers ratio 12
- Totaal aantal wachtrijen = 5

Som van geconfigureerde verhoudingen = $10 + 10 + 10 + 10 + 12 = 52$

Resterende bufferverhouding = $100 \% - 52 \% = 48 \%$

[Resterende bufferverhouding] 48% / [Totaal aantal wachtrijen] 5 = 9,6% toegevoegd per wachtrij - Dit is geen geheel getal, dus de uiteindelijke toepassing op wachtrijen moet per wachtrij naar boven of naar beneden worden afgerond

Om het uiteindelijke queue-bufferverhoudingsgetal te krijgen dat het systeem gebruikt, moet u 9 of 10 toevoegen aan de reeds geconfigureerde queue-bufferverhouding.

Klassen hoger in de beleidskaart ontvangen de afgeronde waarde, 10. Klassen lager in de beleidskaart ontvangen de afgeronde waarde, 9.

Definitieve wachtrijbuffertoewijzing berekenen

Resterende bufferverhouding = 48

Klasse1 = [Basisbuffer] x ([Huidige wachtrijverhouding(klasse1) + Afgeronde waarde van de resterende gedeelde bufferverhouding])

Klasse 1 = $1200 \times (10\% + 10\%)/100 = 240$ - prioriteitwachtrij, geen verdere berekening

Resterende buffer = $(48 - 10) = 38$

Klasse 2 = [Basisbuffer] x ([Huidige wachtrijverhouding(klasse 2) + Afgeronde waarde van de resterende gedeelde bufferverhouding])

Klasse 2 = $1200 \times (10\% + 10\%)/100 = 240$ - Vermenigvuldig dit met gebruikers- en systeemvermenigvuldigers, aangezien dit geen prioriteitwachtrij is

Klasse2 = [Toewijzing basisbuffer voor deze wachtrij] $240 \times$ [Verborgen vermenigvuldiger] $4 \times$ [Vermenigvuldiger gebruiker] $(100/100) = 960$ - resultaat softmax voor deze wachtrij

Resterende buffer = $(38 - 10) = 28$

Herhaal dit voor wachtrijen die blijven staan:

Klasse 3 = [

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

$120 \times 4 \times (100/100) = 960$

]

Resterende bufferverhouding = $(28 - 10) = 18$

Klasse 4 = [

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

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

]

Resterende bufferverhouding = 9

Class-default= [

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

120 x 4 x (100/100) = 1008

]

Resterende bufferverhouding = 0

Resultaat van de toegepaste test5-beleidskaart ten opzichte van de voorspelling:

<#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 GlbSMin 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>
```

Scenario 5: Bepaal de zachte en harde buffer van een prioriteitsniveau 2-klasse

In dit scenario wordt een klasse in een beleidskaart geconfigureerd met prioriteitsniveau 2.

In tegenstelling tot prioriteitsniveau 1, waarin softmax niet wordt beïnvloed door multipliers en gelijk is gesteld aan hardmax, prioriteitsniveau 2 maakt het mogelijk om softmax te vermenigvuldigen terwijl het ook een harde buffer (hardmax) toewijzing heeft.

Policy-map in gebruik voor dit scenario:

```
<#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
```

Resultaat van de toegepaste test6-beleidskaart:

```
<#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 GlbSMin 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>
```

In de eerder getoonde uitvoer heeft de tweede wachtrij softmax 4 zoveel softmax als de eerste wachtrij. Dit komt omdat prioriteitsniveau 1 softmax specifiek niet wordt beïnvloed door systeemsoftmax-multipliers, maar prioriteitsniveau 2 wel.

Als u een gebruiker softmax multiplier configureert, wordt alleen de wachtrij op prioriteitsniveau 2 beïnvloed:

```
<#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>
```

wachtrijlimiet

De configuratie van de wachtrijlimiet is van invloed op de uiteindelijke toewijzing van de wachtrijbuffer

Het primaire mechanisme om de toewijzing van de wachtrijbuffer te beïnvloeden is de configuratie van de wachtrijbufferverhouding die per wachtrij wordt toegevoegd in een MQC-beleidskaart.

De toewijzing van de wachtrijbuffer wordt echter beïnvloed door andere configuraties.

Wachtrijlimiet definieert drempels waarmee u een bepaalde verkeersklasse laat vallen (via Weighted Tail Drop, WTD), die niet in dit document wordt behandeld.

In een specifieke omstandigheid wijzigt de wachtrijlimiet het systeem Hidden Multiplier voor soft buffer-wachtrijen - wat van invloed is op de totale soft buffer-toewijzing voor die wachtrij waarop de wachtrijlimiet wordt toegepast.

Begrijp eerst dat de wachtrijlimiet tot 3 keer per klasse kan worden geconfigureerd. Dit definieert maximaal 3 drempels voor WTD op een per DSCP, of CoS basis.

In de volgende output worden slechts twee drempels gedefinieerd.

Pas twee wachtrijlimieten toe op een beleidskaart:

```
<#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
```

Bekijk de resultaten van de buffertoewijzing:

```
<#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  G1b1SMin  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>
```

In het volgende voorbeeld wordt een derde wachtrijlimietconfiguratie toegevoegd aan class-default.

Bekijk de resultaten van de buffertoewijzing:

```

<#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   G1b1SMin   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>

```

Wanneer een derde wachtrijlimietconfiguratie wordt toegevoegd aan een wachtrij, wordt de verborgen soft buffer multiplier van het systeem van 400% uitgeschakeld voor die wachtrij. Die wachtrij respecteert echter nog steeds een door de gebruiker geconfigureerde queue-softmax-multiplier <percent>.

Gerelateerde informatie

- [Cisco Catalyst 9000 Switching Platforms: QoS en Queuing White Paper](#)
- [Cisco Technical Support en downloads](#)

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.