

# Ga op 12000 Series routers te werk om MDRR/WRED te configureren

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[MDRR - Overzicht](#)

[prioriteitswachtrij voor MDRR](#)

[MDRR-voorbeeld](#)

[MDRR-ondersteuning voor motortype](#)

[WRED-Overzicht](#)

[Verouderde CoS-Syntax gebruiken voor configuratie](#)

[Gebruik modulair QoS CLI \(MQC\) voor configuratie](#)

[Opdrachten voor het bewaken van congestiebeheer en -preventie](#)

[Opdracht beide interfaces](#)

[De tonen interfaces {aantal} willekeurige Opdracht](#)

[De \(y\) sleuf van de uitvoer tonen de van de controller van de rij {haven} Opdracht](#)

[De uitvoersleuf \(y\) toont controller van QM stat-opdracht](#)

[Monitor inkomende congestiebeheer](#)

[Opdracht beide interfaces](#)

[De extra sleuf \(x\) toont controller in de wachtrij](#)

[De extra sleuf \(x\) toont controller-tofab wachtrij \(sleuf\) \(poort\) Opdracht](#)

[De uitvoersleuf \(x\) toont controller-fab QM stat-opdracht](#)

[casestudy](#)

[Gerelateerde informatie](#)

## [Inleiding](#)

Dit document onderzoekt hoe u Cisco IOS®-software voor congestiebeheer en congestievermijding kunt configureren op de Cisco 12000 Series Internet-router.

Nadat u dit document hebt gelezen, kunt u:

- Begrijp waarom het belangrijk is om Gewijzigd tekort round robin (MDRR) en Weighted Random Early Detection (WRED) in uw kernnetwerk te configureren.
- Begrijp de implementatie die de basis vormt van MDRR en WRED op Cisco 12000 Series.

- Configureer MDRR en WRED met de syntaxis van de legacy-klasse (CoS) en de modulaire QoS CLI (MQC).

## Voorwaarden

### Vereisten

Lezers van dit document zouden kennis moeten hebben van deze onderwerpen:

- Een algemene vertrouwdheid met de Cisco 12000 Series Internet Router architectuur.
- In het bijzonder een bewustzijn van de wachtende architectuur en deze termen: ToFab (ToFabric) - dat de ontvangstzijwachtrijen op een inkomende lijnkaart beschrijft. FrFab (From the Fabric) - die de verzendzijde wachtrijen op een uitgaande lijnkaart beschrijft.

**Opmerking:** Ook wordt aangeraden om [Hoe u de uitvoer van het controllerformaat wilt lezen | Opdrachten in een tofab-wachtrij op Cisco 12000 Series internetrouter](#).

### Gebruikte componenten

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

- Alle Cisco 12000-platforms, die de 12008, 12012, 12016, 12404, 12406, 12410 en 12416 omvatten.
- Cisco IOS-software release 12.0(24)S1.

**Opmerking:** Hoewel de configuraties in dit document zijn getest op Cisco IOS-software release 12.0(24)S1, kan elke Cisco IOS-software release die Cisco 12000 Series Internet-router ondersteunt, worden gebruikt.

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 de potentiële impact van elke opdracht begrijpen.

### Conventies

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

## Achtergrondinformatie

Wachtrijen-methoden definiëren het pakketplanningsmechanisme of de volgorde waarin pakketten worden gewijd aan de interface voor transmissie op de fysieke draad. Op basis van de volgorde en het aantal keren dat een wachtrij wordt onderhouden door een plannerfunctie ondersteunen wachtmethoden ook minimale bandbreedtegaranties en lage latenties.

Het is belangrijk om te verzekeren dat een pakket planningsmechanisme de schakelarchitectuur ondersteunt waarop het wordt geïmplementeerd. Weighted Fair Quinging (WFQ) is het bekende planningsalgoritme voor toewijzing van middelen op Cisco-routerplatforms met een op een bus gebaseerde architectuur. Het wordt echter niet ondersteund op Cisco 12000 Series Internet

Router. Traditionele Cisco IOS-software-releases en aangepaste wachtrij worden ook niet ondersteund. In plaats daarvan gebruikt Cisco 12000 Series een speciale vorm van het in de wachtrij stellen van het Gewijzigd tekort round robin (MDRR), dat relatieve bandbreedtegaranties evenals een lage latentiewachtrij biedt. De M van MDRR staat voor "aangepast"; de prioriteitswachtrij wordt toegevoegd in vergelijking met de DRR, waar geen prioritaire wachtrij aanwezig is. Zie voor meer informatie over MDRR het gedeelte [Overzicht MDRR](#).

Daarnaast ondersteunt Cisco 12000 Series Weighted Random Early Detection (WRED) als valbeleid binnen de MDRR-wachtrijen. Dit mechanisme ter voorkoming van congestie biedt een alternatief voor het standaard-staart-valmechanisme. De congestie kan worden vermeden door middel van gecontroleerde druppels.

Vermijding van congestie en beheermechanismen zoals WRED en MDRR zijn met name van belang op de FrFab-wachtrijen van relatief snelle uitgaande interfaces, zoals gekanaliseerde lijnkaarten (LC's). De hoge-snelheids switch stof kan pakketten aan de kanaalgroepen leveren veel sneller dan de kanaalgroepen ze kunnen verzenden. Aangezien de wachtrijen / buffers op het fysieke havenniveau worden beheerd, kan de backpressie op één kanaal alle andere kanalen op die haven beïnvloeden. Daarom is het van groot belang dat deze tegendruk wordt beheerd via WRED/MDRR, die beperkt tot de impact van de rugdruk op het (de) betrokken kanaal(en). Voor informatie over hoe u uitgaande interface-overabonnement te beheren, zie [Problemen oplossen genegeerd Packets en Geen geheugen vallen op Cisco 12000 Series Internet-router](#).

## MDRR - Overzicht

Deze paragraaf geeft een overzicht van het wijzigd tekort van het round robin (MDRR).

Als de MDRR is ingesteld als een wachtrij, worden lege rijen bij de ene na de andere in een ring verzorgd. Elke keer dat een wachtrij wordt bediend, wordt een vaste hoeveelheid gegevens in de wachtrij geplaatst. De algoritme bedient dan de volgende rij. Wanneer een wachtrij wordt geopend, houdt MDRR bij hoeveel bytes van gegevens meer dan de geconfigureerde waarde zijn teruggegeven. In de volgende pas, wanneer de rij opnieuw wordt gediend, zullen minder gegevens worden gedewachtrij geplaatst om de overtollige gegevens te compenseren die eerder werden gediend. Als resultaat hiervan zal de gemiddelde hoeveelheid gegevens die per wachtrij wordt geplaatst, dicht bij de geconfigureerde waarde liggen. Daarnaast houdt de MDRR een prioriteitswachtrij die op preferentiële basis wordt gediend. MDRR wordt in dit gedeelte uitvoeriger uitgelegd.

Elke rij binnen MDRR wordt gedefinieerd door twee variabelen:

- Quantum waarde - Dit is het gemiddelde aantal bytes dat in elke ronde wordt bediend.
- Tekstteller - Dit wordt gebruikt om te volgen hoeveel bytes een wachtrij in elke ronde heeft verzonden. Het wordt geïnitieerd tot de kwantumwaarde.

Pakketten in een rij worden bediend zolang de tekenteller groter is dan nul. Elk pakket dat wordt gediend, verlaagt de tekenteller met een waarde gelijk aan de lengte in bytes. Een wachtrij kan niet langer worden bediend nadat de tekenteller nul of negatief is geworden. In elke nieuwe ronde wordt de tekenteller van elke niet-lege wachtrij verhoogd met de kwantumwaarde ervan.

**Opmerking:** In het algemeen mag de kwantumgrootte van een wachtrij niet kleiner zijn dan de maximale transmissie-eenheid (MTU) van de interface. Dit waarborgt dat de server altijd ten minste één pakje uit elke niet lege wachtrij dient.

Elke MDRR-rij kan een relatief gewicht krijgen, waarbij één van de wachtrijen in de groep als een prioriteitswachtrij wordt gedefinieerd. De gewichten wijzen relatieve bandbreedte voor elke rij toe wanneer de interface wordt geblokkeerd. Het MDRR-algoritme stelt gegevens van elke wachtrij in een ronde-robin-mode op als er gegevens in de wachtrij staan die moeten worden verstuurd.

Als alle reguliere MDRR-wachtrijen gegevens bevatten, worden deze als volgt beheerd:

0-1-2-3-4-5-6-0-1-2-3-4-5-6...

Tijdens elk programma kan een wachtrij een kwantum definiëren op basis van het ingestelde gewicht. Op de lijnkaarten van motor 0 en motor 2 is een waarde van 1 gelijk aan het geven van een gewicht van de interface. Bij elke toename boven 1 wordt het gewicht van de wachtrij met 512 bytes verhoogd. Als de MTU van een bepaalde interface bijvoorbeeld 4470 is en het gewicht van een wachtrij 3 is, mag elke keer door de draaiing  $4470 + (3-1)*512 = 5494$  bytes worden afgesteld. Als twee normale DRR-wachtrijen, Q0 en Q1, worden gebruikt, wordt Q0 ingesteld met een gewicht van 1 en Q1 met een gewicht van 9. Als beide wachtrijen worden geblokkeerd, wordt Q0 elke keer door de rotatie 4470 bytes verzenden en Q1  $[4470 + (9-1)*512] = 8566$  bytes. Dit zou verkeer geven dat naar Q0 ongeveer 1/3 van de bandbreedte gaat, en het verkeer dat door Q1 gaat ongeveer 2/3 van de bandbreedte.

**Opmerking:** De standaard de-wachtrij formule die wordt gebruikt om de MDRR-bandbreedtetoeewijzing te berekenen is  $D = MTU + (\text{gewicht}-1)*512$ . In versies eerder dan Cisco IOS-software release 12.0(21)S/ST, gebruikte Engine 4-lijnkaarten een andere dewachtrij-formule. Om het gewicht MDRR te configureren voor een correcte bandbreedte-toewijzing, zorg er dan voor dat u een Cisco IOS-software release later dan 12.0(21)S/ST draait.

**Opmerking:** De kwantumformule voor de Engine 4+ lijnkaarten is (voor de toFab richting is dit niet geldig voor FrFab)  $\text{Quantum} = \text{BaseWeight} + \{\text{BaseWeight} * (\text{QueueWeight} - 1) * 512\} / \text{MTU}$ . Het basisgewicht wordt verkregen met deze formule:  $\text{BaseWeight} = \{(MTU + 512 - 1) / 512\} + 5$

**Opmerking:** Alle berekeningen worden afgerond; alle decimalen worden dus genegeerd .

**Opmerking:** Om te weten of een specifieke motorlijnkaart MDRR ondersteunt, zie [MDRR-ondersteuning door motortype](#).

