

瞭解IOS軟體平台上的佇列限制和輸出捨棄專案

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[類別型加權公平排隊引物](#)

[瞭解佇列限制和輸出丟棄](#)

[使用Priority命令配置的使用者定義類](#)

[HQF前行為](#)

[HQF行為](#)

[使用Bandwidth命令配置的使用者定義類](#)

[HQF前行為](#)

[HQF行為](#)

[類預設行為](#)

[HQF前行為](#)

[HQF行為](#)

[相關資訊](#)

簡介

本檔案介紹Cisco IOS®軟體平台和舊版存取路由器上的佇列限制和輸出捨棄專案。

必要條件

需求

思科建議您瞭解以下主題：

- [Cisco IOS®服務品質解決方案配置指南](#)
- [QoS — 分層排隊框架\(HQF\)](#)

採用元件

本檔案中的資訊是根據以下軟體版本：

- 對於Pre-HQF：運行Cisco IOS軟體版本12.3(26)的Cisco路由器
- 對於HQF：運行Cisco IOS軟體版本12.4(22)T的Cisco路由器

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

慣例

如需文件慣例的詳細資訊，請參閱思科技術提示慣例。

背景資訊

本文檔僅適用於思科IOS®軟體平台，通常包括Cisco 7200VXR和Cisco ISR 3800、2800、1800系列路由器，以及包括3700、3600、2600和1700系列路由器的舊版思科接入路由器。

類別型加權公平排隊引物

在HQF之前的Cisco IOS映像中，任何包含命令的類 bandwidth 通常可以基於類的權重按照沒有頻寬或優先順序的類進行優先排序。為了瞭解基於類的加權公平隊列(CBWFQ)排程演算法，您必須首先瞭解權重的概念，該概念特定於基於流的公平隊列，並且特定於基於類的加權公平隊列中的各個基於類的隊列。

推導基於流的公平隊列的權重的公式為：

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

基於類的加權公平隊列中的使用者定義類根據配置在該類中的命令匯出其各自的權 bandwidth 重，該權重是作為基於類的隊列中所有頻寬類的SUM的比例配置的。確切的公式是專有公式。

在HQF映像中，基於流的公平隊列（可在使用者定義的類中配置）和使用fair-queue的類預設值中配置）被平均排程（而不是按Weight）。此外，在HQF中，基於類的隊列的排程優先順序是基於HQF排程器，而不是基於類的傳統權重公式確定的。



注意：本節並非針對基於類的排隊操作進行全面的行為分析。意圖是解釋CBWFQ如何應用隊列限制和輸出丟棄。

瞭解隊列限制和輸出丟棄

使用Priority命令配置的使用者定義類

對於使用命令的任何變體配置的MQC使用者定義 priority 類，包含 priority、priority <kbps>和 priority percent。

HQF前行為

從技術上講，即使沒有用於檢視CLI的CLI，而且該CLI不可配置，但存在一個由所有優先順序類資料共用的隱藏系統隊列。在優先順序資料被分類之後，在擁塞感知策略器允許之後，它充當優先順序資料的臨時儲存庫。如果在接收中斷期間無法將LLQ資料包直接置於輸出介面傳輸環上，則將其置於此隱藏系統隊列中，否則稱為功能擁塞。在這種情況下，由於存在功能擁塞，在接收中斷期間和資

料包仍然由接收介面驅動程式所擁有時，將根據LLQ條件監察器對資料包進行評估。如果LLQ條件監察器沒有丟棄該資料包，則將其放置在此隱藏的LLQ系統隊列中，並釋放接收中斷。因此，位於此隱藏系統隊列中的所有資料包都已符合LLQ擁塞感知管制器，並且可以由LLQ/CBWFQ排程式立即將資料包出隊到出口介面傳輸環中。

儘管存在此隊列，但此行為與為非LLQ資料（例如公平隊列和頻寬隊列）建立的Cisco IOS隊列不同，因為此隊列中的資料包可以立即由LLQ/CBWFQ排程式排入傳輸環，所以不會產生額外的排隊延遲（高於傳輸環延遲）。如果在接收中斷期間，優先順序類資料包未被條件監察器丟棄，則此LLQ資料包可能短暫地存在於此隱藏系統隊列中，然後被出隊到輸出介面的傳輸環中。在這種情況下，LLQ/CBWFQ排程式可以立即將資料包呈現給輸出介面傳輸環。條件管制器在其將封包允許到LLQ/CBWFQ之前已執行，因此，它將LLQ限制為已設定的優先順序速率。

