

# De limieten van wachtrijen en de uitgangsdalingen op IOS-softwareplatforms begrijpen

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[Op klasse gebaseerde Weighted Fair Queueing Primer](#)

[De limieten van de wachtrij en de uitgangsdalingen begrijpen](#)

[Door gebruiker gedefinieerde klassen geconfigureerd met de opdracht "prioriteit"](#)

[Pre-HQF gedrag](#)

[HQF-gedrag](#)

[Door gebruiker gedefinieerde klassen geconfigureerd met de opdracht "bandbreedte"](#)

[Pre-HQF gedrag](#)

[HQF-gedrag](#)

[Standaardgedrag van klasse](#)

[Pre-HQF gedrag](#)

[HQF-gedrag](#)

[Gerelateerde informatie](#)

## Inleiding

Dit document beschrijft de limieten van wachtrijen en de uitgangsdalingen op Cisco IOS®-softwareplatforms en oudere toegangsrouteurs.

## Voorwaarden

### Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- [Configuratiehandleiding voor Cisco IOS® Quality-of-Service oplossingen](#)
- [QoS—Hierarchical Queueing Framework \(HQF\)](#)

### Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende softwareversies:

- Voor Pre-HQF: Cisco-routers waarop Cisco IOS-software release 12.3(26) wordt uitgevoerd
- Voor HQF: Cisco-routers waarop Cisco IOS-software release 12.4(22)T wordt uitgevoerd

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.

## Conventies

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

## Achtergrondinformatie

Dit document is alleen van toepassing op Cisco IOS®-softwareplatforms, die doorgaans Cisco 7200VXR- en Cisco ISR 3800-, 2800-, 1800- en oudere Cisco Access Routers omvatten, waaronder 3700-, 3600-, 2600- en 1700-Series routers.

## Op klasse gebaseerde Weighted Fair Queueing Primer

In pre-HQF Cisco IOS-afbeeldingen kan elke klasse met een *bandwidth* De opdracht kan over het algemeen worden geprioriteerd tegen klassen zonder *bandbreedte* of *prioriteit* gebaseerd op het *Gewicht* van de klassen. Om het op klasse gebaseerde Weighted Fair Queueing (CBWFQ) planningsalgoritme te begrijpen, moet u eerst het concept van een gewicht begrijpen, dat flow-specifiek is voor Flow Based Fair Queues en Class specifiek voor individuele Class Based Queues binnen de Class Based Weighted Fair Queue.

De formule om het gewicht af te leiden voor een Flow Based Fair Queue is:

$$32384 / (\text{IP Prec} + 1)$$

Door gebruiker gedefinieerde klassen binnen een op klasse gebaseerde Weighted Fair-wachtrij leiden hun respectieve gewichten af als functie van de *bandwidth* opdracht die in de klasse is geconfigureerd als een deel van de SOM van alle *bandbreedte*-klassen in de Class Based Queue. De exacte formule is eigendom.

In HQF-afbeeldingen zijn op stroom gebaseerde Fair-Queues, configureerbaar in zowel door de gebruiker gedefinieerde klassen als klasse default met *Fair-Queue*, op dezelfde manier gepland (in plaats van op gewicht). Bovendien wordt in HQF de planningsprioriteit van een Class Based Queue bepaald op basis van de HQF-planner en niet op basis van de oude gewichtsformule van de klassen.

**Opmerking:** deze sectie is niet bedoeld als uitgebreide gedragsanalyse voor op klasse gebaseerde wachtrijbewerkingen. De intentie is een verklaring van hoe CBWFQ wachtrijlimieten en output dalingen toepast.

## De limieten van de wachtrij en de uitgangsdalingen begrijpen

## Door gebruiker gedefinieerde klassen geconfigureerd met de opdracht "prioriteit"

Voor MQC door gebruiker gedefinieerde klassen geconfigureerd met elke variant van de `priority` commando, met `priority` , `priority` , en `priority percent` inbegrepen.