## [prioriteitswachtrij voor MDRR](#)

Cisco 12000 Series ondersteunt een prioriteitswachtrij (PQ) binnen MDRR. Deze rij biedt de lage vertraging en de lage vertraging die door tijdgevoelig verkeer zoals Voice over IP (VoIP) vereist worden.

Zoals hierboven vermeld, steunt Cisco 12000 Series geen gewogen eerlijke wachtrij (WFQ). Dus werkt het PQ in MDRR anders dan de Cisco IOS software Low Latency Queing (LLQ) die voor andere platforms beschikbaar is.

Een belangrijk verschil is hoe de MDRR-planner kan worden ingesteld om de PQ in een van de twee modi te bedienen, zoals aangegeven in [tabel 1](#):

**Tabel 1 - Hoe de MDRR Scheduler te configureren om de PQ in twee modi te onderhouden**

	Alternatieve modus	Strikte prioriteitsmodus
Voor delen	Hier wordt het PQ onderhouden tussen de	Hier wordt op het PQ onderhoud uitgevoerd

	andere wachtrijen. Met andere woorden: de MDRR-server biedt alternatief diensten aan de PQ en andere geconfigureerde wachtrijen.	wanneer het niet leeg is. Dit biedt de laagst mogelijke vertraging voor het afstemmen van verkeer.
nadel en	Deze modus kan scherp en vertraagd introduceren in vergelijking met de strikte prioriteitsmodus.	Deze modus kan andere wachtrijen veroorzaken, vooral als de matchingsstromen agressieve zenders zijn.

In een andere modus kan minder controle over de startvertraging worden uitgeoefend. Als de MDRR planner begint met het bedienen van MTU-vormige frames uit een gegevenswachtrij en dan arriveert een spraakpakket in de PQ, dient de server in alternatieve modus volledig de wachtrij met geen prioriteit totdat de tekortteller nul bereikt heeft. Gedurende deze tijd wordt er geen onderhoud uitgevoerd aan het PQ en worden de VoIP-pakketten uitgesteld.

In de prioriteitsmodus daarentegen kunnen de plannerservices alleen het huidige niet-prioriteitspakket uitvoeren en vervolgens switches naar het PQ. De planner begint pas een rij zonder prioriteit te bedienen nadat de PQ volledig leeg is.

Het is belangrijk om op te merken dat de prioriteitswachtrij in alternatieve prioriteitsmodus meer dan eens in een cyclus wordt onderhouden en dus meer bandbreedte vergt dan andere wachtrijen met hetzelfde nominale gewicht. Hoeveel meer is een functie van hoeveel wachtrijen worden gedefinieerd. Met drie wachtrijen wordt de rij voor lage latentie bijvoorbeeld twee keer zo vaak onderhouden als de andere wachtrijen en twee keer zo zwaar per cyclus. Als acht wachtrijen worden gedefinieerd, wordt de rij met lage latentie zeven keer vaker onderhouden en is het effectieve gewicht zeven keer hoger. Dus de bandbreedte die de wachtrij kan nemen is gerelateerd aan hoe vaak deze per ronde robin wordt gediend, wat op zijn beurt afhangt van hoeveel wachtrijen over het geheel genomen worden gedefinieerd. In alternatieve prioriteitsmodus wordt de prioriteitswachtrij om deze specifieke reden doorgaans met een klein gewicht ingesteld.

Ga er als voorbeeld van uit dat vier wachtrijen zijn gedefinieerd: 0, 1, 2 en de prioriteitswachtrij. In alternatieve prioriteitsmodus, als alle wachtrijen geblokkeerd zijn, worden ze als volgt beheerd: **0, llq, 1, llq, 2, llq, 0, llq, 1, ....** waar llq staat voor een lage latentierij.

Elke keer dat er een wachtrij wordt onderhouden, kan deze naar het geconfigureerde gewicht doorsturen. Daarom is de minimale bandbreedte die de lage latency wachtrij kan hebben:

- WL = gewicht van de rij met lage latencie.
- W0, W1, ... Wn = gewichten van de reguliere DRR-wachtrijen.
- n = aantal regelmatige DRR-rijen voor deze interface.
- BW = bandbreedte van de link.

Voor alternatieve prioriteitsmodus, de minimale bandbreedte van de lage latentiewachtrij =  $BW * n * WL / (n * WL + \text{som}(W0, Wn))$ .

Het gewicht is de enige variabele parameter in MDRR die kan worden ingesteld. Het beïnvloedt de relatieve hoeveelheid bandbreedte die een verkeersklasse kan gebruiken, en hoeveel verkeer in één bocht wordt verzonden. Het gebruik van grotere gewichten betekent dat de gehele cyclus langer duurt en mogelijk de latencie verhoogt.

## Configuratierichtsnoeren

- Het is beter om het gewicht van de klasse te vormen die de laagste bandbreedte vereiste aan 1 heeft om de vertraging en de jitter zo laag mogelijk onder de andere klassen te houden.
- Selecteer gewichtswaarden die zo laag mogelijk zijn. Begin met een gewicht van 1 voor de klasse met de laagste bandbreedte. Bijvoorbeeld, wanneer u twee klassen met 50% bandbreedte voor elke klasse gebruikt, moet u 1 en 1 configureren. Het slaat niet op om 10 en 10 te gebruiken, omdat er geen impact is op de prestaties wanneer u 1 kiest. Ook introduceert een hoger gewicht meer jitter.
- Een lage gewichtswaarde voor de LLQ is zeer belangrijk, vooral in alternatieve modus om niet te veel vertraging of jitter aan de andere klassen toe te voegen.

## MDRR-voorbeeld

Het voorbeeld in deze sectie wordt genomen van *Inside Cisco IOS® Software Architecture*, Cisco Press.

Stel dat we drie wachtrijen hebben:

- **Wachtrij 0** - heeft een hoeveelheid van 1500 bytes; het is de lage latency wachtrij, ingesteld om in alternatieve modus te werken.
- **Wachtrij 1** - heeft een hoeveelheid van 3000 bytes.
- **Wachtrij 2** - heeft een hoeveelheid van 1500 bytes.

[Afbeelding 1](#) illustreert de aanvankelijke status van de wachtrijen, samen met enkele pakketten die werden ontvangen en in de wachtrij werden geplaatst.

### Afbeelding 1 - MDRR Eerste Staat

	Queues					Deficit Counters
Queue 0			3/250	2/1500	1/250	0
Queue 1			6/1500	5/1500	4/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

Wachtrij 0 wordt eerst onderhouden, de kwantum wordt toegevoegd aan de tekenteller, Packet 1, dat 250 bytes is, wordt doorgegeven en de grootte ervan wordt afgetrokken van de tekenteller. Omdat de tekenteller van rij 0 nog steeds groter is dan 0 ( $1500 - 250 = 1250$ ) wordt pakket 2 ook verzonden, en de lengte ervan wordt afgetrokken van de tekenteller. De tekort-teller van wachtrij 0 is nu -250, dus wordt rij 1 als volgt beheerd. [Afbeelding 2](#) geeft deze status aan.

### Afbeelding 2 - MDRR - vervolgstadium

Queues					Deficit Counters			
Queue 0					3/250	-250		
Queue 1					6/1500	5/1500	4/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0		

De tekortbalie van rij 1 is ingesteld op 3000 ( $0 + 3000 = 3000$ ) en de pakketten 4 en 5 worden doorgegeven. Met elk verzonden pakket, trek de grootte van het pakket af van de tekenteller, zodat de tekenteller van rij 1 wordt verminderd tot 0. [Afbeelding 3](#) illustreert deze staat.

**Afbeelding 3 - MDRR-staat wanneer de tekortteller van wachtrij 1 op nul staat**

Queues					Deficit Counters	
Queue 0					3/250	-250
Queue 1					6/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

We moeten terugkeren van alternatieve prioriteitsmodus naar dienstenrij 0. Opnieuw wordt het kwantum toegevoegd aan de huidige tekenteller en wordt de tekortteller van rij 0 ingesteld op het resultaat ( $-250 + 1500 = 1250$ ). Packet 3 wordt nu doorgegeven, omdat de tekort-teller groter is dan 0, en rij 0 nu leeg is. Wanneer een wachtrij wordt geleegd, wordt de tekortteller op 0 ingesteld, zoals in [afbeelding 4](#).

**Afbeelding 4 - MDRR-status wanneer een wachtrij wordt geopend**

Queues					Deficit Counters	
Queue 0						0
Queue 1					6/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

Wachtrij 2 wordt hierna onderhouden; de tekortquote bedraagt 1500 ( $+ 1500 = 1500$ ). Pakketten 7 tot en met 10 worden doorgegeven, waardoor de tekortteller op 500 ( $1500 - (4 \cdot 250) = 500$ ) wordt achtergelaten. Omdat de tekenteller nog groter is dan 0, wordt pakje 11 ook verzonden.

Wanneer pakket 11 wordt verzonden, is wachtrij 2 leeg en is de tekortteller op 0 ingesteld, zoals in [afbeelding 5](#).

**Afbeelding 5 - MDRR-status wanneer Packet 11 wordt verzonden**



Queues	Deficit Counters
Queue 0	0
Queue 1	0
Queue 2	0

Wachtrij 0 wordt daarna opnieuw onderhouden (omdat we in alternatieve prioriteitsmodus zijn). Omdat het leeg is, dienen we wachtrij 1 volgende in en verzenden we pakket 6.

### MDRR-ondersteuning voor motortype

Cisco 12000 Series ondersteunt vijf lijnkaartmodellen met unieke Layer 3 (L3) verzendende Engine architecturen. De ondersteuning van MDRR varieert per L3-motor en per kaarttype. Bijvoorbeeld, er is geen steun voor MDRR en WRED op Engine 0 ATM lijnkaarten. U kunt de opdracht **Show** diag gebruiken om het L3 Engine type van uw geïnstalleerde lijnkaarten te bepalen:

```
router#show diags | include (SLOT | Engine)
!--- The regular expression is case-sensitive. ... SLOT 1 (RP/LC 1 ): 1 port ATM Over SONET
OC12c/STM-4c Multi Mode L3 Engine: 0 - OC12 (622 Mbps) SLOT 3 (RP/LC 3 ): 3 Port Gigabit
Ethernet L3 Engine: 2 - Backbone OC48 (2.5 Gbps)
```

### MDRR op ToFab (RX) wachtrijen

U kunt de "Legacy CoS Syntax" of de "modulaire QoS opdrachtregel" gebruiken om de MDRR op Cisco 12000 Series te configureren. De latere delen in dit document bespreken hoe u MDRR kunt configureren met behulp van bestaande QoS of modulair QoS. U dient de ToFab wachtrijen met de legacy CoS-syntaxis alleen te configureren omdat deze de nieuwere syntaxis van MQC niet ondersteunt. Zie [tabel 2](#) voor meer informatie.

