

# Comprensión y configuración de MDRR/WRED en routers de la serie 12000

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

[Descripción general de MDRR](#)

[Cola prioritaria de MDRR](#)

[Ejemplo de MDRR](#)

[Soporte MDRR por tipo de motor](#)

[Información general sobre WRED](#)

[Utilizar sintaxis de CoS heredada para la configuración](#)

[Utilizar CLI de QoS modular \(MQC\) para la configuración](#)

[Comandos para supervisar la gestión de la congestión y la prevención](#)

[El comando show interfaces](#)

[Comando show interfaces {number} random](#)

[Comando exec slot \(y\) show controller frfab queue {port}](#)

[Comando exec slot \(y\) show controller frfab QM stat](#)

[Supervisar la administración de congestión entrante](#)

[El comando show interfaces](#)

[Comando exec slot \(x\) show controller tofab queue](#)

[Comando exec slot \(x\) show controller tofab queue \(slot\) \(port\)](#)

[Comando exec slot \(x\) show controller tofab QM stat](#)

[Caso Práctico](#)

[Información Relacionada](#)

## [Introducción](#)

Este documento revisa cómo configurar las funciones de administración de congestión y prevención de congestión del software Cisco IOS® en el router de Internet de la serie Cisco 12000.

Después de leer este documento, debe poder:

- Comprenda por qué es importante configurar el método de selección cíclica de déficit modificado (MDRR) y la detección temprana aleatoria ponderada (WRED) en la red principal.

- Comprenda la implementación que subyace a MDRR y WRED en la serie Cisco 12000.
- Configure MDRR y WRED utilizando la sintaxis heredada de Clase de servicio (CoS) y modular CLI QoS (MQC).

## Prerequisites

### Requirements

Quienes lean este documento deben tener conocimiento de los siguientes temas:

- Familiaridad general con la arquitectura del router de Internet de la serie Cisco 12000.
- En particular, un reconocimiento de la arquitectura de colocación en cola y estos términos: ToFab (Hacia el fabric): que describe las colas del lado de recepción en una tarjeta de línea entrante. FrFab (Desde el fabric): que describe las colas del lado de transmisión en una tarjeta de línea saliente.

**Nota:** También se recomienda buscar [Cómo leer el resultado del comando show controller frfab | Comandos tofab queue en un router de Internet de la serie 12000 de Cisco](#).

### Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Todas las plataformas Cisco 12000, que incluyen las 12008, 12012, 12016, 12404, 12406, 12410 y 12416.
- Versión 12.0(24)S1 del software del IOS de Cisco.

**Nota:** Aunque las configuraciones de este documento se probaron en Cisco IOS Software Release 12.0(24)S1, se puede utilizar cualquier versión de Cisco IOS Software que soporte el Cisco 12000 Series Internet Router.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

### Convenciones

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

## Antecedentes

Los métodos de almacenamiento en cola definen el mecanismo de programación de paquetes o el orden en que los paquetes se quitan de la cola a la interfaz para su transmisión en el cable físico. En función del orden y el número de veces que una función del programador atiende una cola, los métodos de colocación en cola también admiten garantías de ancho de banda mínimo y latencias bajas.

Es importante asegurarse de que un mecanismo de programación de paquetes soporte la arquitectura de conmutación en la que se implementa. Weighted fair queuing (WFQ) es el

algoritmo de programación conocido para la asignación de recursos en las plataformas de router de Cisco con una arquitectura basada en bus. Sin embargo, no es compatible con el router de Internet de la serie Cisco 12000. Tampoco se admiten la colocación en cola de prioridad del software Cisco IOS tradicional ni la colocación en cola personalizada. En su lugar, la serie 12000 de Cisco utiliza una forma especial de colocación en cola llamada Turno de déficit modificado (MDRR), que proporciona garantías de ancho de banda relativo y una cola de latencia baja. La M de MDRR significa "modificado"; agrega la cola de prioridad comparada con DRR donde no hay cola de prioridad presente. Para obtener más información sobre MDRR, consulte la sección [Descripción General de MDRR](#).

Además, la serie 12000 de Cisco admite la Detección temprana aleatoria ponderada (WRED) como política de caída para las colas MDRR. Este mecanismo de prevención de congestión proporciona una alternativa al mecanismo de caída de cola predeterminado. La congestión puede evitarse mediante eliminaciones controladas.

La prevención de la congestión y los mecanismos de gestión como WRED y MDRR son particularmente importantes en las colas FrFab de interfaces salientes de velocidad relativamente baja, como las tarjetas de línea canalizadas (LC). El entramado de switches de alta velocidad puede enviar paquetes a los grupos de canales mucho más rápido de lo que los grupos de canales pueden transmitirlos. Como la colocación en cola / los búfers se administran en el nivel de puerto físico, la contrapresión en un canal puede afectar a todos los demás canales en ese puerto. Por lo tanto, es muy importante gestionar esa contrapresión a través de WRED/MDRR, que limita el impacto de la contrapresión a los canales en cuestión. Para obtener información detallada sobre cómo administrar la sobresuscripción de la interfaz saliente, vea [Resolución de problemas de paquetes ignorados y caídas sin memoria en el router de Internet de la serie 12000 de Cisco](#).

## [Descripción general de MDRR](#)

Esta sección proporciona una descripción general de la Ordenación cíclica por déficit modificado (MDRR).

Con MDRR configurado como estrategia de colocación en cola, las colas que no están vacías se atienden una tras otra, de forma rotativa. Cada vez que se atiende una cola, se quita una cantidad fija de datos de la cola. A continuación, el algoritmo presta servicios a la cola siguiente. Cuando se atiende una cola, MDRR realiza un seguimiento del número de bytes de datos que se quitaron de la cola en exceso del valor configurado. En el paso siguiente, cuando se vuelve a enviar la cola, se quitarán menos datos para compensar el exceso de datos que se había suministrado anteriormente. Como resultado, la cantidad promedio de datos quitados de la cola por cola será cercana al valor configurado. Además, MDRR mantiene una cola de prioridad que se atiende de forma preferencial. MDRR se explica con más detalle en esta sección.

Cada cola dentro de MDRR se define mediante dos variables:

- Valor cuántico: es el número promedio de bytes que se sirven en cada ronda.
- Contador de déficit: se utiliza para realizar un seguimiento de cuántos bytes ha transmitido una cola en cada ronda. Se inicializa al valor cuántico.

Los paquetes de una cola se sirven siempre y cuando el contador de déficit sea mayor que cero. Cada paquete en servicio hace disminuir el contador en déficit en un valor equivalente a su longitud en bytes. Una cola no puede ser atendida después de que el contador en déficit sea cero o negativo. En cada nueva ronda, el contador de déficit de cada cola no vacía se incrementa por

su valor cuántico.

**Nota:** En general, el tamaño cuántico de una cola no debe ser menor que la unidad máxima de transmisión (MTU) de la interfaz. Esto asegura que el planificador siempre atiende al menos un paquete de cada cola no vacía.