### Pre-HQF gedrag

Technisch, alhoewel er geen CLI bestaat om het te zien, en het niet configureerbaar is, bestaat er een "verborgen" systeemwachtrij die wordt gedeeld door alle prioritaire klassegegevens. Dit fungeert als een tijdelijke responsabilisator voor prioritaire gegevens nadat deze zijn geclassificeerd en nadat het is toegestaan door de congestiebewuste politieagent. LLQ-pakketten worden in deze "verborgen" systeemwachtrij geplaatst als ze niet direct op de uitgaande interface-verzending kunnen worden geplaatst tijdens de ontvangstonderbreking, die anders functionele congestie wordt genoemd. In deze situatie, omdat er functionele congestie bestaat, wordt het pakket beoordeeld tegen de LLQ voorwaardelijke politieagent door de ontvangstonderbreking en terwijl het nog steeds eigendom is van de ontvangende interfacestuurprogramma. Als het pakket niet door de voorwaardelijke agent van LLQ wordt gelaten vallen, wordt het geplaatst in deze "verborgen" LLQ systeemrij en ontvang onderbreking wordt vrijgegeven. Daarom zijn alle pakketten die in deze "verborgen" systeemwachtrij zijn geplaatst in overeenstemming met de LLQ congestiebewuste policer en kunnen worden dewachted naar de uitgaande interface verzenden ring onmiddellijk door de LLQ/CBWFQ planner.

Ondanks het bestaan van deze wachtrij is het gedrag anders dan Cisco IOS-wachtrijen die zijn gemaakt voor niet-LLQ-gegevens (zoals Fair-Queue en Bandbreedtewachtrijen), in die zin dat er geen extra wachtrij (boven de latentie van de verzending) kan worden gemaakt omdat pakketten in deze wachtrij onmiddellijk naar de verzending kunnen worden gedraaid door LLQ/CBWFQ-planner. Als een prioriteitsklasse-pakket niet door de voorwaardelijke policer tijdens de ontvangstonderbreking wordt gelaten vallen, dan kan dit LLQ-pakket kort bestaan in deze "verborgen" systeemwachtrij voordat de dewachtrij naar de verzending van de uitgangsinterface loopt. In dit geval, kan de planner LLQ/CBWFQ het pakket aan de uitgaande interface verzenden ring onmiddellijk voorstellen. De Conditional Policer is uitgevoerd voordat het pakket wordt toegelaten tot de LLQ/CBWFQ, dus beperkt het de LLQ tot de ingestelde prioriteitspercentage.

Samenvattend, is het aanbevolen om LLQ-gegevens die de prioriteit overschrijden in tijden van congestie te laten vallen dan extra wachtrij latency, wat een fundamentele component van LLQ is. Dit mechanisme van voorwaardelijk toezicht staat een strikte prioriteitswachtrij toe en staat de prioriteitswachtrij niet toe om de gehele interface-PLIM te monopoliseren, wat een verbetering is ten opzichte van de functie voor prioriteitswachtrij van Cisco IOS.

- Wachtrijlimiet vóór HQF: NB
- Pre-HQF "prioriteit" + "willekeurig detecteren"-gedrag: NA, WRED niet toegestaan in LLQ.
- Pre-HQF "prioriteit" + "fair-Queueing" gedrag: NA, fair-Queueing niet toegestaan in LLQ.
- Pre-HQF "prioritair" + "willekeurig detecteren" + "eerlijk in wachtrij" gedrag: NA, niet eerlijk in wachtrij of willekeurig detecteren ondersteund in LLQ.

### HQF-gedrag

Hetzelfde als [Pre-HQF](#) behalve dat de verborgen wachtrij niet meer verborgen is en de wachtrijlimiet nu configureerbaar is en standaard 64 pakketten.

- HQF-wachtrijlimiet: 64 pakketten

- HQF "prioriteit" + "willekeurig detecteren"-gedrag: NA, WRED niet toegestaan in LLQ.
- HQF "prioriteit" + "eerlijke rij" gedrag: NA, eerlijke rij niet toegestaan in LLQ.
- HQF "prioritair" + "willekeurig detecteren" + "eerlijk in wachtrij" gedrag: NA, niet eerlijk in wachtrij of willekeurig detecteren ondersteund in LLQ.