總之，建議在擁塞時丟棄超過優先順序速率的LLQ資料，而不是產生額外的排隊等待時間，而排隊等待時間是LLQ的基本組成部分。此條件式管制機制允許嚴格的優先順序佇列，且不允許優先順序佇列獨佔整個介面PLIM，這是對Cisco IOS舊版優先順序佇列功能的改進。

-

Pre-HQF隊列限制：不適用

-

Pre-HQF priority + random-detect behavior: NA，WRED在LLQ中不允許使用。

-

Pre-HQF priority + fair-queue"行為：不適用，LLQ中不允許公平隊列。

-

Pre-HQF priority + random-detect + fair-queue behavior: NA，在LLQ中不支援公平隊列或隨機檢測。

HQF行為

與[Pre-HQF](#)相同，但隱藏隊列不再隱藏，隊列限制現在可配置且預設為64個資料包。

-

HQF隊列限制：64個資料包

-

HQF優先順序+隨機檢測行為：NA、WRED在LLQ中不允許使用。

-

HQF優先順序+公平隊列行為：不適用，LLQ中不允許公平隊列。

-

HQF優先順序+隨機檢測+公平隊列行為：NA，LLQ不支援公平隊列或隨機檢測。

使用Bandwidth命令配置的使用者定義類

適用於使用命令的任何變體配置的MQC使用者定義 bandwidth 類，包括 bandwidth <kbps>、 bandwidth percent 和 bandwidth remaining percent命令。

HQF前行為

預設隊列限制為64個資料包，這是可調的。如果在接收中斷期間，您需要將可能導致隊列中超過64個資料包的資料包入隊，則該資料包將被尾部丟棄。

-

Pre-HQF queue-limit:64個資料包，可通過隊列限制進行調整。

- Pre-HQF頻寬+「random-detect」行為：

範例：

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

如果頻寬的任何變體與隨機檢測的任何變體一起配置，請忽略任何隊列限制CLI，這樣可以有效地刪除類中的任何緩衝區限制。也就是說，在pre-HQF景象中，隨機檢測和隊列限制是互斥的，使用隨機檢測作為丟棄策略，當前隊列大小不受約束，理論上可以佔用分配給基於類的公平隊列的每個緩衝區，其中分配給基於類的公平隊列的緩衝區的數量是基於服務策略連線點得出的：

-

物理介面：1000個資料包，可通過介面CLI保持隊列退出進行調節

•

ATM PVC:500個資料包，使用PVC CLI `vc-hold-queue`可調

•

Frame-Relay map-class:600 packets，可使用frame-relay map-class CLI `frame-relay hold-queue`進行調整