**Tabel 2 - Details over ToFab (RX)-wachtrijen**

	Geïmplementeerd	ToFab MDRR	ToFab Alternatieve PQ	ToFab Streng PQ	ToFab WRED
Eng.0	in Cisco IOS®-software	Nee**	Nee**	Ja	Ja
1 inch	-	Nee	Nee	Nee	Nee
Eng.2	Hardware	Ja	Ja	Ja	Ja
3 ENG	Hardware	Nee	Ja	Ja	Ja
4 eng.	Hardware	Ja	Ja	Ja	Ja
+	Hardware	Ja	Ja	Ja	Ja



ENG4+	e				
-------	---	--	--	--	--

\*\* MDRR wordt ondersteund op Engine 0 LCs in de ToFab (Rx) richting, maar alleen in de prioriteitsmodus, en niet in de alternatieve prioriteitsmodus. De zeven resterende rijen worden zoals gewoonlijk ondersteund.

Inkomende interfaces houden een afzonderlijke virtuele uitvoerwachtrij per bestemming LC. Hoe deze wachtrijen worden geïmplementeerd, is afhankelijk van het L3-machinetype.

- Engine 0 - inkomende LC's onderhouden acht wachtrijen per doelsleuf. Zo is het maximale aantal wachtrijen  $16 \times 8 = 128$ . Elke wachtrij kan afzonderlijk worden ingesteld.
- Motoren 2, 3, 4, en 4+ - Ingebonden LC's onderhouden acht rijen per doelinterface. Met 16 doelsleuven en 16 interfaces per sleuf is het maximale aantal wachtrijen  $16 \times 16 \times 8 = 2048$ . Alle interfaces op een doelsleuf moeten dezelfde parameters gebruiken.

### [MDRR op FrFab \(TX\)-wachtrijen](#)

MDRR op de FrFab wachtrijen werkt consistent, ongeacht of het in hardware of software is geïmplementeerd. Alle L3 motortypen ondersteunen acht klassen wachtrijen voor elke uitgaande interface. Zie [tabel 3](#) voor meer informatie.

**Tabel 3 - Details over MDRR op FrFab (TX)-wachtrijen**

	Geïmplementeerd	FrFab Alternate PQ	Voor stevig PQ	FrFab WRED
Eng.0	Software <sup>1</sup>	Nee	Ja	Ja
1 inch	-	Nee	Nee	Nee
Eng.2	Hardware	Ja <sup>2</sup>	Ja	Ja
3 ENG	Hardware	Nee	Ja	Ja
4 eng.	Hardware	Ja	Ja	Ja
+ ENG4+	Hardware	Ja	Ja	Ja

<sup>1</sup>Ondersteuning voor MDRR op FrFab wachtrijen van Engine 0 LCs wordt in deze versies van Cisco IOS-software geïntroduceerd:

- Cisco IOS-software release 12.0(10)S - 4xOC3 en 1xOC-12 POS, 4xOC-3 en 1xCHOC-12/STM-4.
- Cisco IOS-software release 12.0(15)S - 6xE3 en 12xE3.
- Cisco IOS-software release 12.0(17)S - 2xCHOC3/STM1.

<sup>2</sup> U dient alternatieve MDRR te configureren in de FrFab richting met de legacy CoS syntax.

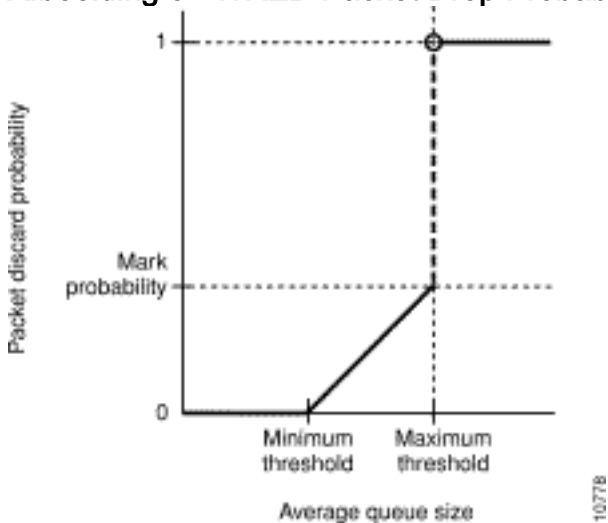
**Opmerking:** 3xGE LC ondersteunt MDRR op de ToFab-wachtrijen en, vanaf Cisco IOS-software release 12.0(15)S, op de FrFab-wachtrijen met twee beperkingen, namelijk een vaste kwantum en een enkele CoS-wachtrij voor elke interface. De prioriteitswachtrij ondersteunt een kwantum dat kan worden geconfigureerd, en zowel de strikte als de alternatieve prioriteitsmodus. Alle drie interfaces hebben één hoofdkwartier.

**Opmerking:** Zie de opmerkingen van Cisco 12000 Series routers release voor de laatste informatie over ondersteunde QoS-functies op Cisco 12000 Series LC's.

## WRED-Overzicht

Weighted Random Early Detection (WRED) is ontworpen om de schadelijke effecten van interfacecongestie op de doorvoersnelheid van het netwerk te voorkomen.

Afbeelding 6 - WRED Packet Drop Probability



Zie [Gewogen Willekeurige vroege detectie op Cisco 12000 Series router](#) voor een verklaring van de WRED parameters. De minimum-, maximum- en mark-waarschijnlijkheidsparameters beschrijven de eigenlijke Random Early Detection (RED)-curve. Wanneer het gewogen rijgemiddelde onder de minimum drempel ligt, worden er geen pakketten verzonden. Wanneer het gewogen rijgemiddelde boven de maximum drempel is, worden alle pakketten gedropt tot de gemiddelde drempel onder de maximum. Wanneer het gemiddelde tussen de minimum- en de maximumdrempels ligt, kan de waarschijnlijkheid dat de pakje wordt afgevoerd, worden berekend door een rechte lijn van de minimumdrempel (de kans op een daling is 0) tot de maximumdrempel (de kans op een daling is gelijk aan de waarschijnlijkheidsnoemer van 1/1).

Het verschil tussen RED en WRED is dat WRED selectief verkeer met lagere prioriteit kan weggooien wanneer de interface wordt geblokkeerd, en gedifferentieerde prestatiekenmerken kan bieden voor verschillende serviceklasse (CoS). WRED gebruikt standaard een ander ROOD profiel voor elk gewicht (IP-voorrang - 8 profielen). Het druppelt minder belangrijke pakketten agressiever dan belangrijker pakketten.

Het is een complexe uitdaging om WRED-parameters af te stemmen om de wachtrijdiepte te beheren, en is afhankelijk van veel factoren, waaronder:

- Aangeboden verkeersbelasting en -profiel.
- Verhouding van de lading tot de beschikbare capaciteit.
- Gedrag van het verkeer in aanwezigheid van congestie.

Deze factoren verschillen per netwerk en zijn op hun beurt afhankelijk van de aangeboden diensten en de klanten die deze diensten gebruiken. We kunnen dus geen aanbevelingen doen die van toepassing zijn op specifieke klantenomgevingen. [Tabel 4](#) beschrijft echter algemeen aanbevolen waarden op basis van de bandbreedte van de link. In dat geval maken wij geen onderscheid tussen de verschillende categorieën diensten.

Tabel 4 - Aanbevolen waarden gebaseerd op de bandbreedte van de link

Bandbreedte	Theoretisch BW	Fysieke BW <sup>1</sup>	Minimumdrempel	Maximumdrempel
-------------	----------------	-------------------------	----------------	----------------

	(kbps)	(kbps)		
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

## <sup>1</sup> Fysieke SONET snelheid

Er zijn verschillende beperkingen waarmee rekening wordt gehouden bij de berekening van de bovengenoemde drempelwaarden. Bijvoorbeeld, de maximalisering van het gebruik van de link terwijl de gemiddelde wachtrijdiepte wordt geminimaliseerd, moet het verschil tussen het maximum en het minimum een vermogen van twee zijn (door hardwarebeperking). Gebaseerd op ervaring en simulatie is de maximale momentane diepte van een rij gecontroleerd door RED minder dan 2 MaxTh. Voor OC48 en hoger, 1 MaxTh, enzovoort. De exacte vaststelling van deze waarden valt echter buiten het toepassingsgebied van dit document.

**Toelichting:** De exponentiële wegingsconstante waarde hoeft niet te worden ingesteld op motoren 2 en boven lijnkaarten, aangezien in plaats daarvan een steekproef van de hardware-wachtrij wordt gebruikt. Voor Engine 0 LCs kunnen deze waarden worden ingesteld:

- DS3 - 9
- 10 - 10
- 12 - 12

**Opmerking:** WRED wordt niet ondersteund op Engine 1 LCs.

Zoals de volgende secties verklaren, kunt u zowel de legacy CoS syntaxis als de nieuwere MQC syntaxis gebruiken om WRED te configureren.

## [Verouderde CoS-Syntax gebruiken voor configuratie](#)

De syntaxis van Cisco 12000 Series legacy Class of Service (CoS) gebruikt een sjabloon **van cos in wachtrij** om een reeks CoS-definities te definiëren. U past de sjabloon vervolgens op ToFab en FrFab wachtrijen op inkomende of uitgaande interfaces toe.

### [Stap 1: Een cos-wachtrij-groep definiëren](#)

De opdracht **cos-wachtrij-groep** maakt een genaamd sjabloon van MDRR- en WRED-parameters. Hier zijn de beschikbare configuratieparameters in CLI:

```
Router(config)#cos-queue-group oc12
Router(config-cos-que)#?
Static cos queue commands:
```

default	Set a command to its defaults
dscp	Set per DSCP parameters, Engine 3 only
exit	Exit from COS queue group configuration mode
exponential-weighting-constant	Set group's RED exponential weight constant. (Not used by engine 0, 1 or 2 line cards)
no	Negate a command or set its defaults
precedence	Set per precedence parameters
queue	Set individual queue parameters

random-detect-label  
traffic-shape

Set RED drop criteria  
Enable Traffic Shaping on a COS queue group

Met MDRR kunt u de IP voorrang in kaart brengen aan de rijen van MDRR en de speciale rij van de lage latentie configureren. U kunt de prioriteitsparameter onder de opdracht **cos-wachtrij-groep** gebruiken voor deze opdracht:

```
precedence <0-7> queue [ <0-6> | low-latency]
```

U kunt een bepaalde IP-voorrang aan een regelmatige MDRR-wachtrij (wachtrij 0 tot 6) in kaart brengen of u kunt deze in de prioriteitswachtrij plaatsen. Met deze opdracht kunt u meerdere IP-voorrang in dezelfde wachtrij toewijzen.

**Opmerking:** Aanbevolen wordt om voorrang 5 te gebruiken voor de rij met lage latentie. Voorschrift 6 wordt gebruikt voor het routeren van updates.

