

# Programación de salida de QoS en switches Catalyst 6500/6000 Series que ejecutan Cisco IOS System Software

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

[Perdidas de la cola de salida](#)

[Capacidad de almacenamiento en cola de salida de diferentes tarjetas de línea en Catalyst 6500/6000](#)

[Comprensión de la Capacidad de Cola de un Puerto](#)

[Configuración, Monitor y Ejemplo de Programación de Resultados en Catalyst 6500/6000](#)

[Configuración](#)

[Supervisión de la programación de resultados y verificación de configuraciones](#)

[Ejemplo de programación de salida](#)

[Utilice la programación de salida para reducir la demora y la fluctuación](#)

[Reducir el retraso](#)

[Reducción de la fluctuación](#)

[Conclusión](#)

[Información Relacionada](#)

## [Introducción](#)

El uso de la programación de salida asegura que el tráfico importante no se interrumpa en caso de que se produzca una suscripción excesiva. Este documento trata todas las técnicas y los algoritmos involucrados en la programación de salida en el switch Catalyst 6500/6000. Este documento también explica cómo configurar y verificar el funcionamiento de la programación de salida en el Catalyst 6500/6000 que ejecuta Cisco IOS® Software.

Consulte [Programación de Salida de QoS en Switches Catalyst 6500/6000 Series que Ejecutan el Software del Sistema CatOS](#) para obtener más información sobre Detección temprana aleatoria ponderada (WRED), Ordenamiento cíclico ponderado (WRR) y caída de cola.

## [Prerequisites](#)

## [Requirements](#)

No hay requisitos específicos para este documento.

## Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

## Convenciones

Consulte [Convenciones de Consejos Técnicos Cisco para obtener más información sobre las convenciones del documento.](#)

## Antecedentes

### Perdidas de la cola de salida

Las caídas de salida son causadas por una interfaz congestionada. Una causa común de esto podría ser el tráfico de un link de ancho de banda alto que se conmuta a un link de ancho de banda inferior, o el tráfico de varios links entrantes que se conmuta a un único link de salida.

Por ejemplo, si una gran cantidad de tráfico saturado entra en una interfaz gigabit y se conmuta hacia una interfaz de 100Mbps, esto podría ocasionar que aumenten las pérdidas en la salida en la interfaz de 100Mbps. Esto ocurre porque la cola de salida en esa interfaz está saturada por el exceso de tráfico debido a la asimetría de la velocidad entre los anchos de banda entrante y saliente. La velocidad de tráfico en la interfaz saliente no puede aceptar todos los paquetes que se deben enviar.

Para resolver el problema, la mejor solución es aumentar la velocidad de línea. Sin embargo, hay maneras de prevenir, disminuir o controlar las caídas de salida cuando no desea aumentar la velocidad de línea. Sólo puede evitar caídas de salida si las caídas de salida son consecuencia de ráfagas cortas de datos. Si las caídas de salida son causadas por un flujo constante de alta velocidad, no puede evitar las caídas. Sin embargo, puede controlarlos.

## Capacidad de almacenamiento en cola de salida de diferentes tarjetas de línea en Catalyst 6500/6000

Si no está seguro de la capacidad de colocación en cola de un puerto, ejecute el comando **show queueing interface {gigabitethernet | fastethernet} mod/port**. Aquí se muestran las primeras líneas de salida de un comando **show queueing interface**. El puerto está en una tarjeta de línea Supervisor Engine 1A:

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin

QoS is disabled globally
Trust state: trust DSCP
Default COS is 0
Transmit group-buffers feature is enabled
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
```

```

-----
 1          WRR low          2
 2          WRR high        2
 3          Priority         1

```

!--- Output suppressed.

El resultado muestra que este puerto tiene un tipo de colocación en cola de salida conocido como 1p2q2t.

Otra manera de ver el tipo de colocación en cola disponible en un puerto específico es ejecutar el comando **show interface capabilities**:

```

la-orion#show interface gigabitethernet 6/2 capabilities
GigabitEthernet6/2
 Model:                WS-SUP720-BASE
 Type:                 No GBIC
 Speed:                1000
 Duplex:               full
 Trunk encap. type:    802.1Q,ISL
 Trunk mode:           on,off,desirable,nonegotiate
 Channel:              yes
 Broadcast suppression: percentage(0-100)
 Flowcontrol:          rx-(off,on,desired),tx-(off,on,desired)
 Membership:           static
 Fast Start:           yes
 QoS scheduling:    rx-(1p1q4t), tx-(1p2q2t)
 CoS rewrite:          yes
 ToS rewrite:          yes
 Inline power:         no
 SPAN:                 source/destination
 UDL D:                yes
 Link Debounce:        yes
 Link Debounce Time:   yes
 Ports on ASIC:        1-2

```

## Comprensión de la Capacidad de Cola de un Puerto

Hay varios tipos de colas disponibles en los switches Catalyst 6500/6000. Esta tabla explica la notación de la arquitectura de QoS del puerto:

Transmisión (Tx)/Recepción (Rx)	Notación de cola	Nº de colas	cola prioritaria	Nº de colas WRR	Nº y tipo de umbral para colas WRR
Tr	2q2t	2	—	2	2 caídas de cola configurables
Tr	1p2q2t	3	1	2	2 WRED configurables
Tr	1p3q1t	4	1	3	1 WRED configurable
Tr	1p2q1t	3	1	2	1 WRED configurable
Rx	1q4t	1	—	—	4 caídas de cola configurables
Rx	1p1q4t	2	1	1	4 caídas de cola configurables

Rx	1p1q0t	2	1	1	No configurable
Rx	1p1q8t	2	1	1	8 WRED configurables
Tr	1p3q8t	4	1	3	8 WRED configurables o eliminación de cola
Tr	1p7q8t	8	1	7	8 WRED configurables o eliminación de cola
Rx	1q2t	1	—	—	1 descarte de cola configurable = 1 no configurable
Rx	1q8t	1	—	—	8 caídas de cola configurables
Rx	2q8t	2	—	2	8 caídas de cola configurables

La siguiente tabla enumera algunos de los módulos y tipos de cola en los lados Rx y Tx de la interfaz o puerto. Si su módulo no aparece aquí, utilice el comando **show interface capabilities** para determinar la capacidad de cola disponible. El comando **show interface capabilities** se describe en la [Capacidad de Colocación en Cola de Salida de Diferentes Tarjetas de Línea en la sección Catalyst 6500/6000](#).

Módulo	Colas Rx	Colas Tx
WS-X6K-S2-PFC2	1p1q4t	1p2q2t
WS-X6K-SUP1A-2GE	1p1q4t	1p2q2t
WS-X6K-SUP1-2GE	1q4t	2q2t
WS-X6501-10GEX4	1p1q8t	1p2q1t
WS-X6502-10GE	1p1q8t	1p2q1t
WS-X6516-GBIC	1p1q4t	1p2q2t
WS-X6516-GE-TX	1p1q4t	1p2q2t
WS-X6416-GBIC	1p1q4t	1p2q2t
WS-X6416-GE-MT	1p1q4t	1p2q2t
WS-X6316-GE-TX	1p1q4t	1p2q2t
WS-X6408A-GBIC	1p1q4t	1p2q2t
WS-X6408-GBIC	1q4t	2q2t
WS-X6524-100FX-MM	1p1q0t	1p3q1t
WS-X6324-100FX-SM	1q4t	2q2t
WS-X6324-100FX-MM	1q4t	2q2t
WS-X6224-100FX-MT	1q4t	2q2t
WS-X6548-RJ-21	1p1q0t	1p3q1t
WS-X6548-RJ-45	1p1q0t	1p3q1t

WS-X6348-RJ-21	1q4t	2q2t
WS-X6348-RJ21V	1q4t	2q2t
WS-X6348-RJ-45	1q4t	2q2t
WS-X6348-RJ-45V	1q4t	2q2t
WS-X6148-RJ-45V	1q4t	2q2t
WS-X6148-RJ21V	1q4t	2q2t
WS-X6248-RJ-45	1q4t	2q2t
WS-X6248A-TEL	1q4t	2q2t
WS-X6248-TEL	1q4t	2q2t
WS-X6024-10FL-MT	1q4t	2q2t

## [Configuración, Monitor y Ejemplo de Programación de Resultados en Catalyst 6500/6000](#)

### [Configuración](#)

Esta sección describe todos los pasos necesarios para configurar la programación de salida en un Catalyst 6500/6000 que ejecuta Cisco IOS Software. Para la configuración predeterminada de Catalyst 6500/6000, vea el [Caso 1: QoS está habilitado y se utiliza un parámetro predeterminado](#) de este documento.

La configuración del Catalyst 6500/6000 implica estos cinco pasos:

1. [Habilitar QoS \(Calidad de servicio\)](#)
2. [Asignar cada valor de clase de servicio \(CoS\) posible a una cola y un umbral](#) (opcional)
3. [Configure el peso WRR](#) (opcional)
4. [Configure los búferes que se asignan a cada cola](#) (opcional)
5. [Configure el nivel de umbral para cada cola](#) (opcional)

**Nota:** Cada uno de estos pasos es opcional, con la excepción del Paso 1. Puede decidir dejar el valor predeterminado para uno o varios parámetros.

### [Paso 1: Habilitar QoS \(Calidad de servicio\)](#)

Primero, active QoS. Recuerde que QoS está desactivada de manera predeterminada. Cuando se inhabilita QoS, la asignación de CoS que ha configurado no afecta el resultado. Hay una cola servida de la forma primero en entrar, primero en salir (FIFO) y todos los paquetes se descartan allí.

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#mls qos
```

```
QoS is enabled globally
Microflow policing is enabled globally
```

```
QoS global counters:
Total packets: 552638
IP shortcut packets: 0
```

```
Packets dropped by policing: 0
IP packets with TOS changed by policing: 0
IP packets with COS changed by policing: 0
Non-IP packets with CoS changed by policing: 0
```