## Door gebruiker gedefinieerde klassen geconfigureerd met de opdracht "bandbreedte"

Voor MQC door gebruiker gedefinieerde klassen geconfigureerd met elke variant van de `bandwidth` commando, met `bandwidth`, `bandwidth percent`, en `bandwidth remaining percent` inbegrepen.

### Pre-HQF gedrag

De standaard wachtrij-limiet is 64 pakketten, die instelbaar is. Als, tijdens de ontvangst onderbreking, u een pakket moet vragen dat in > 64 pakketten in de rij zou resulteren, is het pakket gelaten vallen staart.

- **Pre-HQF wachtrij-limiet:** 64 pakketten, instelbaar via wachtrij-limiet.
- **Pre-HQF "bandbreedte" + "willekeurig detecteren"-gedrag:**

Voorbeeld:

```
policy-map PRE_HQF_BANDWIDTH_WRED
class 1
  bandwidth 32
  random-detect
```

Als een variant van bandbreedte is geconfigureerd samen met een willekeurige-detectiemethode, neger dan elke wachtrij-limiet CLI, waardoor effectief elke bufferlimiet in de klasse wordt verwijderd. Met andere woorden, willekeurig detecteren en wachtrijlimiet sluiten elkaar uit in pre-HQF-afbeeldingen. met willekeurig detecteren als drop-strategie is de huidige wachtrijgrootte onbeperkt en kan theoretisch elke buffer bezetten die is toegewezen aan de op klasse gebaseerde fair-wachtrij, waar dit aantal buffers dat is toegewezen aan de op klasse gebaseerde fair-wachtrij wordt afgeleid op basis van het servicebeleid-aanhechtingspunt:

- Fysieke interface: 1000 pakketten, instelbaar met interface CLI wachtrij uit
- ATM PVC: 500 pakketten, instelbaar met PVC CLI `vc-hold-wachtrij`
- Frame Relay map-class: 600 pakketten, instelbaar met Frame Relay map-class CLI `frame Relay wachtrij`
- Op klasse gebaseerde vormgeving beleid toegepast op (sub)interface (pre-HQF): 1000 pakketten, instelbaar met MQC klasse CLI vorm `max-buffers`.

**Opmerking:** bij alle op Frame Relay en Class Based Shaping voorbeelden wordt ervan uitgegaan dat de som van de shapers de interfacekloksnelheid niet overschrijdt.

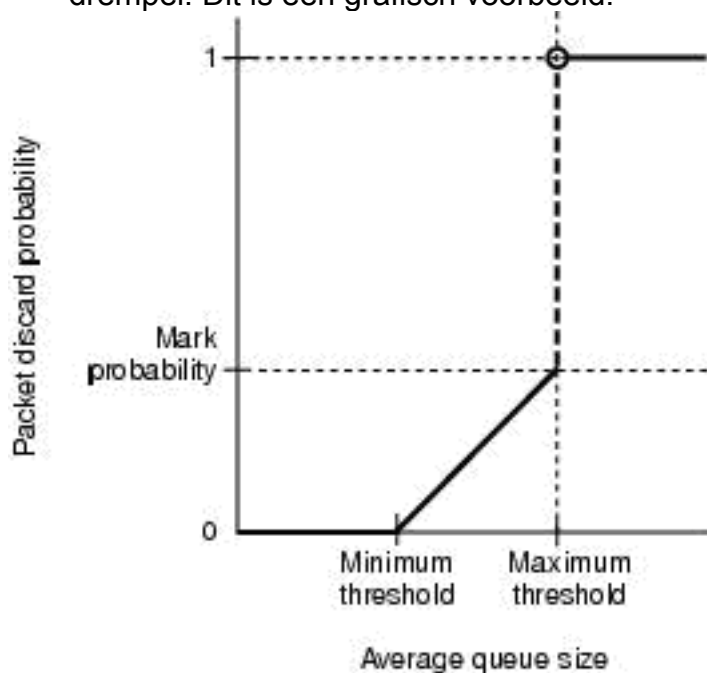
**Opmerking:** In pre-HQF-afbeeldingen, wanneer een Class Based Shaping-beleid wordt toegepast op een (sub)interface, let op de snelheid van de onderliggende fysieke interface, aangezien interfaces <2 Mbps standaard kunnen worden ingesteld op een Weighted Fair Queue en interfaces >2 Mbps kunnen standaard naar FIFO. In pre-HQF, kan de het vormen rij de rij van de interfaceruimte voeden, of het het vormen beleid op de subinterface of het fysieke interfaceniveau in bijlage is.