U kunt elke MDRR-rij een relatief gewicht geven, waarbij een van de wachtrijen in de groep wordt gedefinieerd als een prioriteitswachtrij. U kunt de wachtrijopdracht gebruiken onder de **cos-wachtrij-groep** om dit te doen:

```
queue <0-6> <1-2048>  
queue low-latency [alternate-priority | strict-priority] <1-2048>  
!--- The weight option is not available with strict priority.
```

Gebruik de opdracht **cos-wachtrij-groep** om WRED-parameters te definiëren:

```
random-detect-label
```

Hier is een voorbeeld van een **cos-wachtrij-groep** genaamd **Oc12**. Het definieert drie verkeersklassen: rij 0, 1 en lage latentie. Het brengt IP prioriteitswaarden 0 - 3 in kaart om in rij 0, prioriteitswaarden 4, 6 en 7 in rij 1, en prioriteitswaarde 5 in de rij met lage latentie. Wachtrij 1 wordt meer bandbreedte toegewezen.

### Configuratievoorbeeld

```
cos-queue-group oc12  
!--- Creation of cos-queue-group called "oc12".  
  
precedence 0 queue 0  
!--- Map precedence 0 to queue 0. precedence 0 random-  
detect-label 0 !--- Use RED profile 0 on queue 0.  
precedence 1 queue 0 precedence 1 random-detect-label 0  
precedence 2 queue 0 precedence 2 random-detect-label 0  
precedence 3 queue 0 precedence 3 random-detect-label 0  
!--- Precedence 1, 2 and 3 also go into queue 0.  
precedence 4 queue 1 precedence 4 random-detect-label 1  
precedence 6 queue 1 precedence 6 random-detect-label 1  
precedence 7 queue 1 precedence 7 random-detect-label 1  
precedence 5 queue low-latency !--- Map precedence 5 to  
special low latency queue. !--- We do not intend to drop
```

```

any traffic from the LLQ. We have an SLA !--- that
commits not to drop on this queue. You want to give it
all !--- the bandwidth it requires. Random-detect-label
0 375 2423 1 !--- Minimum threshold 375 packets, maximum
threshold 2423 packets. !--- Drop probability at maximum
threshold is 1. random-detect-label 1 375 2423 1 queue 1
20 !--- Queue 1 gets MDRR weight of 20, thus gets more
Bandwidth. queue low-latency strict-priority !--- Low
latency queue runs in strict priority mode.

```

## Stap 2 - Maak een sleuf-tabel-code voor ToFab-wachtrijen

Om het hoofd van lijn blokkeren te voorkomen, houden de inkomende interfaces op Cisco 12000 Series een virtuele uitvoerwachtrij per doelsleuf. Ga naar een lijnkaart met de aanhechtingsopdracht en voer het aanvinkbevel van de showcontroller uit om in de wachtrij te staan (of voer direct de aanloopsleuf 0 uit om controllers in de wachtrij te tonen) om deze virtuele uitvoerwachtrijen te bekijken. Hieronder vindt u voorbeelden van uitvoer die rechtstreeks vanuit de LC-console is opgenomen. Zie [Hoe u de uitvoer van het controllerformaat van de show wilt lezen | Opdrachten in een tofab-wachtrij op Cisco 12000 Series internetrouter.](#)

LC-Slot1#**show controllers tofab queues**

Carve information for ToFab buffers

SDRAM size: 33554432 bytes, address: 30000000, carve base: 30029100

33386240 bytes carve size, 4 SDRAM bank(s), 8192 bytes SDRAM pagesize, 2 carve(s)

max buffer data size 9248 bytes, min buffer data size 80 bytes

40606/40606 buffers specified/carved

33249088/33249088 bytes sum buffer sizes specified/carved

Qnum	Head	Tail	#Qelem	LenThresh
----	----	----	-----	-----

5 non-IPC free queues:

20254/20254	(buffers specified/carved), 49.87%, 80 byte data size			
1	17297	17296	20254	65535
12152/12152	(buffers specified/carved), 29.92%, 608 byte data size			
2	20548	20547	12152	65535
6076/6076	(buffers specified/carved), 14.96%, 1568 byte data size			
3	32507	38582	6076	65535
1215/1215	(buffers specified/carved), 2.99%, 4544 byte data size			
4	38583	39797	1215	65535
809/809	(buffers specified/carved), 1.99%, 9248 byte data size			
5	39798	40606	809	65535

IPC Queue:

100/100	(buffers specified/carved), 0.24%, 4112 byte data size			
30	72	71	100	65535

Raw Queue:

31	0	17302	0	65535
----	---	-------	---	-------

**ToFab Queues:**

Dest				
Slot				
0	0	0	0	65535
1	0	0	0	65535
2	0	0	0	65535
3	0	0	0	65535
4	0	0	0	65535
5	0	17282	0	65535
6	0	0	0	65535
7	0	75	0	65535
8	0	0	0	65535
9	0	0	0	65535

10	0	0	0	65535
11	0	0	0	65535
12	0	0	0	65535
13	0	0	0	65535
14	0	0	0	65535
15	0	0	0	65535
<b>Multicast</b>	0	0	0	65535

LC-Slot1#

Gebruik de opdracht **sleuf-tabel-cos** om een genoemde **groep cos-in-wachtrij**-voor een doelvirtuele uitvoerwachtrij in kaart te brengen. U kunt een unieke **cos-wachtrij-groepssjabloon** voor elke uitvoerwachtrij configureren

```
Router(config)#slot-table-cos table1
Router(config-slot-cos)#destination-slot ?
<0-15> Destination slot number
all Configure for all destination slots
Router(config-slot-cos)#destination-slot 0 oc48
Router(config-slot-cos)#destination-slot 1 oc48
Router(config-slot-cos)#destination-slot 2 oc48
Router(config-slot-cos)#destination-slot 3 oc48
Router(config-slot-cos)#destination-slot 4 oc12
Router(config-slot-cos)#destination-slot 5 oc48
Router(config-slot-cos)#destination-slot 6 oc48
Router(config-slot-cos)#destination-slot 9 oc3
Router(config-slot-cos)#destination-slot 15 oc48
```

**Opmerking:** De bovenstaande configuratie gebruikt drie sjablonen, oc48, oc12 en oc3 genoemd. De configuratie voor de **cos-wachtrij-groep** genaamd oc12 is zoals in Stap 1 weergegeven. Stel ook oc3 en oc48 in. Aanbevolen wordt om een unieke sjabloon toe te passen op een reeks interfaces gebaseerd op de bandbreedte en toepassing.

### [Stap 3 - Pas een sleuf-tabel-code op een inkomende interface toe](#)

Gebruik de opdracht **rx-cos-sleuf** om een **sleuf-tabel-code** op een LC toe te passen.

```
Router(config)#rx-cos-slot 0 ?
WORD Name of slot-table-cos
Router(config)#rx-cos-slot 0 table1
Router(config)#rx-cos-slot 2 table1
```

### [Stap 4 - Pas een groep van de cos-wachtrij op een uitgaande interface toe](#)

Cisco 12000 Series onderhoudt een aparte rij per uitgaande interface. Om deze wachtrijen te bekijken, sluit u deze aan op de lijnkaart CLI. Gebruik de **attach**-opdracht en voer vervolgens de opdracht **voor controller uit-feldwachtrij uit**, zoals hier wordt weergegeven:

```
LC-Slot1#show controller frfab queue
===== Line Card (Slot 2) =====
Carve information for FrFab buffers
SDRAM size: 16777216 bytes, address: 20000000, carve base: 2002D100
16592640 bytes carve size, 0 SDRAM bank(s), 0 bytes SDRAM pagesize, 2 carve(s)
max buffer data size 9248 bytes, min buffer data size 80 bytes
20052/20052 buffers specified/carved
```

```

16581552/16581552 bytes sum buffer sizes specified/carved
  Qnum      Head      Tail      #Qelem  LenThresh
  ----      -
5 non-IPC free queues:
  9977/9977 (buffers specified/carved), 49.75%, 80 byte data size
  1         101       10077     9977    65535
  5986/5986 (buffers specified/carved), 29.85%, 608 byte data size
  2         10078     16063     5986    65535
  2993/2993 (buffers specified/carved), 14.92%, 1568 byte data size
  3         16064     19056     2993    65535
  598/598 (buffers specified/carved), 2.98%, 4544 byte data size
  4         19057     19654     598     65535
  398/398 (buffers specified/carved), 1.98%, 9248 byte data size
  5         19655     20052     398     65535
IPC Queue:
  100/100 (buffers specified/carved), 0.49%, 4112 byte data size
  30       77        76        100     65535
Raw Queue:
  31       0         82        0        65535
Interface Queues:
  0         0         0         0        65535
  1         0         0         0        65535
  2         0         0         0        65535
  3         0         0         0        65535

```

Gebruik de opdracht **TX-CoS** om een **cos-wachtrij-groepssjabloon** op een interfacewachtrij toe te passen. Zoals hier wordt aangegeven, past u de parameter direct op de interface toe. er zijn geen tabellen nodig . In dit voorbeeld is *pos48* de naam van een parameter set.

```

Router(config)#interface POS 4/0
Router(config-if)#tx-cos ?
WORD Name of cos-queue-group
Router(config-if)#tx-cos pos48

```

Gebruik de opdracht **Show cos** om uw configuratie te bevestigen:

```

Router#show cos
!--- Only some of the fields are visible if MDRR is configured on Inbound !--- or Outbound
interfaces. Interface Queue cos Group Gi4/0 eng2-frfab !--- TX-cos has been applied. Rx Slot
Slot Table 4 table1 !--- rx-cos-slot has been applied. Slot Table Name - table1 1 eng0-tofab 3
eng0-tofab !--- slot-table-cos has been defined. cos Queue Group - eng2-tofab !--- cos-queue-
group has been defined. Prec Red Label [min, max, prob] Drr Queue [deficit] 0 0 [6000, 15000,
1/1] 0 [10] 1 1 [10000, 20000, 1/1] 1 [40] 2 1 [10000, 20000, 1/1] 1 [40] 3 1 [10000, 20000,
1/1] 0 [10] 4 2 [15000, 25000, 1/1] 2 [80] 5 2 [15000, 25000, 1/1] 2 [80] 6 no drop low latency
7 no drop low latency

```

**Opmerking:** de legacy CLI maakt ook gebruik van de prioriteitssyntaxis voor Multiprotocol Label Switching (MPLS) verkeer. De router behandelt de bits MPLS alsof ze IP-bits Type of Service (ToS) bits zijn en zet de juiste pakketten in de juiste wachtrijen. Dit geldt helemaal niet voor de kwantitatieve versoepeling. MPLS QoS valt buiten het toepassingsgebied van dit document.