## Paso 2: Asignar cada valor CoS posible a una cola y un umbral

Para todos los tipos de cola, asigne el CoS a una cola y un umbral. El mapping definido para un tipo de puerto 2q2t no se aplica a ningún puerto 1p2q2t. Además, el mapping para 2q2t se aplica a todos los puertos que tienen un mecanismo de colocación en cola 2q2t. Ejecute estos comandos **cos-map** en la interfaz:

```
wrr-queue cos-map Q_number_(1-2) threshold_number_(1-2) cos_value_1 cos_value_2
priority-queue cos-map Q_number_(always 1) cos_value_1 cos_value_2
```

**Nota:** Cada uno de estos comandos debe estar en una línea.

Puede configurar por separado la cola WRR. Si hay una cola de prioridad, puede configurarla con el comando **priority-queue**.

**Nota:** Las colas siempre se numeran comenzando por la cola de prioridad más baja posible y terminando con la cola de prioridad estricta disponible. Por ejemplo:

- La cola 1 es la cola WRR de baja prioridad.
- La cola 2 es la cola WRR de alta prioridad.
- La cola 3 es la cola de prioridad estricta.

Repita esta operación para todos los tipos de colas, de lo contrario la asignación CoS predeterminada permanecerá. Este es un ejemplo de configuración para 1p2q2t:

```
cosmos#configure terminal
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#priority-queue cos-map 1 5
!--- Assign a CoS of 5 to priority queue. cos-map configured on: Gil/1 Gil/2 cosmos(config-
if)#wrr-queue cos-map 1 1 0 1
!--- Assign CoS 0 and 1 to the first threshold of low-priority WRR queue. cos-map configured on:
Gil/1 Gil/2 cosmos(config-if)#wrr-queue cos-map 1 2 2 3
!--- Assign CoS 2 and 3 to the second threshold of low-priority WRR queue. cos-map configured
on: Gil/1 Gil/2 cosmos(config-if)#wrr-queue cos-map 2 1 4 6
!--- Assign CoS 4 and 6 to the first threshold of high-priority WRR queue. cos-map configured
on: Gil/1 Gil/2 cosmos(config-if)#wrr-queue cos-map 2 2 7
!--- Assign CoS 7 to the first threshold of high-priority WRR queue. cos-map configured on:
Gil/1 Gil/2
```

### Verifique la configuración

```
cosmos#show queueing interface gigabitethernet 1/1
!--- Output suppressed. queue thresh cos-map ----- 1 1 0 1 1 2
2 3 2 1 4 6 2 2 7 3 1 5 !--- Output suppressed.
```

## Paso 3: Configuración del peso de WRR

Configure el peso WRR para las dos colas WRR. Ejecute este comando de interfaz:

```
wrr-queue bandwidth weight_for_Q1 weight_for_Q2
```

El peso 1 se relaciona con la cola 1, que debe ser la cola WRR de baja prioridad. Mantenga siempre este peso un nivel inferior al peso 2. El peso puede tomar cualquier valor entre 1 y 255. Utilice estas fórmulas para asignar el porcentaje:

- Para la cola 1— $[\text{peso } 1 / (\text{peso } 1 + \text{peso } 2)]$
- Para la cola 2— $[\text{peso } 2 / (\text{peso } 1 + \text{peso } 2)]$

Debe definir el peso para todos los tipos de colas. No es necesario que estos tipos de peso sean iguales. Este es un ejemplo para 2q2t, donde la cola 1 se atiende el 20% del tiempo y la cola 2 se atiende el 80% del tiempo:

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue bandwidth ?
<1-255> enter bandwidth weight between 1 and 255
cosmos(config-if)#wrr-queue bandwidth 20 80
!--- Queue 1 is served 20% of the time, and queue 2 is served !--- 80% of the time.
cosmos(config-if)#
```

## Verifique la configuración

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
   1           WRR low      2
   2           WRR high     2
   3           Priority     1

  WRR bandwidth ratios:  20[queue 1]  80[queue 2]
  queue-limit ratios:    90[queue 1]  5[queue 2]
!--- Output suppressed.
```