Tijdens de ontvangstonderbreking, telkens als een pakket een kandidaat voor een rij van de interfaceoutput wordt, wordt de gemiddelde grootte van de rij WRED berekend met deze formule:

$$\text{Average Queue Size} = (\text{old\_average} * (1 - 1/2^n)) + (\text{current\_queue\_size} * 1/2^n)$$

Als de resulterende gemiddelde grootte van de wachtrij:

- Kleiner dan de WRED min-drempel, vraagt het pakket en geeft het ontvang onderbreking vrij.
- Tussen de WRED min-drempel en de WRED max-drempel laat u mogelijk het pakket vallen met een grotere waarschijnlijkheid als de Gemiddelde Queue Size dichterbij de WRED max-drempel komt. Als de grootte van de gemiddelde wachtrij precies gelijk is aan de WRED max-drempel, wordt het pakket op basis van de noemer van de markeringswaarschijnlijkheid laten vallen. De noemer van de markeringswaarschijnlijkheid dient ook als basislijn om te bepalen welk percentage pakketten kan worden laten vallen wanneer de Gemiddelde grens van de Wachtrij niet precies gelijk aan WRED maximum-drempel is maar hoger is dan WRED min-drempel. Dit is een grafisch voorbeeld:



10.7.7.8

Gemiddelde wachtrij niet gelijk aan WRED Max-drempel

maar hoger dan WRED Min-drempel

- Als het pakket wordt gelaten vallen, ontvang onderbreking wordt vrijgegeven en een Willekeurige daling wordt verhoogd. Als het pakket niet wordt gelaten vallen, wordt het pakket onderzocht en ontvang onderbreking wordt vrijgegeven. Hoger dan de WRED max-drempel, laat het pakket vallen, laat het ontvang onderbreking vrij, en verhoog een Staartdaling.

**Opmerking:** op IP Precedence gebaseerde (standaard) en op DSCP gebaseerde WRED maken het mogelijk dat de min-drempel, max-drempel en mark-waarschijnlijkheidsnoemer voor verschillende waarden verschillend worden gedefinieerd. Dit is waar de gewogen component van Willekeurige Vroege Opsporing duidelijk is. U kunt bepaalde ToS-waarden ten opzichte van andere waarden beveiligen wanneer u hun relatieve drempels en waarschijnlijkheidsnoemers afstemt.

Wanneer de willekeurige detectie en bandbreedte samen zijn geconfigureerd, kan de huidige grootte van de wachtrij groter zijn dan de WRED max-drempel op een bepaald moment. Dit komt doordat de minimum en maximum drempels WRED alleen werken op basis van de gemiddelde (niet huidige) grootte van de wachtrij. Dit biedt de mogelijkheid om alle buffers te verlopen die zijn

toegewezen aan de op klasse gebaseerde wachtrij, wat zou kunnen leiden tot geen-bufferdalingen die overal binnen de op klasse gebaseerde eerlijke wachtrij plaatsvinden (zie Cisco bug-id [CSCsm94757](#)).

- **Pre-HQF "bandbreedte" + "eerlijke rij" gedrag:** NA, eerlijke rij niet toegestaan in bandbreedteklasse.
- **Pre-HQF "bandbreedte" + "willekeurig detecteren" + "eerlijk in wachtrij" gedrag:** NA, eerlijk in wachtrij niet toegestaan in bandbreedteklasse

## HQF-gedrag

Het gedrag is hetzelfde als beschreven in het gedeelte [Pre-HQF](#).

- **HQF-wachtrij-limiet:** 64 pakketten, instelbaar via wachtrij-limiet.

Dit is hetzelfde als in het pre-HQF.