•

應用於(子)介面(HQF之前)的基於類的整形策略：1000個資料包，可使用MQC類CLI `shape max-buffers`進行調整。



注意：所有幀中繼和基於類的整形示例都假設整形器的總和不超過介面時鐘速率。



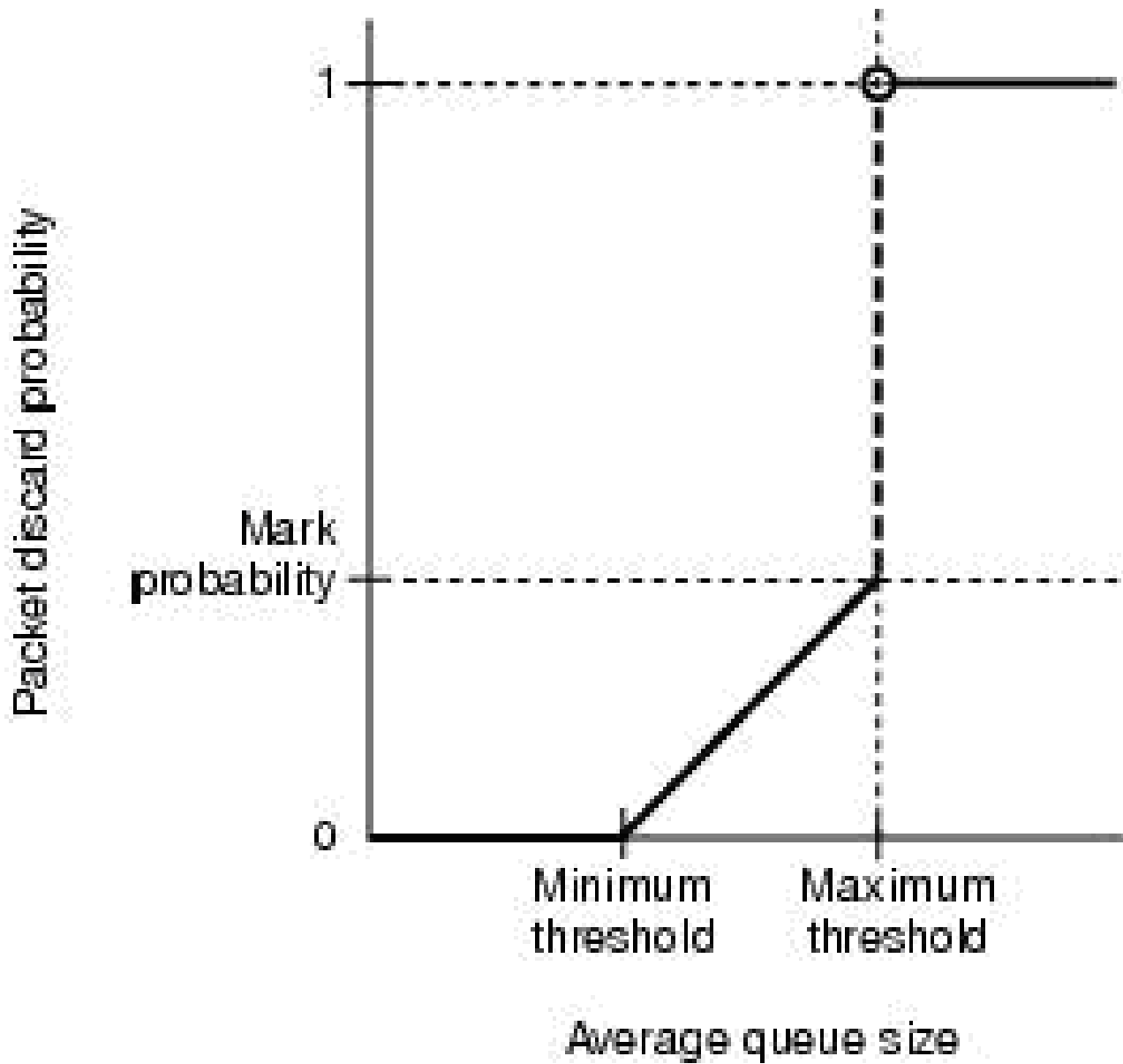
注意：在HQF之前的映像中，將基於類的整形策略應用到(子)介面時，請注意底層物理介面的速度，因為介面<2Mbps可以預設為加權公平隊列，而介面>2Mbps可以預設為FIFO。在pre-HQF中，無論整形策略是在子介面級別還是在物理介面級別連線，整形隊列都可以饋送介面保持隊列。

在接收中斷期間，每次資料包成為介面輸出隊列的候選時，WRED平均隊列大小使用以下公式計算：

$$\text{Average Queue Size} = (\text{old_average} * (1 - 1/2^n)) + (\text{current_queue_size} * 1/2^n)$$

如果生成的平均隊列大小為：


- 小於WRED最小閾值，將資料包入隊並釋放接收中斷。
- 在WRED min-threshold和WRED max-threshold之間，當平均隊列大小接近WRED max-threshold時，可能會丟棄概率增加的資料包。如果平均隊列大小完全等於WRED最大閾值，則根據標籤概率分母丟棄資料包。標籤概率分母還用作基線，以確定當平均隊列限制不完全等於WRED最大閾值但大於WRED最小閾值時，可以丟棄的資料包百分比。以下是一個圖形範例：