Se le puede adjudicar un peso relativo a cada cola MDRR. Una de las colas del grupo se define como cola de prioridad. Los pesos asignan ancho de banda relativo para cada cola cuando la interfaz está congestionada. El algoritmo MDRR quita de la cola los datos de cada cola de un modo de ordenamiento si hay datos en la cola que se van a enviar.

Si todas las colas MDRR regulares tienen datos, se atienden de la siguiente manera:

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

Durante cada ciclo, una cola puede quitarle la cola a una cantidad en función de su peso configurado. En las tarjetas de línea de Motor 0 y Motor 2, un valor de 1 es equivalente a dar a la interfaz un peso de su MTU. Para cada incremento superior a 1, el peso de la cola aumenta en 512 bytes. Por ejemplo, si la MTU de una interfaz particular es 4470 y el peso de una cola está configurado como 3, son  $4470 + (3-1)*512 = 5494$  los bytes que pueden salir de la cola por vez a través de la rotación. Si se utilizan dos colas DRR normales, Q0 y Q1, Q0 se configura con un peso de 1 y Q1 se configura con un peso de 9. Si ambas colas están congestionadas, cada vez que se realiza la rotación, Q0 podría enviar 4470 bytes y Q1 podría enviar  $[4470 + (9-1)*512] = 8566$  bytes. Esto daría al tráfico que llega a Q0 aproximadamente 1/3 del ancho de banda, y al tráfico que pasa a través de Q1 aproximadamente 2/3 del ancho de banda.

**Nota:** La fórmula estándar de eliminación de cola utilizada para calcular la asignación de ancho de banda MDRR es  $D = MTU + (\text{peso}-1)*512$ . En las versiones anteriores a la versión 12.0(21) S/ST del software del IOS de Cisco, las tarjetas de línea del Motor 4 utilizaron una fórmula de eliminación de cola diferente. Para configurar el peso MDRR para una asignación de ancho de banda correcta, asegúrese de ejecutar una versión de software del IOS de Cisco posterior a 12.0(21) S/ST.

**Nota:** La fórmula cuántica para las tarjetas de línea del Motor 4+ es (para la dirección toFab, esto no es válido para FrFab)  $\text{Quantum} = \text{PesoBase} + \{\text{PesoBase} * (\text{PesoCola} - 1) * 512\} / \text{MTU}$ . BaseWeight se obtiene con esta fórmula:  $\text{Peso base} = \{(MTU + 512 - 1) / 512\} + 5$

**Nota:** Todos los cálculos se redondean hacia fuera; es decir, se ignoran todos los decimales.

**Nota:** Para saber si una tarjeta de línea específica del motor admite MDRR, consulte [Soporte MDRR por Tipo de Motor](#).

## [Cola prioritaria de MDRR](#)

La serie 12000 de Cisco admite una cola de prioridad (PQ) dentro de MDRR. Esta cola proporciona la baja demora y la baja fluctuación requeridas por el tráfico sensible al tiempo, como Voz sobre IP (VoIP).

Como se observó anteriormente el Cisco series 12000 no admite weighted fair queueing (WFQ). Por lo tanto, el PQ dentro de MDRR opera de manera diferente de la función de colocación en Cola de tiempo de latencia bajo (LLQ) del software del IOS de Cisco disponible para otras plataformas.

Una diferencia clave es cómo se puede configurar el planificador MDRR para que atienda la PQ en uno de los dos modos, como se muestra en la [tabla 1](#):

**Tabla 1: Cómo Configurar el Planificador MDRR para que Preste Servicio a la PQ en Dos Modos**

	Modo alternativo	Modo de prioridad estricta
Ventajas	Aquí, el PQ se mantiene entre las otras colas. En otras palabras, el planificador MDRR proporciona servicios alternativos a la PQ y a cualquier otra cola configurada.	Aquí se atiende la PQ siempre que no esté vacía. Esto proporciona el menor retraso posible para el tráfico coincidente.
Desventajas	Este modo puede introducir fluctuación y retardo cuando se compara con el modo de prioridad estricta.	Este modo puede dejar de tener otras colas, particularmente si los flujos coincidentes son remitentes agresivos.

El modo alternativo puede ejercer menos control sobre la fluctuación y el retraso. Si el planificador MDRR comienza a prestar servicio a tramas de tamaño MTU desde una cola de datos y luego llega un paquete de voz a la PQ, el programador en modo alternativo atiende completamente a la cola de no prioridad hasta que su contador de déficit alcance cero. Durante este tiempo, el PQ no se mantiene y los paquetes VoIP se retrasan.

Por el contrario, en el modo de prioridad estricta, el planificador sólo presta servicios al paquete no prioritario actual y luego pasa a la PQ. El planificador comienza a prestar servicio a una cola no prioritaria sólo después de que la PQ se vacíe completamente.

Es importante tener en cuenta que la cola de prioridad en el modo de prioridad alternativo se atiende más de una vez en un ciclo y, por lo tanto, toma más ancho de banda que otras colas con el mismo peso nominal. Cuánto más depende de la cantidad de colas definidas. Por ejemplo, con tres colas, la cola de latencia baja se atiende dos veces más que las otras colas y envía el doble de su peso por ciclo. Si se definen ocho colas, la cola de latencia baja recibe servicio con una frecuencia siete veces mayor y el peso eficaz es siete veces superior. Por lo tanto, el ancho de banda que puede tomar la cola se relaciona con la frecuencia con la que se atiende por ordenamiento cíclico, lo que a su vez depende de cuántas colas se definen en general. En el modo de prioridad alternativa, la cola de prioridad se configura generalmente con un peso pequeño por esta razón en particular.

Como ejemplo, supongamos que se definen cuatro colas: 0, 1, 2 y la cola de prioridad. En el modo de prioridad alternativo, si todas las colas están congestionadas, se les mantendría de la siguiente manera: 0, llq, 1, llq, 2, llq, 0, llq, 1, .... donde llq representa la cola de latencia baja.

Cada vez que se le presta servicio a una cola, ésta puede enviar hasta su peso configurado. Por lo tanto, el ancho de banda mínimo que puede tener la cola de latencia baja es:

- WL = peso de la cola de latencia baja.
- W0, W1, ... Wn = pesos de las colas DRR normales.
- n = número de colas DRR regulares utilizadas para esta interfaz.
- BW = Ancho de banda del link.

Para el modo de prioridad alternativo, el ancho de banda mínimo de la cola de tiempo de latencia bajo=  $BW * n * WL / (n * WL + \text{Sum}(W0,Wn))$ .

El peso es el único parámetro variable en MDRR que se puede configurar. Influye en la cantidad relativa de ancho de banda que puede utilizar una clase de tráfico y en la cantidad de tráfico que se envía en un turno. El uso de pesos más grandes significa que el ciclo general lleva más tiempo y posiblemente aumenta la latencia.