**Nota:** Puede configurar diferentes pesos WRR para cada interfaz cuando no es posible utilizar el software CatOS.

## Paso 4: Configure los búferes que se asignan a cada cola

Debe definir la relación de cola de transmisión. Esto determina cómo se dividen las memorias intermedias entre las diferentes colas.

```
wrr-queue queue-limit percentage_WRR_Q1 percentage_WRR_Q2
cosmos(config)#interface gigabitethernet 1/2
cosmos(config-if)#wrr-queue queue-limit 70 15
!--- Queue 1 has 70% of the buffers. !--- Queues 2 and 3 both have 15% of the buffers. queue-
limit configured on: Gi1/1 Gi1/2
```

**Nota:** Si la capacidad de colocación en cola de su puerto gigabit es 1p1q2t, debe utilizar el mismo nivel para la cola de prioridad estricta y para la cola WRR de alta prioridad. Estos niveles no

pueden variar por motivos de hardware. Sólo se configura el ancho de banda para las dos colas WRR. Utilice automáticamente el mismo valor para la cola WRR de alta prioridad y la cola de prioridad estricta, si existe alguna.

Algunos tipos de cola no tienen un tamaño de cola ajustable. Un ejemplo es el 1p3q1t, que está disponible en WS-X6548RJ45. Estos tipos de cola están corregidos y no se pueden modificar.

## Verifique la configuración

```
cosmos#show queueing interface gigabitethernet 1/2
Interface GigabitEthernet1/2 queueing strategy:  Weighted Round-Robin
  Port QoS is enabled
  Port is untrusted
  Default cos is 0
  Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1             WRR low    2
  2             WRR high   2
  3             Priority   1

  WRR bandwidth ratios:    5[queue 1] 255[queue 2]
  queue-limit ratios:     70[queue 1] 15[queue 2]
```

**Nota:** Es mejor dejar la mayor parte de los búfers para la cola WRR de baja prioridad. Esta es la cola en la que debe habilitar el almacenamiento en búfer adicional. Las otras colas tienen prioridad de servicio más alta.

## [Paso 5: Configurar el nivel de umbral para cada cola](#)

Como paso final, configure el nivel de umbral para la cola WRED o para la cola de descarte de cola. Esta lista proporciona los comandos:

- Para las colas que utilizan WRED como mecanismo de caída para el umbral, ejecute estos comandos:

```
wrr-queue random-detect min-threshold Q_number threshold_1_value threshold_2_value
wrr-queue random-detect max-threshold Q_number threshold_1_value threshold_2_value
```

**Nota:** Cada uno de estos comandos debe estar en una línea.

- Para las colas que utilizan el descarte de cola como mecanismo de descarte, ejecute este comando:

```
wrr-queue threshold Q_number threshold_1_value threshold_2_value
```

**Nota:** Este comando debe estar en una línea.

## Configuración para una cola WRED:

```
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue random-detect min-threshold 1 20 50
!--- This sets the threshold of queue 1 to 20 and 50% minimum threshold !--- configured on Gi1/1
cosmos(config-if)#wrr-queue random-detect min-threshold 2 20 50
!--- This sets the threshold of queue 2 to 20 and 50% minimum threshold !--- configured on Gi1/1
```

```

Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 1 50 80
!--- This sets the threshold of queue 1 to 50 and 80% maximum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 2 40 60
!--- This sets the threshold of queue 2 to 49 and 60% maximum threshold !--- configured on Gi1/1
Gi1/2.

```

## Configuración para una cola de eliminación de cola:

```

cosmos(config)#interface fastethernet 3/1
cosmos(config-if)#wrr-queue threshold ?
<1-2> enter threshold queue id (1-2)
cosmos(config-if)#wrr-queue threshold 1 ?
<1-100> enter percent of queue size between 1 and 100
cosmos(config-if)#wrr-queue threshold 1 50 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 1 (low-priority)
to 50 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)# cosmos(config-if)# cosmos(config-if)#wrr-
queue threshold 2 40 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 2 (high-priority)
to 40 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)#

```

## Verifique la configuración

```

cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = lp2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low     2
2             WRR high    2
3             Priority     1

WRR bandwidth ratios: 20[queue 1] 80[queue 2]
queue-limit ratios: 70[queue 1] 15[queue 2]

queue random-detect-min-thresholds
-----
1      20[1] 50[2]
2      20[1] 50[2]

queue random-detect-max-thresholds
-----
1      50[1] 80[2]
2      40[1] 60[2]

```

```

cosmos#show queueing interface fastethernet 3/1
Interface FastEthernet3/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low     2
2             WRR high    2

WRR bandwidth ratios: 100[queue 1] 255[queue 2]

```

```
queue-limit ratios:      90[queue 1] 10[queue 2]
```

```
queue tail-drop-thresholds
```

```
-----
```

```
1      50[1] 100[2]
2      40[1] 100[2]
```

No puede configurar el umbral y asignar el CoS a la cola por puerto. Todos los cambios se aplican a un conjunto de puertos contiguos:

- Cuatro puertos para tarjetas de línea gigabit: los puertos 1 a 4 están juntos y los puertos 5 a 8 están juntos.
- Doce puertos para puertos de 10/100 o 100 puertos de fibra basados en cola 1q4/2q2t: de 1 a 12, de 13 a 24, de 25 a 36 y de 36 a 48.
- Para determinar el puerto exacto que pertenece al mismo ASIC, utilice el comando **show interface capabilities**.