## [Gebruik modulair QoS CLI \(MQC\) voor configuratie](#)

Het doel van Cisco's modulaire QoS CLI (MQC) is om alle verschillende QoS-functies op een logische manier aan te sluiten, om de configuratie van Cisco IOS Software Quality of Service (QoS) te vereenvoudigen. De classificatie wordt bijvoorbeeld afzonderlijk uitgevoerd van de wachtrijen, controle en vormgeving. Het biedt één configuratiekader voor QoS dat op sjabloon



gebaseerd is. Hier zijn een paar punten om te herinneren aan de MQC-configuratie:

- U kunt deze applicatie eenvoudig toepassen op en verwijderen uit een interface.
- Het kan gemakkelijk opnieuw worden gebruikt (hetzelfde beleid kan op meerdere interfaces worden toegepast).
- Het biedt één configuratiekader voor QoS dat u in staat stelt om gemakkelijk voorziening, monitor en probleemoplossing te bieden.
- Het biedt een hoger niveau van abstractie.
- Het is een platform dat onafhankelijk is.

Op de Cisco 12000 Series kunnen MQC-opdrachten worden gebruikt in plaats van de syntaxis van de legacy Class of Service (CoS).

MQC-ondersteuning op Cisco 12000 Series betekent niet dat dezelfde QoS-functie die op een ander platform beschikbaar is, zoals Cisco 7500 Series, nu beschikbaar is op Cisco 12000. De MQC biedt een gemeenschappelijke syntaxis waarin een opdracht resulteert in een gedeelde functie of gedrag. Bijvoorbeeld, het **bandbreedte** bevel voert een minimum bandbreedtegarantie uit. Cisco 12000 Series gebruikt MDRR als het planningsmechanisme om de bandbreedtereservering te maken, terwijl Cisco 7500 Series WFQ gebruikt. Het belangrijkste algoritme vult het specifieke platform aan.

Belangrijk is dat alleen de FrFab-wachtrijen de configuratie van QoS-functies via de MQC ondersteunen. Omdat de ToFab-wachtrijen virtuele uitvoerrijen zijn en geen echte invoerwachtrijen, worden deze niet ondersteund door de MQC. Ze moeten worden ingesteld met CoS-opdrachten van de vorige taak.

[Tabel 5](#) biedt ondersteuning voor de MQC per L3-motortype.

**Tabel 5 - Ondersteuning van MQC voor L3-motortypen**

L3 Type motor	Engine 0	Engine 1	Engine 2	Engine 3	Engine 4	Engine 4+
MQC-ondersteuning	Ja	Nee	Ja	Ja	Ja	Ja
IOS-softwareversie	12.0(15)S	-	12.0(15)S <sup>1</sup>	12.0(21)S	12.0(22)S	12.0(22)S

<sup>1</sup> Onthoud deze uitzonderingen met MQC steun op Engine 0 en 2 lijnkaarten (LC) s:

- 2xCHOC3/STM1 - Inleiding in 12.0(17)S.
- 1xOC48 DPT - geïntroduceerd in 12.0(18)S.
- 8xOC3 ATM - gepland voor 12.0(22)S.

De MQC gebruikt deze drie stappen om een QoS-beleid te creëren:

1. Definieer een of meer verkeersklassen met de **class-map** opdracht.
2. Maak een QoS-beleid met de opdracht **beleidskaart** en wijs QoS-acties toe zoals **bandbreedte** of **prioriteit** aan een genoemde verkeersklasse.

3. Gebruik de opdracht **service-beleid** om een beleidskaart aan de FrFab-wachtrij van een uitgaande interface toe te voegen.

Gebruik de opdracht **Beleids-Kaart-interface tonen** om uw beleid te controleren.

Zie [Modular Quality of Service Optie Interface - Overzicht](#) voor meer informatie.

### Stap 1 - Leerkaarten definiëren

De opdracht **class-map** wordt gebruikt om verkeersklassen te definiëren. Intern, op Cisco 12000 Series, wijst de **class-map** opdracht een klasse toe aan een specifieke CoS wachtrij op de lijnkaart (zie [Stap 4](#) voor details).

De opdracht **class-map** ondersteunt "match-any", waar pakketten worden geplaatst die overeenkomen met een van de overeenkomende statements in de class en "match-all", die pakketten alleen in deze class plaatsen als alle statements waar zijn. Deze opdrachten maken een klasse met de naam "Prec\_5" en classificeren alle pakketten met een IP-voorrang van 5 naar deze klasse:

```
Router(config-cmap)#match ?
  access-group      Access group
  any               Any packets
  class-map        Class map
  destination-address Destination address
  fr-dlci          Match on fr-dlci
  input-interface  Select an input interface to match
  ip               IP specific values
  mpls             Multi Protocol Label Switching specific values
  not              Negate this match result
  protocol         Protocol
  qos-group        Qos-group
  source-address   Source address
Router(config-cmap)#match ip precedence 5
```

[Tabel 6](#) bevat de ondersteunde matchcriteria voor elk L3-motortype.

**Tabel 6 - Ondersteunde aanpassingscriteria voor L3-motoren**

	Modus 0, 2	Engine 3	Engine 4	Engine 4+
ip-voorrang	Ja	Ja	Ja	Ja <sup>1</sup>
toegangs groep	Nee	Ja	Nee	Nee
mpls exp	Nee	Ja	Nee	Ja (12.0.26S)
IP-telefoon	Nee	Ja	Nee	Ja (12.0.26S)
qos-groep	Nee	Ja	Nee	Nee
match-input-	Nee	Ja (alleen ontvange	Nee	Nee

interface POS x/y		n beleid)		
----------------------	--	-----------	--	--

<sup>1</sup> ingang/uitgang sinds 12.0.26S

## [Stap 2 - Maak een beleidskaart](#)

Het **beleid-kaart** bevel wordt gebruikt om pakket behandelingsbeleid of acties aan één of meer gedefinieerde klassen toe te wijzen. Bijvoorbeeld, wanneer u een bandbreedtereservering toewijst, of een willekeurig dalingsprofiel toepast.

Cisco 12000 Series ondersteunt een subset van MQC-functies, gebaseerd op de snelle architectuur van de L3-motoren. [Tabel 7](#) toont de opdrachten die worden ondersteund:

**Tabel 7 - Ondersteunde opdrachten**

Opdracht	Beschrijving
bandbreedte	Biedt een minimale bandbreedtegarantie tijdens perioden van congestie. Het wordt gespecificeerd als percentage van de verbindingssnelheid of als een absolute waarde. Als een klasse geen bandbreedte gelijk aan de gereserveerde kbps gebruikt of nodig heeft, kan de beschikbare bandbreedte door andere bandbreedte klassen worden gebruikt.
politie, vorm	Bepenkt de hoeveelheid verkeer die een klasse kan verzenden. Deze opdrachten hebben een iets andere functie. De <b>politie</b> opdracht identificeert verkeer dat de ingestelde bandbreedte overschrijdt en laat het vallen of opmerken. De <b>vorm</b> commando buffert elk overtollig verkeer en organiseert het voor transmissie met een constante snelheid, maar laat het niet vallen of opmerken.
Wachrijlimiet	wijst een vaste rijlengte aan een bepaalde klasse van verkeer toe. U kunt dit specificeren in een aantal pakketten die in de wachrij kunnen worden gehouden.
prioriteit	Identificeert een rij als een rij met lage latentie. MQC ondersteunt strikte modus alleen voor een PQ. De alternatieve modus wordt niet ondersteund door MQC. Gebruik de opdracht <b>prioriteit</b> zonder een procentwaarde om de strikte prioriteitsmodus mogelijk te maken. <b>Opmerking:</b> de implementatie van de <b>prioriteits</b> opdracht op Cisco 12000 Series verschilt van de implementatie op andere routers die Cisco IOS-software gebruiken. Op dit platform is het prioriteitsverkeer niet beperkt tot de geconfigureerde kbps-waarde tijdens

	congestieperioden. Dientengevolge, moet u ook de <b>politie</b> opdracht vormen om te beperken hoeveel bandbreedte een prioriteitsklasse kan gebruiken en om adequate bandbreedte voor andere klassen te verzekeren. Op dat moment wordt de <b>politie</b> -opdracht alleen ondersteund op Engine 3 lijnkaarten. Op de andere motorlijnkaarten is alleen class-default toegestaan wanneer u een prioriteitsklasse configureren.
willekeurige detectoren	Kent een WRED-profiel toe. Gebruik de opdracht <b>precedence</b> op willekeurige <b>detectie</b> om niet-standaard WRED-waarden per IP-prioriteitswaarde te configureren.

Op engine 3 LCs moet u de FrFab-wachtrijen configureren met de modulaire QoS CLI (MQC); de CLI (legacy Opdracht Line) wordt niet ondersteund.

Wanneer u de **bandbreedte**-opdracht configureren, let op dat Engine 0 en 2 LCs alleen zes bandbreedte-klassen ondersteunen. Een zevende klasse kan worden gebruikt voor de dienst met lage latentie en een achtste klasse, die class-default is, wordt gebruikt voor al het niet-overeenkomende verkeer. Daarom heb je in totaal acht rijen. Class-default wordt niet gebruikt als prioriteitsklasse.

Op Engine 3 LCs wordt de opdracht **bandbreedteprocent** vertaald in een kbps waarde, die varieert met de onderliggende linksnelheid en vervolgens direct op de wachtrij wordt ingesteld. De nauwkeurigheid van deze minimale bandbreedte-garantie is 64 kbps.

Hoewel er geen conversie naar een kwantumwaarde wordt uitgevoerd met de **bandbreedte** opdracht, hebben alle wachtrijen een kwantum. Op motor 3 LC's wordt de kwantumwaarde intern ingesteld op basis van de maximale transmissie-eenheid (MTU) van de interface en voor alle wachtrijen op gelijke wijze ingesteld. Er is geen MQC CLI-mechanisme om deze kwantumwaarde direct of indirect aan te passen. De kwantumwaarde moet groter zijn dan of gelijk aan de interface MTU. Intern is de kwantumwaarde in eenheden van 512 bytes. Met een MTU van 4470 bytes moet de minimale kwantumwaarde van MTU dus 9 zijn.

## [MDRR op motor 3 LC](#)

Deze sectie verschaft configuratienotities om WRED en MDRR te implementeren op engine 3 LCs.

- De bandbreedte van MDRR die in CLI is geconfigureerd wordt vertaald naar een bedrag dat overeenkomt met L2 (de overhead L1 wordt bijvoorbeeld verwijderd). Dat bedrag wordt dan afgerond tot de volgende 64 kbps en geprogrammeerd in hardware.
- Drie verschillende WRED-profielen worden voor één klasse ondersteund.
- De WRED (maximumdrempel - minimumdrempel) is ongeveer gelijk aan het dichtstbijzijnde vermogen van 2. De minimumdrempel wordt dan automatisch aangepast, terwijl de maximumdrempel ongewijzigd blijft.
- Mark kanswaarde 1 wordt ondersteund.
- Uitzonderlijke wegingsconstante configuratie wordt niet ondersteund.
- IP-voorrang, MPLS EXP-bits en DSCP-waarden worden ondersteund.