### Pautas de Configuración

- Es mejor configurar el peso de la clase que tiene el menor requerimiento de ancho de banda a 1 para mantener el retardo y la fluctuación lo más bajos posible entre las otras clases.
- Seleccione valores de peso que sean tan bajos como sea posible. Comience con un peso de 1 para la clase con el ancho de banda más bajo. Por ejemplo, cuando utiliza dos clases con un ancho de banda del 50% para cada clase, debe configurar 1 y 1. No tiene sentido utilizar 10 y 10, porque no hay impacto en el rendimiento cuando elige 1. Además, un peso mayor introduce mayor fluctuación.
- Un valor de bajo peso para el LLQ es muy importante, especialmente en el modo alternativo para no agregar demasiada demora o fluctuación a las otras clases.

### Ejemplo de MDRR

El ejemplo de esta sección se toma de *Dentro de la Arquitectura de Cisco IOS® Software*, Cisco Press.

Suponga que tenemos tres colas:

- **Queue 0** – tiene una cantidad determinada de 1500 bytes; es la cola de latencia baja, configurada para operar en el modo alternativo.
- **Queue 1** – tiene una cantidad determinada de 3000 bytes.
- **Queue 2** – tiene una cantidad determinada de 1500 bytes.

[La figura 1](#) ilustra el estado inicial de las colas, junto con algunos paquetes que se han recibido y en cola.

Figura 1: Estado inicial de MDRR

	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

La cola 0 se mantiene primero, su cantidad se agrega a su contador de déficit, el paquete 1, que es de 250 bytes, se transmite y su tamaño se resta del contador de déficit. Debido a que el contador de déficit de la cola 0 es aún mayor que 0 ( $1500 - 250 = 1250$ ), el paquete 2 también se transmite, y su longitud se sustrae del contador de déficit. El contador de déficit de la cola 0 es ahora -250, por lo que la cola 1 se atiende a continuación. [La figura 2](#) indica este estado.

Figura 2: Estado posterior de MDRR

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

El contador de déficit de la cola 1 se establece en 3000 ( $0 + 3000 = 3000$ ) y los paquetes 4 y 5 se transmiten. Con cada paquete transmitido, reste el tamaño del paquete del contador de déficit, de modo que el contador de déficit de la cola 1 se reduzca a 0. [La figura 3](#) ilustra este estado.

Figura 3 - Estado de MDRR cuando el contador de déficit de la cola 1 es Cero

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

Necesitamos volver del modo de prioridad alternativo a la cola de servicio 0. Una vez más, la cantidad se agrega al contador de déficit actual, y el contador de déficit de la cola 0 se fija en el resultado ( $-250 + 1500 = 1250$ ). El paquete 3 es transmitido ahora debido a que el contador de déficit es mayor que 0 y la cola 0 se encuentra vacía. Cuando se vacía una cola, su contador de déficit se establece en 0, como se muestra en la Figura 4.

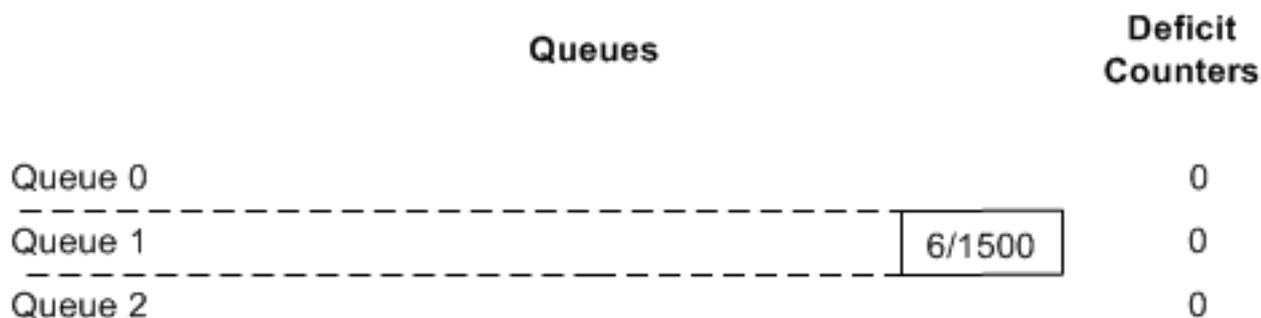
Figura 4: Estado de MDRR cuando una cola está vacía

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

La cola 2 se atiende a continuación; Su contador en déficit se configura a 1500 ( $0 + 1500 = 1500$ ). Los paquetes 7 a 10 se transmiten, lo que deja el contador de déficit en 500 ( $1500 - (4 \times 250) = 500$ ). Debido a que el contador del déficit es aún mayor que 0, también se transmite el paquete 11.

Cuando el paquete 11 es transmitido, la cola dos está vacía y el contador de déficit está configurado en 0, como se muestra en la Figura 5.

Figura 5: Estado de MDRR cuando se transmite el paquete 11



Luego, se vuelve a realizar un mantenimiento de la Cola 0 (ya que estamos en modo de prioridad alterna). Debido a que está vacío, servimos a la cola 1 a continuación y transmitimos el paquete 6.

### Soporte MDRR por tipo de motor

Las Series 12000 de Cisco soportan cinco modelos de tarjeta de línea con Capa 3 única (L3) reenviando arquitecturas de motor. El soporte para MDRR varía con el tipo de Motor L3 y el tipo de tarjeta. Por ejemplo, no hay soporte para MDRR y WRED en las tarjetas de línea ATM Engine 0. Para determinar el tipo de motor L3 de las tarjetas de línea instaladas, puede utilizar el comando show diag.

```
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)
```

### Colas ToFab (Rx) de MDRR

Puede utilizar la "Sintaxis de CoS heredada" o la "Interfaz de línea de comandos de QoS modular" para configurar MDRR en la serie Cisco 12000. En las secciones posteriores de este documento se explica cómo configurar MDRR con CoS heredada o QoS modular. Debe configurar las colas ToFab con la sintaxis heredada de CoS sólo porque no soportan la sintaxis más reciente de MQC. Véase [el cuadro 2](#) para más detalles.

Tabla 2: Detalles de MDRR en las colas ToFab (Rx)

	Implementados	ToFab MDRR	PQ alternativa ToFab	ToFab Strict PQ	ToFab WRED
Eng0	Software	No**	No**	Yes	Yes
Eng1	-	No	No	No	No
Eng2	Hardware	Yes	Yes	Yes	Yes
Eng3	Hardware	No	Yes	Yes	Yes
Eng4	Hardware	Yes	Yes	Yes	Yes

	e				
Eng4+	Hardware	Yes	Yes	Yes	Yes

\*\* MDRR es soportado en Motores LC 0 en la dirección ToFab (Rx), pero sólo el modo prioridad estricta, no el modo prioridad alternada. Las siete colas restantes se admiten como de costumbre.

Las interfaces entrantes mantienen una cola de salida virtual independiente por LC de destino. El modo de implementación de estas colas depende del tipo de motor L3.

- Motor 0 - Las LC entrantes mantienen ocho colas por ranura de destino. Por lo tanto, el número máximo de colas es  $16 \times 8 = 128$ . Cada cola se puede configurar por separado.
- Motores 2, 3, 4 y 4+ - Las LC entrantes mantienen ocho colas por interfaz de destino. Con 16 ranuras de destino y 16 interfaces por ranura, el número máximo de colas es  $16 \times 16 \times 8 = 2048$ . Todas las interfaces en una ranura de destino deben utilizar los mismos parámetros.