- **HQF "bandbreedte" + "willekeurig detecteren"-gedrag:**

Voorbeeld:

```
policy-map HQF_BANDWIDTH_WRED
class 1
  bandwidth 32
  queue-limit 512
  random-detect
```

**Opmerking:** de standaardwachtrij-limiet is 64 pakketten.

Het gedrag is hetzelfde als in de equivalente pre-HQF sectie, met één belangrijke uitzondering. In HQF-afbeeldingen kunnen willekeurig detecteren en wachtrijlimieten naast elkaar bestaan in dezelfde door de gebruiker gedefinieerde klasse (of class class-default) en wachtrijlimieten kunnen worden ingeschakeld en ingesteld op 64 pakketten in een standaardconfiguratie. Als zodanig kan de wachtrij-limiet dienen als de maximale grootte van de huidige wachtrij in een klasse van willekeurige detectie, daarom kan deze een mechanisme bieden om geen bufferdruppels te beperken, zoals besproken in de equivalente sectie vóór HQF. Wegens deze toevoeging, moet de gevormde rij-grens minstens zo groot zijn zoals de willekeurig-ontdekt maximum-drempel, waar de willekeurig-ontdekt maximum-drempel aan 40 pakketten kan in gebreke blijven, of anders kan de syntactische parser de configuratie verwerpen.

Dit introduceert een huidige wachtrij-limiet controle in WRED-klassen, waarbij, zelfs als de berekening van de gemiddelde wachtrij-diepte kleiner is dan de max-drempel, als de huidige (niet gemiddelde) wachtrij-grootte groter is dan de wachtrij-limiet, het pakket kan worden gedropt, de ontvang wordt onderbroken en een Tail Drop wordt opgenomen. Vergeet niet dat, als de wachtrij-limiet hoog genoeg is ingesteld om de totale wachtrij-buffers voor de op klasse gebaseerde wachtrij uit te putten, er nog steeds geen bufferdruppels kunnen optreden. Geaggregeerde wachtbuffers voor HQF worden hier gedefinieerd:

- Fysieke interface: 1000 pakketten, instelbaar met interface CLI wachtrij uit
- ATM PVC: 500 pakketten, instelbaar met PVC CLI vc-hold-wachtrij
- Frame Relay map-class: 600 pakketten, instelbaar met Frame Relay map-class CLI frame Relay wachtrij

- Op klasse gebaseerde vormgevingsbeleid toegepast op fysieke interface in HQF-code: 1000 pakketten, instelbaar met een combinatie van interface CLI hold-wachtrij uit en wachtrij-limiet voor kinderbeleid waar wachtrij-limiet voor kinderbeleid een bovengrens heeft van wachtrij-wachtrij voor interface.
- Op klasse gebaseerde vormgeving beleid toegepast op subinterface in HQF code: 512 pakketten, niet afstembaar (onderzoek met NSSTG QoS platform Team, moet het afstembaar zijn)

**Opmerking:** bij alle op Frame Relay en Class Based Shaping voorbeelden wordt ervan uitgegaan dat de som van de shapers de interfacekloksnelheid niet overschrijdt.

Hier is een voorbeeld uit de echte wereld:

```
policy-map JACKLYN
class 1
  bandwidth 64
  queue-limit 500 packets
  random-detect
  random-detect precedence 1 22 300
```

Tijdens deze output, wordt geen verkeer geproduceerd door de interface:

```
F340.11.25-7200-5_LAC#show policy-map interface | i queue
queue limit 500 packets
(queue depth/total drops/no-buffer drops) 0/387595/0
```

```
!--- Current_q_depth is 0 Mean queue depth: 107 packets !--- last calculation of
Average_queue_depth
```