**Opmerking:** Elke poort of elk kanaal op de Tetra (4GE-SFP-LC=) of CHOC12/DS1-IR-SC= Frostbite linecards hebben vier wachtrijen die standaard worden toegewezen. De vier wachtrijen bestaan uit:

- LLQ-klasse (één prioriteitswachtrij)
- Eén standaardrijklasse
- Twee normale niet-prioriteitsklassen

Bij de toepassing van een dienstverleningsbeleid dat meer dan deze vier klassen (1 hoofdkwartier, 2 LPQ's en class-default) op de interface bevat, wordt de volgende fout gerapporteerd:

**Router (configuratie-als)#service-beleid mdr-beleid**

**% onvoldoende middelen om in de wachtrij te voorzien om aan het verzoek te voldoen.**

Vanaf 12.0(26)S is er een opdracht toegevoegd voor de 4GE-SFP-LC= Tetra lijnkaart die de configuratie van acht wachtrijen/VLAN in plaats van vier toestaat. De acht wachtrijen bestaan uit:

- Eén LLQ
- Eén class-default wachtrij
- Zes normale wachtrijen

Het gebruik van deze opdracht zal een microcode-herlading van de lijnkaart vereisen en zal in de mogelijkheid resulteren om slechts 508 VLANs in plaats van 1022 te configureren. De opdrachtsyntax is als volgt:

**[nee] sleuf <sleuf#> QoS interfacewachtrijen 8**

Bijvoorbeeld:

**Routersleuf 2 qos interfacewachtrijen 8**

**Waarschuwing: Laad de lijnkaart om deze opdracht in werking te stellen**

**Router (configuratie)#microcode reload 2**

Deze opdracht is beschikbaar voor de CHOC12/DS1-IR-SC= Frostbite linecard in 12.0(32)S

**Voorbeeld #1 - bandbreedte percentage opdracht**

Dit voorbeeld wijst 20% van beschikbare bandbreedte toe aan klasse Prec\_4 verkeer en 30% aan verkeer van klasse Prec\_3 verkeer. Het laat de resterende 50 procent over aan de class-default klasse.

Daarnaast vormt het WRED als het valmechanisme op alle gegevensklassen.

#### Voorbeeld #1 - bandbreedte percentage

```
policy-map GSR_EXAMPLE
  class Prec_4
    bandwidth percent 20
    random-detect
    random-detect precedence 4 1498 packets 9690 packets 1
  !--- All data classes should have WRED configured. class
  Prec_3 bandwidth percent 30 random-detect random-detect
```

```

precedence 3 1498 packets 9690 packets 1 class class-
default !--- Class-default uses any leftover bandwidth.
random-detect random-detect precedence 2 1498 packets
9690 packets 1 random-detect precedence 1 1498 packets
9690 packets 1 random-detect precedence 0 1498 packets
9690 packets 1

```

## Voorbeeld #2 - bandbreedte {kbps} Opdracht

Dit voorbeeld illustreert hoe de bandbreedte-opdracht als een absolute kbps waarde in plaats van een percentage moet worden toegepast.

### Voorbeeld #2 - bandbreedte {kbps}

```

policy-map GSR_EXAMPLE
  class Prec_4
    bandwidth 40000
    !--- Configures a minimum bandwidth guarantee of 40000
    kbps or 40 Mbps in !--- times of congestion. Random-
    detect random-detect precedence 4 1498 packets 9690
    packets 1 class Prec_3 bandwidth 80000 !--- Configures a
    minimum bandwidth guarantee of 80000 kbps or 80 Mbps in
    !--- times of congestion. Random-detect random-detect
    precedence 3 1498 packets 9690 packets 1 class class-
    default !--- Any remaining bandwidth is given to class-
    default. Random-detect random-detect precedence 2 1498
    packets 9690 packets 1 random-detect precedence 1 1498
    packets 9690 packets 1 random-detect precedence 0 1498
    packets 9690 packets 1

```

## Voorbeeld #3 - Opdracht met prioriteit

Dit voorbeeld is ontworpen voor serviceproviders die de Cisco 12000 Series router als een MPLS provider edge (PE) router gebruiken en een QoS service beleid moeten configureren op het link tussen de PE-router en de Customer Edge (CE) router. Het plaatst IP voorrang 5 pakketten in een prioritaire rij, en beperkt de uitvoer van die rij tot 64 Mbps. Het wijst dan een deel van de resterende bandbreedte aan de bandbreedteklassen toe.

Alle wachtrijen die geen prioriteit hebben, worden ingesteld met de opdracht **aselecte** detectie om WRED als valbeleid in te schakelen. Alle bandbreedte class en class-default moeten expliciet WRED zijn ingesteld.

### Voorbeeld #3 - prioriteit

```

policy-map foo
  class Prec_5
    police 64000000 conform-action transmit exceed-
    action drop
    !--- The police command is supported on Engine 3 line
    cards. priority class Prec_4 bandwidth percent 30
    random-detect random-detect precedence 4 1498 packets
    9690 packets 1 class Prec_3 bandwidth percent 10 random-
    detect random-detect precedence 3 1498 packets 9690
    packets 1 class Prec_2 bandwidth percent 10 random-
    detect random-detect precedence 2 1498 packets 9690
    packets 1 class Prec_1 bandwidth percent 10 random-
    detect random-detect precedence 1 1498 packets 9690
    packets 1 class Prec_0 bandwidth percent 25 random-

```

```
detect random-detect precedence 0 1498 packets 9690
packets 1 class class-default random-detect random-
detect precedence 6 1498 packets 9690 packets 1 random-
detect precedence 7 1498 packets 9690 packets 1
```

### Stap 3 - Wijs een beleidskaart aan een uitgaande interfacekaart toe

Zoals hierboven vermeld werkt de MQC alleen met de FrFab-wachtrijen op een uitgaande interface. Om een bepaald beleid-kaart toe te passen, gebruik de **service-beleid output** opdracht, zoals hier getoond wordt:

```
Router(config)#interface POS 0/0
Router(config-if)#service-policy ?
  history  Keep history of QoS metrics
  input    Assign policy-map to the input of an interface
  output   Assign policy-map to the output of an interface
Router(config-if)#service-policy output ?
  WORD    policy-map name
Router(config-if)#service-policy output GSR_EXAMPLE
```

### Stap 4 - Toezicht op en verificatie van het servicebeleid

Gebruik de opdracht **Beleids-kaart tonen** om de toepassing van een beleid te bekijken. De opdracht **Show beleid-map** toont het volgende:

- Configureerde bandbreedte- en prioriteitsklassen en de matchingcriteria.
- Alle WRED-profielen.
- Vorm en politie parameters.
- Verkeersboekhouding en -tarieven.
- De interne CoS rij waaraan een bepaalde klasse in kaart wordt gebracht. Deze wachtrijen worden van referentiekader voorzien door dezelfde index die wordt gebruikt in de uitvoer van de opdracht voor een **controller-feldwachtrij**.

Hier is een voorbeeld van een volledige configuratie en de **show** opdrachten om het beleid te controleren:

#### **Complete configuratie**

```
class-map match-all class1
  match ip precedence 1
class-map match-all class2
  match ip precedence 2
!--- Step 1 - Configure traffic classes. ! policy-map
policyle Class class1 bandwidth percent 10 random-detect
random-detect precedence 1 375 packets 2423 packets 1
Class class2 bandwidth percent 20 random-detect !---
Step 2 - Configure a policy-map. ! interface POS6/0 ip
address 12.1.1.1 255.255.255.0 no ip directed-broadcast
no keepalive service-policy output policyle !--- Step 3-
Attach policy-map to the interface.
```

Gebruik de opdracht **Show beleid-map** om het beleid te bekijken dat op de interface is geconfigureerd, samen met alle geconfigureerde klassen. Dit is de uitvoer van de opdracht:



Router#show policy-map int pos6/0

POS6/0

Service-policy output: policyle (1071)

Class-map: class1 (match-all) (1072/3)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: ip precedence 1 (1073)

Class of service queue: 1

Tx Queue (DRR configured)

bandwidth percent Weight

10 1

Tx Random-detect:

Exp-weight-constant: 1 (1/2)

Precedence	RED Label	Min	Max	Mark
------------	-----------	-----	-----	------

1	1	375	2423	1
---	---	-----	------	---

Class-map: class2 (match-all) (1076/2)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: ip precedence 2 (1077)

Class of service queue: 2

Tx Queue (DRR configured)

bandwidth percent Weight

20 9

Tx Random-detect:

Exp-weight-constant: 1 (1/2)

Precedence	RED Label	Min	Max	Mark
------------	-----------	-----	-----	------

Class-map: class-default (match-any) (1080/0)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: any (1081)

0 packets, 0 bytes

5 minute rate 0 bps

## Opdrachten voor het bewaken van congestiebeheer en -preventie

Dit gedeelte bevat de opdrachten die u kunt gebruiken om het beleid voor congestiebeheer en -vermijding te controleren.

[Tabel 8](#) bevat de opdrachten voor de lijnkaarten van Ingress en Egress.

**Tabel 8 - Opdrachten voor de lijnkaarten**

Ingress-lijnkaart	Egress-lijnkaart
<ul style="list-style-type: none"><li>• tonen interfaces</li><li>• verbindingssleuf &lt;x&gt; sh controller in de wachtrij</li><li>• Exec-sleuf &lt;x&gt; toont controller-fab-wachtrij &lt;sleuf&gt; &lt;poort&gt;</li><li>• uitvoersleuf &lt;x&gt; toont controller-fab qm stat</li></ul>	<ul style="list-style-type: none"><li>• tonen interfaces</li><li>• interfaces &lt;y&gt; willekeurig tonen</li><li>• uitvoersleuf &lt;y&gt; Geeft controller-frafb-wachtrij weer</li><li>• uitvoersleuf &lt;y&gt; Geeft controller in de wachtrij</li></ul>

	<p>&lt;poort&gt; weer</p> <ul style="list-style-type: none"> <li>• Sleuf &lt;y&gt; toont controller van QM stat</li> </ul>
--	--

Deze opdrachten worden in deze sectie uitgelegd.

## Opdracht beide interfaces

Voordat u deze opdracht gebruikt, dient u de juiste "Wachtrij-strategie" te bevestigen. Als de uitvoer First In, First Out (FIFO) toont, zorg er dan voor dat de opdracht **Service-beleid** verschijnt in de actieve configuratie (als MQC is gebruikt om MDRR te configureren).

Controleer het aantal uitvoerdruppels, die het totale aantal WRED FrFab-druppels vertegenwoordigen dat is voorgekomen voor uitgaande verkeer op deze interface. Het aantal uitvoerdruppels in de opdrachtoutput van **showinterfaces** moet gelijk zijn aan of hoger dan het aantal uitvoerdruppels in de **output** van **showinterfaces <aantal> willekeurige** opdracht.