### [MDRR en colas FrFab \(Tx\)](#)

MDRR en las colas FrFab funciona de manera uniforme tanto si se implementa en hardware como en software. Todos los tipos de motor L3 admiten ocho colas de clase para cada interfaz saliente. Véase el [cuadro 3](#) para más detalles.

**Tabla 3: Detalles sobre MDRR en las colas FrFab (Tx)**

	Implementados	PQ alternativo de FrFab	PQ estricta FrFab	FrFab WRED
Eng0	Software	No	Yes	Yes
Eng1	-	No	No	No
Eng2	Hardware	Si <sup>2</sup>	Yes	Yes
Eng3	Hardware	No	Yes	Yes
Eng4	Hardware	Yes	Yes	Yes
Eng4+	Hardware	Yes	Yes	Yes

<sup>1</sup> El soporte para MDRR en colas FrFab de LCs de Motor 0 se introduce en estas versiones del software Cisco IOS:

- Cisco IOS Software Release 12.0(10)S - 4xOC3 y 1xOC12 POS, 4xOC3 y 1xCHOC12/STM4.
- Software Cisco IOS versión 12.0(15)S - 6xE3 y 12xE3.
- Versión 12.0(17)S del software del IOS de Cisco - 2xCHOC3/STM1.

<sup>2</sup> Debe configurar un MDRR alternativo en la dirección FrFab con la sintaxis de CoS heredada.

**Nota:** La LC 3xGE soporta MDRR en las colas ToFab y, a partir de Cisco IOS Software Release 12.0(15)S, en las colas FrFab con dos restricciones, a saber, una cantidad fija y una cola CoS única para cada interfaz. La cola de prioridad admite una cantidad que se puede configurar y los modos de prioridad estricta y alternativa. Las tres interfaces comparten una única PQ.

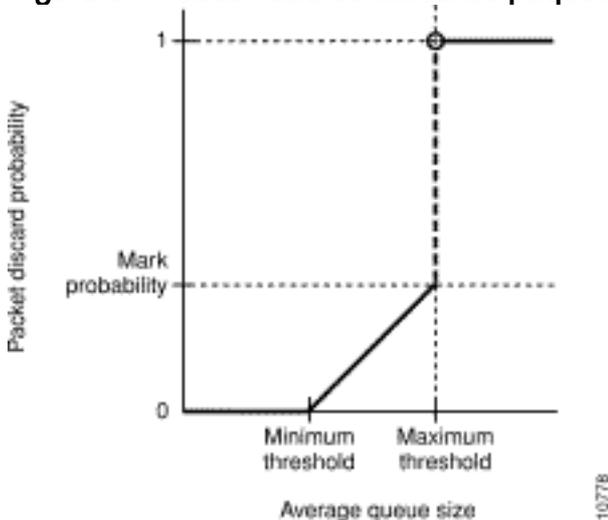
**Nota:** Consulte las notas de la versión de los routers de la serie 12000 de Cisco para obtener la

información más reciente sobre las funciones de QoS admitidas en las LC de la serie 12000 de Cisco.

## Información general sobre WRED

La detección temprana aleatoria ponderada (WRED) está diseñada para prevenir los efectos dañinos de la congestión de interfaz en el flujo de datos de la red.

Figura 6: Probabilidad de caída de paquetes WRED



Consulte [Detección temprana aleatoria ponderada en el Cisco 12000 Series Router](#) para obtener una explicación de los parámetros WRED. Los parámetros de probabilidad mínima, máxima y de marca describen la curva real de detección temprana aleatoria (RED). Cuando el promedio ponderado de la cola está por debajo del umbral mínimo, no se descartan paquetes. Cuando el promedio ponderado de cola está por encima del umbral máximo de cola, todos los paquetes se descartan hasta que el promedio cae por debajo del umbral máximo. Cuando el promedio se encuentra entre los umbrales mínimo y máximo, la probabilidad de que el paquete se descarte puede ser calculada por una línea recta desde el umbral mínimo (la probabilidad de caída será 0) hasta el umbral máximo (la probabilidad de caída es igual al denominador de probabilidad  $1/\text{marca}$ ).

La diferencia entre RED y WRED es que WRED puede descartar selectivamente el tráfico de menor prioridad cuando la interfaz comienza a congestionarse y puede proporcionar características de rendimiento diferenciadas para diferentes clases de servicio (CoS). De forma predeterminada, WRED utiliza un perfil RED diferente para cada peso (precedencia IP - 8 perfiles). Descarta los paquetes menos importantes de manera más agresiva que los paquetes más importantes.

Es un desafío complejo ajustar los parámetros WRED para administrar la profundidad de la cola y depende de muchos factores, entre los que se incluyen:

- Carga de tráfico y perfil ofrecidos.
- Relación de carga con la capacidad disponible.
- Comportamiento del tráfico en presencia de congestión.

Estos factores varían según la red y, a su vez, dependen de los servicios ofrecidos y de los clientes que los utilizan. Por lo tanto, no podemos hacer recomendaciones que se apliquen a entornos de clientes específicos. Sin embargo, en la [tabla 4](#) se describen los valores generalmente recomendados en función del ancho de banda del link. En ese caso, no

diferenciamos las características de eliminación entre las diferentes clases de servicios.

**Tabla 4: Valores recomendados basados en el ancho de banda del enlace**

Ancho de banda	BW teórico (kbps)	BW1 física (kbps)	Umbral mínimo	Umbral máximo
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

<sup>1</sup> Velocidad de SONET física

Existen varias limitaciones que se consideran al calcular los valores de umbral anteriores. Por ejemplo, la maximización de la utilización del link mientras se minimiza la profundidad promedio de la cola, la diferencia entre el máximo y el mínimo debe ser una potencia de dos (debido a la limitación del hardware). En función de la experiencia y simulación, la profundidad instantánea máxima de una cola controlada por RED es menor a 2 MaxTh. Para OC48 y superiores, 1 MaxTh y así sucesivamente. No obstante, la determinación exacta de estos valores está más allá del alcance de este documento.

**Nota:** No es necesario configurar el valor constante de ponderación exponencial en las tarjetas de línea del Motor 2 y superiores, ya que en su lugar se utiliza el muestreo de cola de hardware. Para las LC de Motor 0, estos valores se pueden configurar:

- ds3 - 9
- oc3 - 10
- oc12 - 12

**Nota:** WRED no se soporta en las LC del Motor 1.

Como se explica en las secciones siguientes, puede utilizar tanto la sintaxis heredada de CoS como la sintaxis más reciente de MQC para configurar WRED.

## [Utilizar sintaxis de CoS heredada para la configuración](#)

La sintaxis de la clase de servicio (CoS) heredada del Cisco 12000 usa una plantilla `cos-queue-group` para establecer un conjunto de definiciones de CoS. A continuación, aplica la plantilla a las colas ToFab y FrFab en las interfaces de entrada o de salida, respectivamente.