Op dit punt is het verkeer gestart. De stroom is niet-bursty bij 400PPS en bestaat uit 1000 bytesframes:

```
F340.11.25-7200-5_LAC#show policy-map interface | i queue
queue limit 500 packets
(queue depth/total drops/no-buffer drops) 461/387641/0
```

```
!--- 461 is Current_q_depth > Prec 1 max-thresh of 300 !--- but < "queue-limit 500 packets".
Mean queue depth: 274 packets !--- Avg_q_depth is rising, Mark Prob Denom is being used.
F340.11.25-7200-5_LAC#show policy-map interface | i queue queue limit 500 packets (queue
depth/total drops/no-buffer drops) 363/387919/0 !--- 363 is Current_q_depth and it is falling
compared to last !--- iteration because WRED is random dropping packets. Mean queue depth: 351
packets !--- Avg_q_depth is now above max_thresh, WRED starts to tail-drop !--- in addition to
random-drop. F340.11.25-7200-5_LAC#show policy-map interface | i queue queue limit 500 packets
(queue depth/total drops/no-buffer drops) 199/388263/0 Mean queue depth: 312 packets F340.11.25-
7200-5_LAC#show policy-map interface | i queue queue limit 500 packets (queue depth/total
drops/no-buffer drops) 303/388339/0 Mean queue depth: 276 packets F340.11.25-7200-5_LAC#show
policy-map interface | i queue queue limit 500 packets (queue depth/total drops/no-buffer drops)
325/388498/0 Mean queue depth: 314 packets F340.11.25-7200-5_LAC#show policy-map interface | i
queue queue limit 500 packets (queue depth/total drops/no-buffer drops) 298/390075/0 Mean queue
depth: 300 packets
```

Merk op hoe, uiteindelijk, met een niet-bursty stroom, de WRED gemiddelde wachtrijdiepte de Huidige Diepte van de Wachtrij kan evenaren, die het verwachte gedrag is.

- HQF "bandbreedte" + "eerlijke wachtrij"-gedrag:

Wanneer bandbreedte en fair-wachtrij samen worden toegepast op een door HQF-gebruiker

gedefinieerde klasse, wordt aan elke op stroom gebaseerde wachtrij een wachtrij-limiet toegewezen gelijk aan  $.25 * \text{wachtrij-limiet}$ . Omdat de standaard wachtrij-limiet 64 pakketten is, kan elke op stroom gebaseerde wachtrij in een fair-wachtrij 16 pakketten toegewezen krijgen. Als vier stromen deze klasse doorkruisen, zou elke flow-wachtrij standaard 16 pakketten hebben, daarom zou je nooit verwachten om totale pakketten te zien die van  $>64$  ( $4 * 16$ ) worden gevraagd. Alle staatdruppels uit een individuele flowwachtrij worden geregistreerd als flowdruppels. Als het aantal flow-wachtrijen aanzienlijk hoog was, zoals de wachtrij-limiet was, dan is er nog een kans voor no-buffer dalingen. Bijvoorbeeld, als u veronderstelt het beleid band-punt een fysieke interface is, waar 1000 gezamenlijke buffers worden toegewezen:

```
policy-map TEST
class 1
  bandwidth 32
  fair-queue 1024
  queue-limit 128
```

In deze configuratie kan aanzienlijk verkeer in alle stroomwachtrijen samengevoegde interfacebuffers ontmoedigen en in geen-bufferdalingen in andere door de gebruiker gedefinieerde klassen resulteren (zie Cisco bug-id [CSCsw98427](#)). Dit komt doordat 1024-flowwachtrijen, elk met een 32-pakketwachtrij-limiet, gemakkelijk de 1000 geaggregeerde interface Class Based Queuing-buffertoewijzing kunnen overtekenen.

- **HQF "bandbreedte" + "willekeurige detectie" + "eerlijke rij"-gedrag:**

**Voorbeeld:**

```
policy-map TEST
class 1
  bandwidth 32
  fair-queue 1024
  queue-limit 128
  random-detect
```

Hetzelfde als bandbreedte en fair-wachtrij in sectie behalve WRED Gemiddelde Queue Grootte wordt berekend telkens als een pakket aankomt om te beslissen of het pakket willekeurig moet worden gedropt of staart moet worden gedropt. Zoals met pre-HQF, kunnen alle stroom-rijen één geval van WRED drempels delen, wat betekent dat alle die pakketten aan alle stroom-rijen worden gevraagd worden gebruikt om WRED gemiddelde wachtrijdiepte te berekenen, dan past het dalingsbesluit de minimum WRED en maximumdrempels tegen de gezamenlijke pakketten in alle stroomrijen toe. Echter, een andere afwijking van bandbreedte en fair-wachtrij in sectie, omdat één geval van WRED drempels van toepassing is op alle op stroom gebaseerde wachtrijen, wordt de individuele stroom-wachtrij' wachtrij-limiet ( $.25 * \text{"wachtrij-limiet"}$ ) genegeerd en eer in plaats daarvan de Klassen geaggregeerde wachtrij-limiet voor een controle van de Huidige Wachtrij.