**Opmerking:** Op Cisco 12000 Series router worden de druppels voor interface-uitvoer bijgewerkt nadat de WRED-druppels zijn bijgewerkt. Er is een kleine kans dat als u een gereedschap gebruikt om beide valtellers te vragen, de interfacekaarten nog niet zijn bijgewerkt.

```
Router#show interfaces POS 4/0
POS4/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: link to c12f9-1
  Internet address is 10.10.105.53/30
  MTU 4470 bytes, BW 622000 Kbit, DLY 100 usec, rely 255/255, load 82/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive set (10 sec)
  Scramble enabled
  LCP Open
  Open: IPCP, CDPCP, OSICP, TAGCP
  Last input 00:00:02, output 00:00:05, output hang never
  Last clearing of "show interface" counters 00:04:54
Queueing strategy: random early detection (WRED)
  Output queue 0/40, 38753019 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 200656000 bits/sec, 16661 packets/sec
    135 packets input, 6136 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
      0 parity
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  7435402 packets output, 11182627523 bytes, 0 underruns
  0 output errors, 0 applique, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
```

## De tonen interfaces {aantal} willekeurige Opdracht

Wanneer u deze opdracht gebruikt, dient u:

- Controleer dat de juiste **cos-wachtrij-groep** sjabloon op deze interface is toegepast.
- Controleer de MDRR-gewichten. Bij elke MDRR-wachtrij kunt u het gewogen gemiddelde voor de rijlengte en de hoogste waarde die wordt bereikt (in pakketten) controleren. De waarden

worden berekend als een gewogen gemiddelde en hoeven niet de werkelijk bereikte maximale rijdiepte weer te geven.

- Controleer de minimum- en maximumdrempels van de WRED.
- Controleer het aantal willekeurige druppels en drempeldruppels voor elk ROOD etiket ("Aan stof"-druppels geven de totale hoeveelheid druppels voor dit label op alle lijnkaarten aan).
- De "TX-wachtrij-limiet-teller" wordt alleen gebruikt voor Engine 1 LCs die geen WRED ondersteunen. Met motor 1-kaarten kunt u de limiet van de MDRR-wachtrijen instellen met de opdracht **TX-wachtrij-limiet**. Waar WRED wordt ondersteund, bepalen de WRED-drempels de diepte van de MDRR-wachtrijen.

```
Router#show interfaces POS 4/0 random
```

```
POS4/0
```

```
cos-queue-group: oc12
```

```
RED Drop Counts
```

#### TX Link

#### To Fabric

RED Label	Random	Threshold	Random	Threshold
<b>0</b>	<b>29065142</b>	<b>73492</b>	<b>9614385</b>	<b>0</b>
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0

```
TX-queue-limit drops: 0
```

#### Queue Lengths

```
TX Queue (DRR configured) oc12
```

Queue	Average	High Water Mark	Weight
0	<b>0.000</b>	<b>2278.843</b>	<b>1</b>
1	0.000	0.000	<b>73</b>
2	0.000	0.000	10
3	0.000	0.000	10
4	0.000	0.000	10
5	0.000	0.000	10
6	0.000	0.000	10
Low latency	0.000	0.000	10

#### TX RED config

```
Precedence 0: 375 min threshold, 2423 max threshold, 1/1 mark weight
```

```
Precedence 1: not configured for drop
```

```
Precedence 2: not configured for drop
```

```
Precedence 3: not configured for drop
```

```
Precedence 4: 375 min threshold, 2423 max threshold, 1/1 mark weight
```

```
Precedence 5: not configured for drop
```

```
Precedence 6: 375 min threshold, 2423 max threshold, 1/1 mark weight
```

```
Precedence 7: not configured for drop weight 1/2
```

## [De \(y\) sleuf van de uitvoer tonen de van de controller van de rij {haven} Opdracht](#)

Deze opdracht geeft de onmiddellijke rijdiepte voor een bepaalde poort op een bepaalde sleuf weer. De steekproefuitvoer in deze sectie toont de MDRR- rij op interface POS 4/1. U ziet een rijdiepte voor MDRR rij 1 van 1964 pakketten. Het gewicht is het aantal bytes dat op deze wachtrij kan worden geplaatst. Dit gewicht bepaalt het percentage van bandbreedte dat u aan deze rij wilt

geven. Het tekort is de waarde die het DRR-algoritme vertelt hoeveel pakketten er nog moeten worden bediend. U kunt zien dat er in de LLQ geen pakketten in de wachtrij staan (DRR-wachtrij 7).

```
Router#execute-on slot 4 show controllers frfab queue 1
===== Line Card (Slot 4) =====
FrFab Queue
Interface 1
DRR#      Head      Tail      Length  Average          Weight  Deficit
0         95330    40924     0        0.000           4608    0
1        211447   233337   1964     1940.156        41472   35036
2          0         0         0         0.000           9216    0
3          0         0         0         0.000           9216    0
4          0         0         0         0.000           9216    0
5          0         0         0         0.000           9216    0
6          0         0         0         0.000           9216    0
7          0         0         0         0.000           9216    0
```

Deze opdracht wordt met name gebruikt om de diepte van de Prioritaire Wachtrij van de Stress Line Card te controleren. Wanneer u ziet dat pakketten op deze LLQ beginnen te wachten, is het een goede indicatie dat u wat Voice over IP (VOIP) verkeer naar andere IP lijnkaarten moet afleiden. In een goed ontwerp moet de lengte altijd 0 of 1 zijn. In een echt netwerk zal je bursty verkeer ervaren, zelfs voor spraakgegevens. De extra vertraging wordt serieuzer wanneer de totale spraaklading voor een korte tijd groter is dan 100% van de enorme bandbreedte. De router kan niet meer verkeer op de draad zetten dan toegestaan is, en dus wordt het stemverkeer in de rij geplaatst op zijn eigen prioriteitswachtrij. Dit creëert spraaklatentie en spraakjitter geïntroduceerd door de uitbarsting van het spraakverkeer zelf.

```
Router#execute-on slot 4 show controllers frfab queue 0
===== Line Card (Slot 4) =====
FrFab Queue
Interface 0
DRR#      Head      Tail      Length  Average          Weight  Deficit
0        181008   53494    2487     2282.937         4608    249
1        16887    45447     7         0.000          41472    0
2          0         0         0         0.000           9216    0
3          0         0         0         0.000           9216    0
4          0         0         0         0.000           9216    0
5          0         0         0         0.000           9216    0
6          0         0         0         0.000           9216    0
7        107818  142207    93         0.000           9216   -183600
```

Wachtrij 7 is de LLQ en de lengte vertelt u hoeveel pakketten in deze LLQ zijn.

## [De uitvoersleuf \(y\) toont controller van QM stat-opdracht](#)

Gebruik deze opdracht als u vermoedt dat het pakketgeheugen van een LC begint te benaderen met de volledige capaciteit. Een stijgende waarde voor de "geen mem druppel"-teller suggereert dat WRED niet is ingesteld of dat de WRED-drempels te hoog zijn ingesteld. Deze teller mag onder normale omstandigheden niet toenemen. Zie [Problemen oplossen en Geheugende pakketten en geen geheugendruppels op de Cisco 12000 Series internetrouter](#) voor meer informatie.

```
Router#execute-on slot 4 show controllers frfab QM stat
```

```

===== Line Card (Slot 4) =====
68142538 no mem drop, 0 soft drop, 0 bump count
0 rawq drops, 8314999254 global red drops, 515761905 global force drops
0 no memory (ns), 0 no memory hwm (Ns)
no free queue
0      0      1968    88
0      0      0      0
0      0      0      0
0      0      0      0
0 multicast drops
TX Counts
  Interface 0
859672328848 TX bytes, 3908130535 TX pkts, 75431 kbps, 6269 pps
  Interface 1
86967615809 TX bytes, 57881504 TX pkts, 104480 kbps, 8683 PPS
  Interface 2
0 TX bytes, 0 TX pkts, 0 kbps, 0 PPS
  Interface 3
0 TX bytes, 0 TX pkts, 0 kbps, 0 PPS

```

## [Monitor inkomende congestiebeheer](#)

In dit gedeelte worden de opdrachten beschreven die worden gebruikt om het inkomende congestiebeheer te bewaken.

### [Opdracht beide interfaces](#)

Voordat u deze opdracht geeft, controleert u of de waarde in de genegeerde teller op de stijging staat. U ziet genegeerde pakketten als u niet voldoende geheugen aan de kant ToFab hebt of als de lijnkaart de pakketten niet snel genoeg accepteert. Zie [Invoerdruppels voor probleemoplossing op de Cisco 12000 Series internetrouter voor](#) meer informatie.

```

Router#show interfaces POS 14/0
POS14/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: agilent 3b for QOS tests
  Internet address is 10.10.105.138/30
  MTU 4470 bytes, BW 2488000 Kbit, DLY 100 usec, rely 234/255, load 1/255
  Encapsulation HDLC, crc 32, loopback not set
  Keepalive not set
  Scramble disabled
  Last input never, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:34:09
  Queueing strategy: random early detection (WRED)
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 2231000 bits/sec, 4149 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    563509152 packets input, 38318622336 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
      0 parity
    166568973 input errors, 0 CRC, 0 frame, 0 overrun, 166568973 ignored, 0 abort
    35 packets output, 12460 bytes, 0 underruns
    0 output errors, 0 applique, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions

```

### [De extra sleuf \(x\) toont controller in de wachtrij](#)

Deze voorbeelduitvoer van de **exec sleuf <x> show controller tofab wachtrij** opdracht werd opgenomen toen er geen stremming op een speerlijnkaart in sleuf 3 was.

```
Router#execute-on slot 13 show controllers tofab queue
===== Line Card (Slot 13) =====
Carve information for ToFab buffers
!--- Output omitted. ToFab Queues: Dest Slot 0 0 0 0 9690 1 0 0 0 9690 2 0 0 0 9690 3 11419
16812 0 9690 4 0 0 0 2423 5 0 0 0 9690 6 0 0 0 9690 7 0 0 0 262143 8 0 0 0 262143 9 0 0 0 606 10
0 0 0 262143 11 0 0 0 262143 12 0 0 0 262143 13 0 0 0 262143 14 0 0 0 262143 15 0 0 0 9690
Multicast 0 0 0 262143
```

De volgende uitvoer werd opgenomen wanneer er sprake was van een congestie op sleuf 3:

```
Router#execute-on slot 13 show controllers tofab queue
===== Line Card (Slot 13) =====
Carve information for ToFab buffers
!--- Output omitted. ToFab Queues: Dest Slot 0 0 0 0 9690 1 0 0 0 9690 2 0 0 0 9690 3 123689
14003 1842 9690 4 0 0 0 2423 5 0 0 0 9690 6 0 0 0 9690 7 0 0 0 262143 8 0 0 0 262143 9 0 0 0 606
10 0 0 0 262143 11 0 0 0 262143 12 0 0 0 262143 13 0 0 0 262143 14 0 0 0 262143 15 0 0 0 9690
Multicast 0 0 0 262143
```

## [De extra sleuf \(x\) toont controller-tofab wachtrij \(sleuf\) \(poort\) Opdracht](#)

Gebruik deze opdracht om te bepalen hoeveel geheugen aan de kant ToFab wordt gebruikt. Let met name op het nummer in de '#Qelem'-kolom. Let op:

- Als er geen geheugen wordt gebruikt, hebben de waarden de hoogste waarde.
- De waarde van de kolom "#Qelem" daalt naarmate pakketten worden gebufferd.
- Als de '#Qelem' kolom nul bereikt, zijn alle gekerfde buffers in gebruik. Op Engine 2 LC kunnen kleine pakketten bufferruimte lenen uit grotere pakketten.