### [Paso 1: Definir un cos-queue-group](#)

El comando `cos-queue-group` crea una plantilla denominada de parámetros MDRR y WRED. Estos son los parámetros de configuración disponibles en la 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	Set RED drop criteria
traffic-shape	Enable Traffic Shaping on a COS queue group

Con MDRR, puede asignar la precedencia IP a las colas MDRR y configurar la cola especial de latencia baja. Puede utilizar el parámetro `precedence` bajo el comando `cos-queue-group` para esto:

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

Puede mapear una precedencia de IP específica a una cola MDRR regular (cola 0 a 6) o puede asignarla a la cola de prioridad. El comando anterior permite asociar varias precedencias IP con la misma cola.

**Nota:** Se recomienda utilizar la precedencia 5 para la cola de latencia baja. La precedencia 6 se utiliza para las actualizaciones del router.

Puede dar a cada cola MDRR un peso relativo, con una de las colas del grupo definida como cola de prioridad. Puede utilizar el comando `queue` bajo el `cos-queue-group` para hacer esto:

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

Use el comando `cos-queue-group` para definir cualquier parámetro WRED:

```
random-detect-label
```

Este es un ejemplo de un `cos-queue-group` llamado `oc12`. Define tres clases de tráfico: cola 0, 1 y baja latencia. Asigna los valores de precedencia IP 0 - 3 a la cola 0, los valores de precedencia 4, 6 y 7 a la cola 1 y la precedencia 5 a la cola de latencia baja. Se asigna mayor ancho de banda a la cola 1.

### Ejemplo de configuración

```
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.

```

## Paso 2 - Cree un slot-table-cos para colas ToFab

Para evitar el bloqueo de cabecera de línea, las interfaces entrantes de la serie Cisco 12000 mantienen una cola de salida virtual por ranura de destino. Vaya a una tarjeta de línea usando el comando attach y ejecute el comando execute-on show controller tofab queue (o ingrese directamente el comando execute-on slot 0 show controllers tofab queue) para ver estas colas de salida virtuales. A continuación se proporciona un ejemplo de salida capturada directamente desde la consola LC. Vea [Cómo Leer el Resultado del comando show controller tofab | Comandos tofab queue en un router de Internet de la serie 12000 de Cisco](#).

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#

Utilice el comando **slot-table-cos** para asignar un **cos-queue-group** con nombre a una cola de salida virtual de destino. Puede configurar una plantilla única **cos-queue-group** para cada cola de salida

```
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
```

**Nota:** La configuración anterior utiliza tres plantillas, denominadas oc48, oc12 y oc3. La configuración para el grupo **cos-queue-group** denominado oc12 es como se muestra en el Paso 1. Del mismo modo, configure oc3 y oc48. Se recomienda aplicar una plantilla única a un conjunto de interfaces basadas en el ancho de banda y la aplicación.

### [Paso 3 - Aplique un slot-table-cos a una interfaz entrante](#)

Use el comando **rx-cos-slot** para aplicar un **slot-table-cos** a un LC.

```
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
```

### [Paso 4 - Aplique un cos-queue-group a una interfaz saliente](#)

La serie 12000 de Cisco mantiene una cola independiente por interfaz saliente. Para ver estas colas, conecte a la CLI de la tarjeta de línea. Utilice el comando **attach** y luego ejecute el comando **show controller frfab queue**, como se ilustra aquí:

```
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
<b>Interface Queues:</b>				
0	0	0	0	65535
1	0	0	0	65535
2	0	0	0	65535
3	0	0	0	65535

Utilice el comando **tx-cos** para aplicar una **plantilla cos-queue-group** a una cola de interfaz. Como se muestra aquí, se aplica el conjunto de parámetros directamente a la interfaz; no se necesitan tablas. En este ejemplo, *pos48* es el nombre de un conjunto de parámetros.

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

Utilice el comando **show cos** para confirmar su configuración:

```
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
```

**Nota:** La CLI heredada también utiliza la sintaxis de precedencia para el tráfico de switching de etiquetas multiprotocolo (MPLS). El router trata los bits MPLS como si fueran bits de tipo de servicio (ToS) IP y pone los paquetes adecuados en las colas correctas. Esto no es cierto para MQC. MPLS QoS está fuera del alcance de este documento.

[\*\*Utilizar CLI de QoS modular \(MQC\) para la configuración\*\*](#)

El objetivo de la CLI de QoS modular (MQC) de Cisco es conectar todas las diferentes funciones de QoS de forma lógica, con el fin de simplificar la configuración de las funciones de calidad de servicio (QoS) del software Cisco IOS. Por ejemplo, la clasificación se realiza por separado de la colocación en cola, la regulación y el modelado. Proporciona un único marco de configuración para QoS basado en plantillas. Estos son algunos de los puntos a recordar sobre la configuración de MQC:

- Se puede aplicar y quitar fácilmente de una interfaz.
- Se puede reutilizar fácilmente (la misma política se puede aplicar a varias interfaces).
- Ofrece un único marco de configuración para QoS que le permite aprovisionar, supervisar y solucionar problemas fácilmente.
- Proporciona un mayor nivel de abstracción.
- Es independiente de la plataforma.

En la serie Cisco 12000, se pueden utilizar los comandos MQC en lugar de la sintaxis heredada de Clase de servicio (CoS).

La compatibilidad con MQC en la serie 12000 de Cisco no implica que el mismo conjunto de funciones de QoS disponible en otra plataforma, como la serie 7500 de Cisco, esté ahora disponible en el Cisco 12000. El MQC proporciona una sintaxis común en la que un comando produce una función o comportamiento compartidos. Por ejemplo, el comando **bandwidth** implementa una garantía de ancho de banda mínimo. La serie 12000 de Cisco utiliza MDRR como mecanismo de programación para hacer la reserva de ancho de banda, mientras que la serie 7500 de Cisco utiliza WFQ. El algoritmo principal complementa la plataforma en particular.

Es importante destacar que sólo las colas FrFab admiten la configuración de las funciones de QoS a través de MQC. Debido a que las colas ToFab son colas de salida virtuales y no colas de entrada verdaderas, MQC no las admite. Deben configurarse con comandos CoS heredados.

[La tabla 5](#) enumera el soporte para el MQC por tipo de motor L3.

**Tabla 5: Soporte de MQC para Tipos de Motor L3**

Tipo de motor L3	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4	Motor 4+
Soporte MQC.	Yes	No	Yes	Yes	Yes	Yes
Versión del IOS	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> Recuerde estas excepciones con compatibilidad con MQC en las tarjetas de línea (LC)s del Motor 0 y 2:

- 2xCHOC3/STM1 – Incorporado en 12.0(17)S.
- 1xOC48 DPT – Introducido en 12.0(18)S.
- 8xOC3 ATM - Planificado para 12.0(22)S.

El MQC utiliza estos tres pasos para crear una política de QoS:

1. Defina una o más clases de tráfico con el comando `class-map`.
2. Cree una política de calidad de servicio (QoS) con el comando `policy-map` y asigne acciones de QoS, tales como ancho de banda o prioridad a una clase de tráfico con nombre.
3. Utilice el comando **service-policy** para adjuntar un `policy-map` a la cola FrFab de una interfaz saliente.

Utilice el comando `show policy-map interface` para monitorear su política.

Consulte [Descripción General de la Interfaz de Línea de Comandos de Calidad de Servicio Modular](#) para obtener más información.

### [Paso 1 - Definir class-maps](#)

El comando **class-map** se utiliza para definir las clases de tráfico. Internamente, en la serie 12000 de Cisco, el comando **class-map** asigna una clase a una cola de CoS específica en la tarjeta de línea (consulte el [Paso 4](#) para obtener detalles).

El comando **class-map** soporta "match-any", que coloca los paquetes que coinciden con cualquiera de las sentencias match en la clase, y "match-all", que coloca los paquetes en esta clase solamente cuando todas las sentencias son verdaderas. Estos comandos crean una clase denominada "Prec\_5" y clasifican todos los paquetes con una precedencia IP de 5 en esta clase:

```
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
```

[En la tabla 6](#) se enumeran los criterios de coincidencia admitidos para cada tipo de motor L3.

**Tabla 6: Criterios de coincidencia admitidos para motores L3**

	Motor 0, 2	Motor 3	Motor 4	Motor 4+
precedencia ip	Yes	Yes	Yes	Sí <sup>1</sup>
access-group	No	Yes	No	No
mpls exp	No	Yes	No	Sí (12.0.26S)
ip dscp	No	Yes	No	Sí (12.0.26S)
qos-group	No	Yes	No	No

match input-interface POS x/y	No	Sí (únicamente como política de recepción)	No	No
-------------------------------	----	--	----	----

<sup>1</sup> ingreso/egreso desde 12.0.26S

## Paso 2 - Creación de un policy-map

El comando **policy-map** se utiliza para asignar políticas o acciones de administración de paquetes a una o más clases definidas. Por ejemplo, cuando asigna una reserva de ancho de banda o aplica un perfil de caída aleatorio.

La serie 12000 de Cisco admite un subconjunto de funciones MQC, basadas en la arquitectura de alta velocidad de los motores L3. [La tabla 7](#) enumera los comandos que se soportan:

**Tabla 7: Comandos admitidos**

Comando	Descripción
ancho de banda	Proporciona una garantía de ancho de banda mínimo durante períodos de congestión. Se especifica como porcentaje de la velocidad del link o como valor absoluto. Si una clase no utiliza o necesita un ancho de banda igual a los kbps reservados, otras clases de ancho de banda pueden utilizar el ancho de banda disponible.
police, shape	Limita la cantidad de tráfico que puede transmitir una clase. Estos comandos son ligeramente diferentes en función. El comando <b>police</b> identifica el tráfico que excede el ancho de banda configurado, y lo descarta o lo recalca. El comando <b>shape</b> almacena en búfer cualquier exceso de tráfico y lo programa para su transmisión a una velocidad constante, pero no descarta ni marca.
Límite de cola	Asigna una longitud fija de cola a una clase determinada de tráfico. Puede especificar esto en el número de paquetes que se pueden retener en la cola.
prioridad	Identifica una cola como cola de baja latencia. MQC soporta el modo estricto solamente para una PQ. El modo alternativo no se soporta a través de MQC. Utilice el comando <b>priority</b> sin un valor de porcentaje para habilitar el modo de prioridad estricta. <b>Nota:</b> La implementación del comando <b>priority</b> en la serie Cisco 12000 difiere de la implementación en otros routers que ejecutan

	<p>el software Cisco IOS. En esta plataforma, el tráfico de prioridad no se limita al valor de kbps configurado durante períodos de congestión. Por lo tanto, también debe configurar el comando <b>police</b> para limitar la cantidad de ancho de banda que puede utilizar una clase de prioridad y asegurar un ancho de banda adecuado para otras clases. En este momento, el comando <b>police</b> sólo se soporta en las tarjetas de línea del Motor 3. En las otras tarjetas de línea del motor, sólo se permite class-default cuando se configura una clase de prioridad.</p>
random-detect	<p>Asigna un perfil WRED. Utilice el comando <b>random-detect precedence</b> para configurar valores WRED no predeterminados por valor de precedencia IP.</p>

En las LC del Motor 3, debe configurar las colas FrFab con la CLI de QoS modular (MQC); no se admite la interfaz de línea de comandos (CLI) heredada.

Cuando configure el comando **bandwidth**, observe que las LC Engine 0 y 2 soportan solamente seis clases de ancho de banda. Una séptima clase se puede utilizar para el servicio de baja latencia y una octava clase, que es la clase predeterminada, se utiliza para todo el tráfico que no coincide. Por lo tanto, tiene un total de ocho colas. Class-default no se utiliza como clase de prioridad.

En las LC del Motor 3, el comando **bandwidth percent** se traduce en un valor de kbps, que varía con la velocidad de link subyacente, y luego se configura directamente en la cola. La precisión de esta garantía mínima del ancho de banda es 64 Kbps.

Aunque no se realiza ninguna conversión a un valor cuántico con el comando **bandwidth**, todas las colas tienen un valor cuántico. En las LC del Motor 3, el valor cuántico se establece internamente en base a la unidad máxima de transmisión (MTU) de la interfaz, y se configura equitativamente para todas las colas. No hay un mecanismo MQC CLI para modificar este valor determinado, ni directa ni indirectamente. El valor cuántico debe ser mayor o igual que la MTU de la interfaz. Internamente, el valor cuántico está en unidades de 512 bytes. Por lo tanto, con una MTU de 4470 bytes, el valor cuántico mínimo de la MTU debe ser 9.

### [MDRR en la LC del Motor 3](#)

Esta sección proporciona notas de configuración para implementar WRED y MDRR en las LC del Motor 3.

- El ancho de banda MDRR configurado en CLI se traduce a una cantidad correspondiente a L2 (por ejemplo, se elimina la sobrecarga L1). Esa cantidad luego se redondea en los próximos 64 kbps y se programa en el hardware.
- Tres perfiles de WRED diferentes son admitidos para una sola clase.
- El WRED (umbral máximo - umbral mínimo) se aproxima a la potencia más cercana de 2. A continuación, el umbral mínimo se ajusta automáticamente mientras que el umbral máximo se mantiene sin cambios.

- Es admitido el valor 1 de mark probability.
- No se admite la configuración constante de ponderación exponencial.
- Se soportan valores de precedencia IP, bits MPLS EXP y DSCP.

**Nota:** Cada puerto o canal en las tarjetas de línea Tetra (4GE-SFP-LC= ) o CHOC12/DS1-IR-SC= Frostbita tienen cuatro colas asignadas de forma predeterminada. Las cuatro colas se componen de lo siguiente:

- Una clase de cola de prioridad (LLQ)
- Una clase de cola predeterminada
- Dos clases no prioritarias normales

Cuando se aplica una política de servicio que contiene más de estas cuatro clases (1 HPQ, 2 LPQ y class-default) a la interfaz, se notificará el siguiente error:

```
Router(config-if)#service-policy output mdr-policy
```

**% No hay suficientes recursos de cola disponibles para satisfacer la solicitud.**

A partir de 12.0(26)S, se ha agregado un comando para la tarjeta de línea 4GE-SFP-LC= Tetra que permite la configuración de ocho colas/VLAN en lugar de cuatro. Las ocho colas se componen de lo siguiente:

- Un LLQ
- Una cola predeterminada de clase
- Seis colas normales

El uso de este comando requerirá una recarga de microcódigo de la tarjeta de línea y dará como resultado la capacidad de configurar sólo 508 VLAN en lugar de 1022. La sintaxis de los comandos es la siguiente:

```
[no] hw-module slot <slot#> qos interface queues 8
```

Por ejemplo:

```
Router(config)#hw-module slot 2 colas de interfaz qos 8
```

**Advertencia:** Haga un micro recargar la tarjeta de línea para que este comando entre en vigor

```
Router(config)#microcode reload 2
```

Este comando estará disponible para la tarjeta de línea CHOC12/DS1-IR-SC= Frostbit en 12.0(32)S

### Ejemplo n.º 1 - Comando bandwidth percent

Este ejemplo asigna el 20 por ciento del ancho de banda disponible al tráfico de clase Prec\_4 y el 30 por ciento al tráfico de clase Prec\_3. Deja el 50% restante a la clase predeterminada de clase.

Además, configura WRED como el mecanismo de descarte en todas las clases de datos.

<b>Ejemplo n.º 1: porcentaje de ancho de banda</b>
--

<pre>policy-map GSR_EXAMPLE</pre>
-----------------------------------

```

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

```

## Ejemplo n.º 2 - Comando bandwidth {kbps}

Este ejemplo ilustra cómo aplicar el comando bandwidth como un valor absoluto de kbps en lugar de un porcentaje.

### Ejemplo n.º 2: ancho de banda {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

```

## Ejemplo Nº 3: comando priority

Este ejemplo está diseñado para proveedores de servicios que utilizan el router de la serie 12000 de Cisco como router de extremo del proveedor MPLS (PE) y necesitan configurar una política de servicio de QoS en el enlace entre el router PE y el router de borde del cliente (CE). Coloca los paquetes de precedencia IP 5 en una cola de prioridad y limita la salida de esa cola a 64 Mbps. Luego asigna una parte del ancho de banda restante a las clases de ancho de banda.

Todas las colas de clase no prioritaria se configuran con el comando **random-detect** para habilitar WRED como política de caída. Todas las clases de ancho de banda y class-default deben tener WRED configurada explícitamente.

### Ejemplo n.º 3: prioridad

```

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

```

### Paso 3 - Asignación de un policy-map a una cola de interfaz saliente

Como se mencionó anteriormente, el MQC funciona solamente con las colas FrFab en una interfaz saliente. Para aplicar un policy-map definido, utilice el comando **service-policy output**, como se muestra aquí:

```

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

```

### Paso 4 - Supervisión y verificación de la política de servicio

Utilice el comando **show policy-map interface** para ver la aplicación de una política. El comando **show policy-map interface** muestra lo siguiente:

- Clases de ancho de banda y prioridad configuradas y criterios de coincidencia.
- Cualquier perfil WRED.
- Parámetros de forma y de regulación.
- Contabilidad y tarifas del tráfico.
- La cola de CoS interna a la que se asigna una clase determinada. Estas colas son referenciadas por el mismo índice que se utiliza en la salida del comando **show controller frfab queue**.

A continuación se muestra un ejemplo de una configuración completa y de los comandos **show** para monitorear la política:

#### **Configuración completada**

```

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
policy 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 policy !--- Step 3-

```

*Attach policy-map to the interface.*

Utilice el comando **show policy-map interface** para ver la política configurada en la interfaz, junto con todas las clases configuradas. Aquí está el resultado del comando:

```
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
```

## Comandos para supervisar la gestión de la congestión y la prevención

Esta sección enumera los comandos que puede utilizar para monitorear su política de prevención y administración de congestión.

[La tabla 8](#) enumera los comandos relevantes para las tarjetas de línea de ingreso y salida.

**Tabla 8: Comandos para las tarjetas de línea**

Tarjeta de línea de ingreso	Tarjeta de línea de egreso
<ul style="list-style-type: none"><li>• <b>show interfaces</b></li><li>• <b>exec slot &lt;x&gt; sh controller tofab queue</b></li></ul>	<ul style="list-style-type: none"><li>• <b>show interfaces</b></li><li>• <b>show interfaces &lt;y&gt; random</b></li></ul>

<ul style="list-style-type: none"> <li>• <b>exec slot &lt;x&gt; show controller tofab queue &lt;slot&gt; &lt;port&gt;</b></li> <li>• <b>exec slot &lt;x&gt; show controller tofab qm stat</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>exec slot &lt;y&gt; show controller frfab queue</b></li> <li>• <b>exec slot &lt;y&gt; show controller frfab queue &lt;port&gt;</b></li> <li>• <b>exec slot &lt;y&gt; show controller frfab QM stat</b></li> </ul>
--	---

Estos comandos se explican en esta sección.

## [El comando show interfaces](#)

Antes de utilizar este comando, confirme la "estrategia de colas" correcta. Si el resultado muestra First In, First Out (FIFO), asegúrese de que el comando **service-policy** aparezca en la configuración en ejecución (si se ha utilizado MQC para configurar MDRR).

Controle la cantidad de caídas de salida la cual representa la cantidad total de caídas WRED FrFab que han tenido lugar para el tráfico saliente en esta interfaz. El número de caídas de salida en la salida del comando **show interfaces** debe ser igual o superior al número de caídas de salida en la salida del comando **show interfaces <number> random**.

**Nota:** En el Cisco 12000 Series Router, las caídas de salida de la interfaz se actualizan después de actualizar las caídas WRED. Existe una pequeña posibilidad de que si utiliza una herramienta para consultar ambos contadores de caídas, las caídas de la interfaz todavía no se actualizan.

```
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
```

## [Comando show interfaces {number} random](#)

Cuando utiliza este comando, debe:

- Verifique que la plantilla correcta **cos-queue-group** se aplique a esta interfaz.
- Compruebe los pesos del MDRR. Para cada cola MDRR, puede verificar la media ponderada para la longitud de cola y el valor más alto alcanzado (en paquetes). Los valores se calculan como una media ponderada y no necesitan reflejar la profundidad máxima real de cola alcanzada.
- Verifique los umbrales WRED mínimos y máximos.
- Verifique la cantidad de caídas aleatorias y de umbral para cada etiqueta RED (Las caídas "A hacia el entramado" indican la cantidad total de caídas para esta etiqueta en todas las tarjetas de línea).
- El contador "TX-queue-limit drops" se utiliza solamente en las LC del Motor 1, que no soportan WRED. Las tarjetas de motor 1 le permiten establecer el límite de las colas MDRR con el comando **TX-queue-limit interface**. En los casos en que se admite WRED, sus umbrales determinan la profundidad de las colas MDRR.

```
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
<b>0</b>	<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
```