## [Supervisión de la programación de resultados y verificación de configuraciones](#)

El comando más fácil de ejecutar para verificar la configuración actual en tiempo de ejecución de un puerto con respecto a la programación de salida es el comando **show queueing interface {gigabitethernet | fastethernet} slot/puerto**. Este comando muestra el tipo de colocación en cola en el puerto, la asignación de CoS a las diferentes colas y umbrales, el uso compartido del búfer y el peso de WRR. Aquí, es el 20% de WRR para la cola 1 y el 80% de WRR para la cola 2. El comando también muestra toda la información configurada para la programación de salida y el número de paquetes que se descartan en cada cola para cada umbral:

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default COS is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
    1          WRR low           2
    2          WRR high           2
    3          Priority           1
WRR bandwidth ratios:  20[queue 1] 80[queue 2]
queue-limit ratios:    70[queue 1] 15[queue 2]

queue random-detect-max-thresholds
-----
  1      50[1] 80[2]
  2      40[1] 60[2]

queue thresh cos-map
-----
  1      1      0 1
  1      2      2 3
  2      1      4 6
  2      2      7
  3      1      5

Receive queues [type = 1plq4t]:
  Queue Id      Scheduling  Num of thresholds
  -----
```

```

1      Standard      4
2      Priority      1

```

```

queue tail-drop-thresholds
-----
1      100[1] 100[2] 100[3] 100[4]

```

```

queue thresh cos-map
-----
1      1      0 1
1      2      2 3
1      3      4
1      4      6 7
2      1      5

```

Packets dropped on Transmit:  
 BPDU packets: 0

```

queue thresh      dropped  [cos-map]
-----
1      1              0 [0 1 ]
1      2              0 [2 3 ]
2      1              0 [4 6 ]
2      2              0 [7 ]
3      1              0 [5 ]

```

Packets dropped on Receive:  
 BPDU packets: 0

```

queue thresh      dropped  [cos-map]
-----
1      1              0 [0 1 ]
1      2              0 [2 3 ]
1      3              0 [4 ]
1      4              0 [6 7 ]
2      1              0 [5 ]

```

## Ejemplo de programación de salida

Este tráfico se inyecta en el Catalyst 6500/6000:

- En el puerto gigabit 1/2: un gigabit de tráfico con precedencia de cero
- En el puerto gigabit 5/2: 133 MB de tráfico con precedencia de siete 133 MB de tráfico con precedencia de seis 133 MB de tráfico con precedencia de cinco 133 MB de tráfico con precedencia de cuatro 133 MB de tráfico con precedencia de tres 133 MB de tráfico con precedencia de dos 133 MB de tráfico con precedencia de uno

Todo el tráfico de unidifusión sale del switch por puerto gigabit 1/1, que está muy sobresuscrito.

## Caso 1: QoS está habilitado y se utiliza un parámetro predeterminado

El comando **show queueing interface gigabitethernet 1/1** configura todos los resultados en este ejemplo. El comando proporciona información adicional sobre la programación de entrada. Sin embargo, como este documento sólo cubre la programación de salida, suprime esa salida.

Cuando QoS está globalmente habilitado y todos los parámetros predeterminados están en uso, este resultado se produce después de unos minutos:

```

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
    1           WRR low      2
    2           WRR high     2
    3           Priority    1

WRR bandwidth ratios: 100[queue 1] 255[queue 2]
queue-limit ratios:   90[queue 1]   5[queue 2]

queue random-detect-max-thresholds
-----
  1    40[1] 100[2]
  2    40[1] 100[2]

queue thresh cos-map
-----
  1    1      0 1
  1    2      2 3
  2    1      4
  2    2      6 7
  3    1      5