U kunt deze opdracht ook gebruiken om het aantal pakketten in de wachtrij voor een virtuele uitvoerwachtrij te bepalen. Het voorbeeld hier toont hoe u sleuf 14 voor het momentane aantal pakketten op deze wachtrijen kunt controleren voor sleuf 4, poort 1 (POS 4/1). We zien 830 pakketten in de wachtrij voor MDRR 1 staan.

```
Router# execute-on slot 14 show controllers tofab queue 4 1
===== Line Card (Slot 14) =====
ToFab Queue
Slot 4 Int 1
DRR#      Head      Tail      Length  Average      Weight  Deficit
0         0         0         0        0.000       4608   0
1       203005    234676    830      781.093     41472  37248
2         0         0         0        0.000       9216   0
3         0         0         0        0.000       9216   0
4         0         0         0        0.000       9216   0
5         0         0         0        0.000       9216   0
6         0         0         0        0.000       9216   0
7         0         0         0        0.000       9216   0
```

## [De uitvoersleuf \(x\) toont controller-fab QM stat-opdracht](#)

Gebruik deze opdracht om het aantal ToFab-druppels per lijnkaart te zien. Controleer ook op een "geen geheugendruppel"-teller die extra wordt versterkt. Deze tegengtoename wordt ingesteld als CoS niet aan de kant ToFab is ingesteld.

```

Router#execute-on slot 13 show controllers tofab QM stat
===== Line Card (Slot 13) =====
0 no mem drop, 0 soft drop, 0 bump count
0 rawq drops, 1956216536 global red drops, 6804252 global force drops
0 no memory (Ns), 0 no memory hwm (Ns)
no free queue
0      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0
Q status errors
0      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0

```

## casestudy

Deze casestudy toont hoe u een typisch beleid voor de netwerkkern van een serviceprovider-omgeving kunt configureren. Het past rijopdrachten toe en stelt u in staat om MDRR/WRED te gebruiken voor actief rijbeheer. QoS-beleid in randrouters maakt normaal gebruik van verkeersmarkering, conditionering, enzovoort, om routers in de kern in staat te stellen verkeer in klassen te sorteren op basis van IP-voorrang of DSCP-waarden (DiffServ Code Point). Deze casestudy's maken gebruik van Cisco IOS-software QoS-functies om te voldoen aan strikte Service Level Agreements (SLA's) en verschillende serviceniveaus voor spraak-, video- en gegevensservices op dezelfde IP-backbone.

In de aanpak heeft een dienstverlener drie verkeersklassen ingevoerd. Het belangrijkste is de LLQ- of Low Latency Queueing-klasse. Dit is de klasse voor Spraak en Video. Deze klasse moet een minimum vertraging en jitter ervaren en moet nooit pakketverlies of opnieuw geordende pakketten ervaren zolang de bandbreedte van deze klasse de verbindingbandbreedte niet overschrijdt. Deze klasse is gekend als het Versnelde Doorsturen Per-Hop Behavior (EF PHB) verkeer in de DiffServ architectuur. De Internet Service Provider (ISP) ontwierp het netwerk op een manier dat deze klasse niet groter is dan 30% op gemiddelde lading van de link. De andere twee klassen zijn de bedrijfsklasse en de beste inspanningsklasse.

In het ontwerp hebben we de routers zodanig geconfigureerd dat de business class altijd 90% van de resterende bandbreedte krijgt en de best inspanning class 10%. Deze twee klassen hebben minder tijdgevoelig verkeer en kunnen verkeersverlies, hogere vertraging, en jitter ervaren. In het ontwerp ligt de nadruk op Engine 2 lijnkaarten: 1xOC48 rev B, 4xOC12 rev B en 8xOC3 lijnkaarten.

Rev B-lijnkaarten zijn het meest geschikt om VoIP-verkeer over te brengen vanwege een herziene ASIC- en hardwarearchitectuur, die zeer weinig latentie met zich meebrengt. Met de herziene ASIC wordt de rij voor verzendende FIFO door de lijnkaartbestuurder geremd tot ongeveer twee keer de grootste MTU op de kaart. Zoek een "-B" bij het onderdeelnummer, zoals OC48E/POS-SR-SC-B=.

**Opmerking:** Verwar de rij FIFO voor verzenden niet met de wachtrijen FrFab die op de lijnkaarten van Engine 0 kunnen worden afgestemd met de opdracht voor de belasting-in-rij-limiet interface.

[Tabel 9](#) bevat de matchingscriteria voor elke categorie.



**Tabel 9 - Overeenkomende criteria voor elke klasse**

Naam van klasse	Overeenkomende criteria
Prioritaire wachtrij voor spraakverkeer	Voorzorgsmaatregelen 5
Zakelijke wachtrijen	Voorzorgsmaatregelen 4
Best Performance Quelling	Voorzorgsmaatregelen 0

De OC48 lijnkaarten kunnen een groot aantal pakketten in de rijen van ToFab in rij brengen. Daarom is het belangrijk om MDRR/WRED op de Wachtrijen van ToFab te configureren, vooral wanneer de spanning-interface een snelle interface is zoals OC48. Het weefsel kan alleen verkeer naar de ontvangende lijnkaart switches met een theoretisch maximum aantal van 3 Gbps (1500 bytes). Als de totale hoeveelheid verzonden verkeer groter is dan wat het switching materiaal naar zijn ontvangende kaart kan dragen, zullen veel pakketten in de wachtrij worden geplaatst op ToFab.

```

Interface POS3/0
  description OC48 egress interface
  ip address 10.10.105.53 255.255.255.252
  no ip directed-broadcast
  ip router Isis encapsulation ppp
  mpls traffic-eng tunnels
  tag-switching ip
  no peer neighbor-route
  crc 32
  clock source internal
  POS framing sdh
  POS scramble-atm
  POS threshold sf-ber 4
  POS flag s1s0 2
  TX-cos oc48
  Isis metric 2 level-1
  Isis metric 2 level-2
  ip rsvp bandwidth 2400000 2400000
!
interface POS4/1
  description OC12 egress interface
  ip address 10.10.105.121 255.255.255.252
  no ip directed-broadcast
  ip router Isis encapsulation ppp
  mpls traffic-eng tunnels
  no peer neighbor-route
  crc 32
  clock source internal
  POS framing sdh
  POS scramble-ATM POS threshold sf-ber 4
  POS flag s1s0 2
  TX-cos oc12
  Isis metric 2 level-1
  Isis metric 2 level-2
  ip RSVP bandwidth 600000 60000
!
interface POS9/2
  description OC3 egress interface
  ip address 10.10.105.57 255.255.255.252
  no ip directed-broadcast
  ip router Isis crc 16
  POS framing sdh
  POS scramble-ATM POS flag s1s0 2

```

```

TX-cos oc3
Isis metric 200 level-1
Isis metric 2 level-2
!
interface POS13/0
description agilent 3a for QOS tests - ingress interface.
ip address 10.10.105.130 255.255.255.252
no ip directed-broadcast
no ip route-cache cef
no ip route-cache
no ip mroute-cache
no keepalive
crc 32
POS threshold sf-ber 4
TX-cos oc48
!
interface POS14/0
description agilent 3b for QOS tests - ingress interface.
ip address 10.10.105.138 255.255.255.252
no ip directed-broadcast
no keepalive
crc 32
POS threshold sf-ber 4
TX-cos oc48
!
interface POS15/0
description agilent 4A for QOS tests - ingress interface
ip address 10.10.105.134 255.255.255.252
no ip directed-broadcast
no ip mroute-cache
no keepalive
crc 32
POS threshold sf-ber 4
TX-CoS oc48
!
rx-cos-slot 3 StotTable
rx-cos-slot 4 StotTable
rx-cos-slot 9 StotTable
rx-cos-slot 13 StotTable
rx-cos-slot 14 StotTable
rx-cos-slot 15 StotTable
!
slot-table-cos StotTable
destination-slot 0 oc48
destination-slot 1 oc48
destination-slot 2 oc48
destination-slot 3 oc48
destination-slot 4 oc12
destination-slot 5 oc48
destination-slot 6 oc48
destination-slot 9 oc3
destination-slot 15 oc48
!
cos-queue-groupoc3
precedence 0 random-detect-label 0
precedence 4 queue 1
precedence 4 random-detect-label 1
precedence 5 queue low-latency
precedence 6 queue 1
precedence 6 random-detect-label 1
random-detect-label 0 94 606 1
random-detect-label 1 94 606 1
queue 0 1
queue 1 73

```

```

queue low-latency strict-priority
!--- Respect the tight SLA requirements. !--- No packets drop/low delay and jitter for the
priority queue. ! CoS-queue-groupoc12
precedence 0 random-detect-label 0
precedence 4 queue 1
precedence 4 random-detect-label 1
precedence 5 queue low-latency
precedence 6 queue 1
precedence 6 random-detect-label 1
random-detect-label 0 375 2423 1
random-detect-label 1 375 2423 1
queue 0 1
queue 1 73
queue low-latency strict-priority
!
CoS-queue-groupoc48
precedence 0 random-detect-label 0
precedence 4 queue 1
precedence 4 random-detect-label 1
precedence 5 queue low-latency
precedence 6 queue 1
precedence 6 random-detect-label 1
random-detect-label 0 1498 9690 1
random-detect-label 1 1498 9690 1
queue 0 1
queue 1 73
queue low-latency strict-priority

```

Verwacht wordt dat hoe meer VOIP-verkeer u heeft, hoe meer zakelijke verkeer moet wachten voordat het wordt bediend. Dit is echter geen probleem omdat de strakke SLA geen pakketdruppels vereist, en een zeer lage latentie en jitter voor de prioriteitswachtrij.

## [Gerelateerde informatie](#)

- [De uitvoer van het beeldcontrollerformaat lezen | Wachtrij voor tofab op Cisco 12000 Series Internet-router](#)
- [Problemen oplossen en genezen pakketjes zonder geheugen verminderen op Cisco 12000 Series Internet-router](#)
- [Stapels voor probleemoplossing in de Cisco 12000 Series Internet-router](#)
- [Weighted Random vroege Detectie op Cisco 12000 Series router](#)
- [Overzicht van de servicelijn met modulaire kwaliteit](#)
- [Ondersteuning van 12000 Series internetrouters](#)
- [Technische ondersteuning - Cisco-systemen](#)