平均隊列限制不等於WRED最大閾值但高於WRED最小閾值

如果資料包被丟棄，則釋放接收中斷，並遞增隨機丟棄。如果資料包未被丟棄，則資料包將入隊並釋放接收中斷。

大於WRED最大閾值，丟棄資料包，釋放接收中斷，增加尾部丟棄。

 註：基於IP優先順序（預設）和基於DSCP的WRED允許針對不同值以不同方式定義最小閾值、最大閾值和標籤概率分母。這就是隨機早期檢測的加權分量的明顯之處。在調整其相對閾值和標籤概率分母時，可以保護某些ToS值相對於其他值。

同時配置隨機檢測和頻寬時，在任何給定時間點上，當前隊列大小都可以大於WRED最大閾值。這是因為WRED最小和最大閾值僅基於平均（非當前）隊列大小而運行。這提供了一個機會使分配給基於類的隊列的所有緩衝區失效，這些緩衝區可能導致基於類的公平隊列中的任何位置發生無緩衝區丟棄(請參閱思科錯誤ID [CSCsm94757](#))。

- Pre-HQF頻寬+公平隊列行為：NA，頻寬類中不允許公平隊列。

- Pre-HQF頻寬+隨機檢測+公平隊列行為：NA，頻寬類中不允許公平隊列

HQF行為

此行為與[Pre-HQF](#)一節中描述的行為相同。

- HQF queue-limit:64個資料包，可通過queue-limit進行調整。

這與HQF之前的情況相同。

- HQF頻寬+隨機檢測行為：

範例：

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

 註：預設隊列限制為64個資料包。

行為與等效的pre-HQF部分相同，只有一個重要例外。在HQF映像中，隨機檢測和隊列限制可以共存於同一個使用者定義的類（或類class-default）中，隊列限制可以啟用並調整為預設配置中的64個資料包。因此，隊列限制可以充當隨機檢測類中的最大當前隊列大小，因此，它可以提供一種限制在等效pre-HQF部分中討論的無緩衝區丟棄的機制。由於此新增，所配置的queue-limit必須至少與random-detect max-threshold一樣大，其中random-detect max-threshold可以預設為40個資料包，否則分析器可以拒絕配置。

這在WRED類中引入了當前隊列限制檢查，由此即使平均隊列深度計算小於最大閾值，如果當前（非平均）隊列大小大於隊列限制，資料包也可以被丟棄，接收中斷被釋放，並且記錄尾部丟棄。請記住，如果隊列限制調整得足夠高，以耗盡基於類的隊列的聚合隊列緩衝區，則仍有可能發生無緩衝區丟棄。HQF的聚合隊列緩衝區定義如下：

-

物理介面：1000個資料包，可通過介面CLI hold-queue out進行調整。

-

ATM PVC:500個資料包，使用PVC CLI vc-hold-queue可調。

-

Frame-Relay map-class:600個資料包，可使用frame-relay map-class CLI frame-relay hold-queue進行調整。

-

應用於HQF代碼中物理介面的基於類的整形策略：1000個資料包，可通過介面CLI保留隊列退出和子策略隊列限制的組合進行調整，其中子策略隊列限制具有介面保留隊列退出上限。

-

應用於HQF代碼中子介面的基於類的整形策略：512個資料包，不可調（使用NSSTG QoS平台團隊進行調查，必須對其進行調）。



注意：所有幀中繼和基於類的整形示例都假設整形器的總和不超過介面時鐘速率。

下面是一個真實世界的示例：

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


在此輸出期間，不會透過介面產生流量：

```
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

此時，流量已啟動。資料流在400PPS時無突發，由1000位元組幀組成：