Packets dropped on Transmit:
  BPDU packets: 0

queue thresh      dropped  [cos-map]
-----
  1    1      149606424  [0 1 ]
  1    2           0  [2 3 ]
  2    1      16551394  [4 ]
  2    2      4254446   [6 7 ]
  3    1           0  [5 ]

```

En este resultado, los valores predeterminados son:

- Peso WRR para la cola 1— $100 / (100 + 255) = 28\%$
- Peso WRR para cola 2— $255 / (255 + 100) = 72\%$
- Uso compartido de búfer: —90% para la cola 1, 5% para la cola 2 y 5% para la cola de prioridad estricta

La mayoría de los paquetes en la cola WRR de baja prioridad se descartan, pero algunos se siguen descartando en la cola WRR de alta prioridad para ambos umbrales. Hay un total de 170.412.264 gotas ( $149.606.424 + 16.551.394 + 4.254.446$ ). Estas caídas se dividen de la siguiente manera:

- $149,606,424 / 170,412,264 = 88\%$  de las caídas en la cola 1 (primer paquete de umbral con CoS 0 y 1)
- $16\,551\,394 / 170\,412\,264 = 10\%$  de las caídas en la cola 2 (primer paquete de umbral con CoS 4)
- $4,254,446 / 170,412,264 = 2\%$  de las caídas en la cola 2 (paquete de segundo umbral con CoS de 6 ó 7)

**Nota:** No ve ninguna caída en la cola de prioridad estricta.

## Caso 2: Modificar el peso de WRR

Como se señala en el [caso 1: QoS está habilitado y se utiliza un parámetro predeterminado](#), los paquetes en la cola 2 siguen siendo eliminados. Modifique el peso de WRR para dar más ancho de banda a la cola 2. Ahora, la cola 1 se vacía el 4% del tiempo y la cola 2 se vacía el 96% del tiempo:

```
show run interface gigabitethernet 1/1
interface GigabitEthernet1/1
  no ip address
  wrr-queue bandwidth 10 255
  mls qos trust dscp
  switchport
  switchport mode access
end

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
  Port QoS is enabled
  Trust state: trust DSCP
  Default cos is 0
  Transmit queues [type = lp2q2t]:
    Queue Id      Scheduling  Num of thresholds
    -----
      1           WRR low           2
      2           WRR high          2
      3           Priority          1

  WRR bandwidth ratios:  10[queue 1] 255[queue 2]
  queue-limit ratios:    90[queue 1]  5[queue 2]

  queue random-detect-max-thresholds
  -----
    1   40[1] 100[2]
    2   40[1] 100[2]

  queue thresh cos-map
  -----
    1   1   0 1
    1   2   2 3
    2   1   4
    2   2   6 7
    3   1   5

  Packets dropped on Transmit:
  BPDU packets:  0

  queue thresh      dropped  [cos-map]
  -----
    1   1           2786205  [0 1 ]
    1   2              0  [2 3 ]
    2   1           11363   [4 ]
    2   2              69  [6 7 ]
    3   1              0   [5 ]
```

Como se ve en este resultado, el porcentaje de caídas en la cola 2 es ahora mucho más bajo. Un total de 2.797.637 caídas se dividen de esta manera:

- $2,786,205 / 2,797,637 = 99,591\%$  de las caídas en la cola 1 (con paquete de CoS 0 y 1)

- $11,363 / 2,797,637 = 0.408\%$  de las caídas en la cola 2 (primer umbral con el paquete CoS 4)
- $69 / 2,797,637 = 0.001\%$  de las caídas en la cola 2 (segundo umbral para el paquete con CoS 6 y 7)

Si utiliza varios pesos WRR, se asegura más QoS en la cola 2.

### Caso 3: Modificación de peso adicional de WRR

Puede ser aún más agresivo con el peso de WRR. En este ejemplo de salida, sólo 0.39 por ciento del peso se da a la cola 1:

```

show run interface gigabitethernet 1/1
interface GigabitEthernet1/1
no ip address
wrr-queue bandwidth 1 255
mls qos trust dscp
switchport
switchport mode access
end

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
    1           WRR low      2
    2           WRR high     2
    3           Priority     1

WRR bandwidth ratios:      1[queue 1] 255[queue 2]
queue-limit ratios:       90[queue 1]   5[queue 2]

queue random-detect-max-thresholds
-----
  1    40[1] 100[2]
  2    40[1] 100[2]

queue thresh cos-map
-----
  1    1    0 1
  1    2    2 3
  2    1    4
  2    2    6 7
  3    1    5