## Standaardgedrag van klasse

### Pre-HQF gedrag

In pre-HQF, de Standaard van de Klasse aan eerlijk-rij, delen alle stroom-rijen de rij-grens voor de klasse (het gebrek is 64), en er is geen bandbreedtereservering. Met andere woorden, het totale aantal onderzochte pakketten in alle flow-wachtrijen kan nooit de wachtrij-limiet overschrijden. De hoeveelheid pakketten die in elke stroom-rij wordt gevraagd kan afhankelijk zijn van het berekende Gewicht van de stroom-rij. Omgekeerd, als de eerlijk-rij en willekeurig-ontdekken samen in klasse-gebrek worden gebruikt, kan de rij-grens worden genegeerd en alle stroom-rijen



kunnen de zelfde drempels delen WRED. Als dusdanig kunnen alle pakketten die momenteel in alle flow-wachtrijen worden opgezocht, worden gebruikt om de WRED Average Queue Size te berekenen. Omdat de huidige grootte van de wachtrij in deze configuratie geen bovengrens heeft, is de kans op geen-bufferdalingen groot (raadpleeg Cisco bug-id [CSCsm94757](#)).

- Als bandbreedte wordt toegevoegd aan class-default, kan dit resulteren in gedrag dat wordt beschreven in de sectie [Pre-HQF Behavior - User Defined Classes Configureerd met de sectie "bandbreedte"-opdracht](#).
- Als bandbreedte en willekeurig-detecteren worden toegevoegd aan klasse, kan de standaardinstelling resulteren in gedrag dat wordt beschreven in de sectie *Pre-HQF "bandbreedte" + "willekeurig detecteren" gedrag* van [Pre-HQF Behavior - User Defined Classes Configureerd met de "bandbreedte"-opdracht](#).

**Opmerking:**In pre-HQF kan bandbreedte niet naast Fair-Queue bestaan in class-default.

## HQF-gedrag

In HQF is Class Default standaard in een FIFO-wachtrij en krijgt het een pseudo-bandbreedtereservering toegewezen op basis van de resterende toewijzingen uit door gebruiker gedefinieerde klassen. Zie voor standaardgedrag op basis van HQF-klasse de sectie [HQF Behavior - User Defined Classes Configureerd met de sectie "bandbreedte"-opdracht](#). Klasse-default in HQF-afbeeldingen kan te allen tijde, ongeacht de configuratie, een impliciete bandbreedtereservering hebben die gelijk is aan de ongebruikte interfacebandbreedte die niet wordt verbruikt door door de gebruiker gedefinieerde klassen. Standaard ontvangt de class-default klasse een minimum van 1% van de interface of parent shape bandbreedte. Het is ook mogelijk om de bandbreedte CLI in klassengebrek uitdrukkelijk te vormen.

- Als fair-Queue is geconfigureerd in klasse-Default, komt het gedrag overeen met de sectie *HQF "bandbreedte" + "fair-Queue" gedrag* van [HQF Behavior - User Defined Classes geconfigureerd met de "bandbreedte"-opdracht](#).
- Als fair-Queue en random-discovery samen zijn geconfigureerd in klasse-Default, komt het gedrag overeen met *de sectie HQF "bandbreedte" + "willekeurig-detecteren" + "eerlijk-Queue" gedrag* van [HQF Behavior - User Defined Classes Configureerd met de "bandbreedte"-opdracht](#).

## Gerelateerde informatie

- [Configuratiehandleiding voor Cisco IOS Quality-of-Service oplossingen, release 12.2SR](#)
- [QoS—Hierarchical Queueing Framework \(HQF\) \[niet ondersteund\]](#)
- [Ondersteuning van routers](#)
- [Cisco technische ondersteuning 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.