```
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

請注意，對於非突發流，WRED平均隊列深度最終如何等於當前隊列深度，這是預期行為。

-

HQF頻寬+公平隊列行為：

當頻寬和公平隊列一起應用於HQF使用者定義的類時，每個基於流的隊列分配一個等於。25 *隊列限制的隊列限制。由於預設隊列限制為64個資料包，因此公平隊列中的每個基於流的隊列可以分配16個資料包。如果四個流經過此類，則預設情況下，每個流隊列將有16個資料包，因此您永遠也不會看到排隊的資料包總數超過64(4 *16)。來自單個流隊列的所有尾部丟棄都記錄為流丟棄。如果流隊列數顯著高於隊列限制，則無緩衝區丟棄的另一個機會。例如，如果您假設策略attach-point是一個物理介面，其中分配了1000個聚合緩衝區：

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

在此組態中，所有流量佇列中的可觀流量可能會耗盡聚合介面緩衝區，並導致其他使用者定義類別中的無緩衝區捨棄專案(請參閱思科錯誤ID [CSCsw98427](#))。這是因為1024個流隊列(每個隊列具有32個資料包隊列限制)可以輕鬆地超訂用1000聚合介面基於類的隊列緩衝區分配。

-

HQF頻寬+隨機檢測+公平隊列行為：

範例：

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

與section中的頻寬和公平隊列相同，不同之處在於，每當資料包到達時，都會計算WRED平均隊列大小，以確定資料包必須隨機丟棄還是尾部丟棄。與pre-HQF一樣，所有流隊列可以共用一個WRED閾值例項，這意味著所有流隊列中排隊的所有資料包都用於計算WRED平均隊列深度，然後丟棄決策針對所有流隊列中的聚合資料包應用WRED最小和最大閾值。但是，另一個偏離節中的頻寬和公平隊列，因為一個WRED閾值例項應用於所有基於流的隊列，所以忽略單個流隊列的隊列限制(.25 * "queue-limit")，而接受當前隊列限制檢查的類聚合隊列限制。

類預設行為

HQF前行為

在pre-HQF中，Class Default預設為fair-queue，所有流隊列共用類的隊列限制（預設為64），並且沒有頻寬保留。換句話說，在所有流隊列中排隊的資料包總數永遠不能超過隊列限制。每個流隊列中排隊的資料包數量可能取決於計算出的流隊列的權重。反之，如果在類預設值中同時使用fair-queue和random-detect，則可以忽略隊列限制，並且所有流隊列可以共用相同的WRED閾值。因此，當前在所有流隊列中排隊的所有資料包都可用於計算WRED平均隊列大小。由於此組態中的目前佇列大小沒有上限，因此無緩衝區捨棄的機會相當高(請參閱思科錯誤ID [CSCsm94757](#))。

-

如果將頻寬新增到類預設值，則可能會導致在[Pre-HQF Behavior - User Defined Classes Configured with the "bandwidth" Command](#)部分中概述的行為。

-

如果將bandwidth和random-detect新增到class class-default中，則可能導致在[Pre-HQF Behavior - User Defined Classes Configured with the "bandwidth"命令](#)的Pre-HQF bandwidth + random-detect behavior部分中概述的行為。



注意：在pre-HQF中，頻寬不能與class-default中的fair-queue共存。

HQF行為

在HQF中，「類預設值」預設為FIFO隊列，並根據使用者定義的類的剩餘分配分配偽頻寬保留。同樣地，有關HQF類的預設行為，請參見[HQF Behavior - User Defined Classes Configured with the "bandwidth"命令](#)部分。在任何時候，無論配置如何，HQF映像中的class class class-default始終可以有一個隱式頻寬保留，該保留等於使用者定義的類未使用的介面頻寬。預設情況下，class-default類至少收到介面或父形狀頻寬的1%。也可以以類預設值顯式配置頻寬CLI。

-

如果在class Class-Default中配置了fair-queue，則行為與[HQF Behavior - User Defined Classes Configured with the "bandwidth"命令](#)的HQF bandwidth + fair-queue behavior部分匹配。

-

如果在Class-Default中同時配置了fair-queue和random-detect，則行為與[HQF Behavior - User Defined Classes Configured with the](#)

["bandwidth"命令的HQF bandwidth + random-detect + fair-queue behavior部分匹配。](#)

相關資訊

- [Cisco IOS服務品質解決方案組態設定指南12.2SR版](#)
- [QoS — 分層排隊框架\(HQF\)\[不支援\]](#)
- [路由器支援](#)
- [思科技術支援與下載](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。