Packets dropped on Transmit:
  BPDU packets: 0

queue thresh      dropped  [cos-map]
-----
  1    1          2535315  [0 1 ]
  1    2              0  [2 3 ]
  2    1           705  [4 ]
  2    2            73  [6 7 ]
  3    1              0  [5 ]

```

Incluso con el peso agresivo de WRR, los paquetes siguen siendo descartados en la cola 2. Sin

embargo, en comparación, no son muchos los paquetes. Ahora sólo hay una caída de paquete del 0,03% en la cola 2.

#### Caso 4: Modificar asignación de búfer de límite de cola

Como se observa en el [caso 2: Modificar el peso de WRR](#) y el [caso 3: Secciones adicionales de modificación del peso de WRR](#), los paquetes siguen cayendo en la cola 2, aunque el porcentaje de WRR garantiza que la caída es mínima. Sin embargo, cuando se alcanza el segundo umbral (que se establece en el 100%) en la cola 2, algunos paquetes siguen siendo descartados.

Para mejorar esto, cambie el límite de cola (tamaño del búfer asignado a cada cola). En este ejemplo, el límite de cola se establece en 70 por ciento para la cola 1, 15 por ciento para la cola 2 y 15 por ciento para la cola de prioridad estricta:

```
show run gigabitethernet 1/1
interface GigabitEthernet1/1
  no ip address
  wrr-queue bandwidth 1 255
  wrr-queue queue-limit 70 15
  mls qos trust dscp
  switchport
  switchport mode access
end
```

```
nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
   1           WRR low           2
   2           WRR high          2
   3           Priority          1

WRR bandwidth ratios:   1[queue 1] 255[queue 2]
queue-limit ratios:    70[queue 1] 15[queue 2]

queue random-detect-max-thresholds
-----
  1    40[1] 100[2]
  2    40[1] 100[2]

queue thresh cos-map
-----
  1    1    0 1
  1    2    2 3
  2    1    4
  2    2    6 7
  3    1    5
```

```
Receive queues [type = lplq4t]:
  Queue Id      Scheduling  Num of thresholds
  -----
   1           Standard          4
   2           Priority          1
```

```

queue tail-drop-thresholds
-----
1      100[1] 100[2] 100[3] 100[4]

queue thresh cos-map
-----
1      1      0 1
1      2      2 3
1      3      4
1      4      6 7
2      1      5

```

```

Packets dropped on Transmit:
BPDU packets: 0

```

```

queue thresh      dropped  [cos-map]
-----
1      1          154253046  [0 1 ]
1      2              0  [2 3 ]
2      1              0  [4 ]
2      2              0  [6 7 ]
3      1              0  [5 ]

```

Ahora, las caídas sólo ocurren en la cola 1.

## [Utilice la programación de salida para reducir la demora y la fluctuación](#)

Los estudios de casos en la sección [Ejemplo de Programación de Salida](#) demuestran la ventaja de implementar la programación de salida para evitar una caída del tráfico VoIP o de misión crítica en caso de una sobresuscripción del puerto de salida. La sobresuscripción no se produce con mucha frecuencia en una red normal (especialmente en un link gigabit). La suscripción excesiva solo debe producirse durante las horas punta del tráfico o durante las ráfagas de tráfico que se producen en un período de tiempo muy breve.

Incluso sin ninguna suscripción excesiva, la programación de salida puede ser de gran ayuda en una red donde la QoS se implementa de principio a fin. Esta sección proporciona ejemplos de cómo la programación de resultados puede ayudar a reducir la demora y la fluctuación.

### [Reducir el retraso](#)

El retraso de un paquete aumenta debido al tiempo "perdido" en el buffer de cada switch mientras espera ser transmitido. Por ejemplo, un paquete de voz pequeño con un CoS de 5 se envía desde un puerto durante una copia de seguridad o transferencia de archivos de gran tamaño. Suponga que no hay QoS para el puerto de salida y que el paquete de voz pequeño se pone en cola después de 10 paquetes grandes de 1500 bytes. En este caso, puede calcular fácilmente que el tiempo de velocidad de gigabit necesario para transmitir los 10 paquetes grandes es:

- $(10 \times 1500 \times 8) = 120\ 000$  bits transmitidos en 120 microsegundos

Si este paquete necesita atravesar ocho o nueve switches mientras pasa por la red, puede producirse un retraso de aproximadamente 1 milisegundo. Esto incluye solamente los retrasos en la cola de salida del switch cruzado en la red.

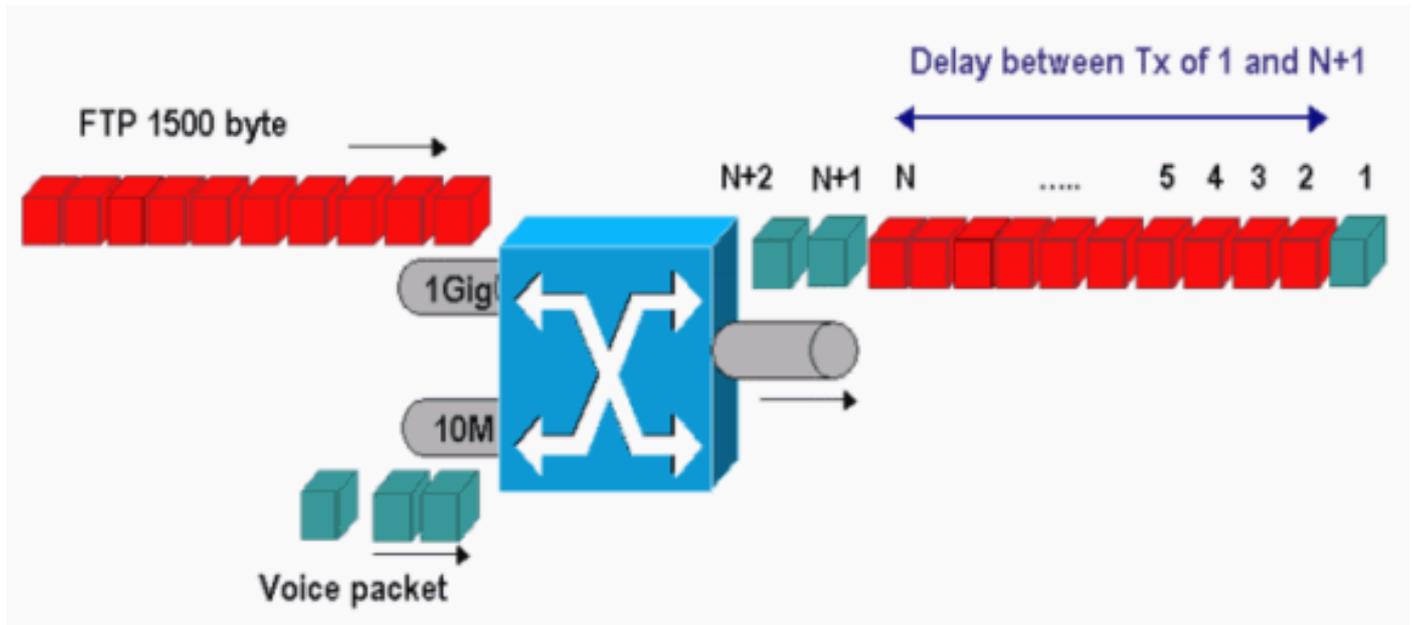
**Nota:** Si necesita poner en cola los mismos 10 paquetes grandes en una interfaz de 10 MB (por ejemplo, conectados a un teléfono IP y a un PC), la presentación de la demora es:

- $(10 \times 1500 \times 8) = 120\,000$  bits transmitidos en 12 milisegundos

La implementación de la programación de salida asegura que los paquetes de voz con una CoS de 5 se coloquen en la cola de prioridad estricta y se envíen antes que cualquier paquete con una CoS inferior a 5. Esto reduce el retraso.

## Reducción de la fluctuación

Otra ventaja importante de la programación de salida es la reducción de la fluctuación. La fluctuación es la variación en la demora para los paquetes dentro del mismo flujo. Este escenario de ejemplo muestra cómo la programación de salida puede reducir la fluctuación:



En este escenario, el mismo puerto de salida necesita enviar dos secuencias:

- Un flujo de voz entrante en un puerto Ethernet de 10 MB.
- Una secuencia FTP entrante en un puerto de enlace ascendente de 1 Gigabit Ethernet.

Ambos flujos dejan el switch a través del mismo puerto de salida. Este ejemplo muestra lo que puede ocurrir sin el uso de la programación de salida. Todos los paquetes de datos de gran tamaño pueden entrelazarse entre dos paquetes de voz. Esto crea fluctuación en la recepción del paquete de voz desde la misma secuencia. Hay una demora mayor entre la recepción del paquete 1 y el paquete  $n + 1$  a medida que el switch transmite el paquete de datos grande. Sin embargo, el retraso entre  $n + 1$  y  $n + 2$  es insignificante. Esto produce fluctuación en el flujo de tráfico de voz. Puede evitar fácilmente este problema con el uso de una cola de prioridad estricta. Asegúrese de asignar el valor CoS de los paquetes de voz a la cola de prioridad estricta.

## Conclusión

En este documento, ha visto casos prácticos de cómo configurar y resolver problemas de programación de cola de salida en un Catalyst 6500/6000 que ejecuta Cisco IOS Software. También ha visto las ventajas de la programación de salida en la mayoría de las redes con tráfico de voz:

- Evita la caída del tráfico crítico en caso de exceso de suscripción del puerto de salida.
- Reduce el retraso.

- Reduce la fluctuación.

## Información Relacionada

- [Programa de salida de QoS en los switches de la serie Catalyst 6500/6000 con software del sistema CatOS](#)
- [La calidad del servicio en la familia de switches Catalyst 6000](#)
- [Páginas de Soporte de Productos de LAN](#)
- [Página de Soporte de LAN Switching](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)