[Comando exec slot \(y\) show controller frfab queue {port}](#)

Este comando muestra la profundidad de cola instantánea para un puerto determinado en una ranura dada. El ejemplo de salida en esta sección muestra la cola MDRR en la interfaz POS 4/1. Verá una profundidad de cola para la cola MDRR 1 de los paquetes 1964. El peso es el número de bytes que se pueden servir en esta cola. Este peso determina el porcentaje de ancho de banda que desea asignar a esta cola. El déficit es el valor que indica al algoritmo DRR cuántos paquetes aún deben ser atendidos. Puede ver que no hay paquetes en cola en el LLQ (cola DRR 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
```

Este comando se utiliza, en particular, para monitorear la profundidad de la cola de prioridad de la tarjeta de línea de salida. Cuando ve que los paquetes comienzan a esperar en este LLQ, es una buena indicación de que debe desviar parte del tráfico de voz sobre IP (VOIP) a otras tarjetas de línea de salida. En un buen diseño, la longitud debe ser siempre 0 o 1. En una red real, experimentará un tráfico saturado, incluso para los datos de voz. El retardo adicional se torna más grave cuando la carga de voz total excede el 100% del ancho de banda de egreso durante un breve lapso de tiempo. El router no puede colocar más tráfico en el cable que el permitido; por lo tanto, el tráfico de voz se coloca en cola en su propia cola prioritaria. Esto crea latencia de voz y fluctuación de voz introducida por la ráfaga del propio tráfico de voz.

```
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
```

La cola 7 es la LLQ y la longitud le indica cuántos paquetes hay en esta LLQ.

## [Comando exec slot \(y\) show controller frfab QM stat](#)

Utilice este comando cuando sospeche que la memoria de paquete de una LC comienza a aproximarse a la capacidad total. Un valor en aumento para el contador "no mem drop" sugiere que WRED no está configurado o que los umbrales WRED están configurados demasiado altos. Este contador no debe incrementarse en condiciones normales. Para obtener más información,

consulte [Resolución de problemas de paquetes ignorados y caídas sin memoria en el router de Internet de la serie 12000 de Cisco](#).

```
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
```

## [Supervisar la administración de congestión entrante](#)

Esta sección describe los comandos utilizados para monitorear la administración de congestión entrante.

### [El comando show interfaces](#)

Antes de ejecutar este comando, verifique si el valor del contador ignorado está en aumento. Verá los paquetes ignorados si se queda sin memoria en el lado ToFab o si la tarjeta de línea no acepta los paquetes lo suficientemente rápido. Para obtener más información, consulte [Resolución de problemas de caídas de entrada en el router de Internet de la serie 12000 de Cisco](#).

```
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
```

## Comando exec slot (x) show controller tofab queue

Este ejemplo de salida del comando `exec slot <x> show controller tofab queue` se capturó cuando no había congestión en una tarjeta de línea de egreso en la ranura 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 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
```

El siguiente resultado se capturó cuando hubo congestión en la ranura 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
```

## Comando exec slot (x) show controller tofab queue (slot) (port)

Utilice este comando para determinar cuánta memoria se utiliza en el lado ToFab. En particular, observe el número en la columna '#Qelem'. Observe que:

- Cuando no se utiliza ninguna memoria, los valores se encuentran en su nivel más alto.
- El valor de la columna "#Qelem" disminuye a medida que los paquetes se almacenan en la memoria intermedia.
- Cuando la columna "#Qelem" llega a cero, todos los búfers divididos se encuentran en uso. En el motor 2 LC, los paquetes pequeños pueden obtener espacio de búfer prestado de paquetes más grandes.

También puede utilizar este comando para determinar el número de paquetes en cola en una cola de salida virtual. El ejemplo aquí muestra cómo verificar la ranura 14 para el número instantáneo de paquetes en estas colas para el puerto 1 del slot 4 (POS 4/1). Vemos 830 paquetes en cola en la cola 1 de MDRR.

```
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

## Comando exec slot (x) show controller tofab QM stat

Utilice este comando para ver la cantidad de caídas ToFab por tarjeta de línea. También verifique si hay un contador de "ausencia de memoria" que aumente. Este contador aumenta cuando la clase de servicio (CoS) no está configurada en el lado ToFab.

```
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
```

## Caso Práctico

Este caso práctico muestra cómo configurar una política típica para el núcleo de red de un entorno de proveedor de servicios. Aplica comandos de cola y le permite utilizar MDRR/WRED para la administración de cola activa. Las políticas de QoS en los routers de borde normalmente utilizan el marcado de tráfico, el acondicionamiento, etc., para permitir que los routers del núcleo clasifiquen el tráfico en clases basadas en valores de precedencia IP o punto de código DiffServ (DSCP). Este caso práctico utiliza las funciones de QoS del software Cisco IOS para cumplir los Acuerdos de nivel de servicio (SLA) estrictos y los diferentes niveles de servicio para los servicios de voz, vídeo y datos en la misma estructura básica IP.

En este enfoque, un proveedor de servicios ha implementado tres clases de tráfico. Lo más importante es el LLQ o la clase de c/.ola de tiempo de latencia bajo. Ésta es la clase para Voz y Video. Esta clase debe experimentar una demora y fluctuación mínimas, y nunca debe experimentar la pérdida de paquetes ni los paquetes reordenados siempre y cuando el ancho de banda de esta clase no exceda el ancho de banda del link. Esta clase se conoce como tráfico de Comportamiento por salto de reenvío acelerado (EF PHB) en la arquitectura DiffServ. El proveedor de servicios de Internet (ISP) diseñó la red de forma que esta clase no supere el 30% de la carga media del ancho de banda del enlace. Las otras dos clases son las de negocios y de esfuerzo razonable.

En el diseño, hemos configurado los routers de tal manera que la clase empresarial siempre obtiene el 90% del ancho de banda restante y la clase de mejor esfuerzo obtiene el 10%. Estas dos clases tienen menos tráfico sensible al tiempo y pueden experimentar pérdida de tráfico, mayor demora y fluctuación. En el diseño, el enfoque se centra en las tarjetas de línea del Motor 2: 1xOC48 rev B, 4xOC12 rev B y 8xOC3 tarjetas de línea.

Las tarjetas de línea Rev B son más adecuadas para transportar tráfico VoIP debido a una

arquitectura de hardware y ASIC revisada, que introduce muy poca latencia. Con el ASIC revisado, el controlador de la tarjeta de línea cambia el tamaño de la cola FIFO de transmisión a aproximadamente el doble de la MTU más grande de la tarjeta. Busque un "-B" agregado al número de pieza, como OC48E/POS-SR-SC-B=.

**Nota:** No confunda la cola FIFO de transmisión con las colas FrFab que se pueden ajustar en las tarjetas de línea del Motor 0 con el comando tx-queue-limit interface.

[La tabla 9](#) enumera los criterios coincidentes para cada clase.

**Tabla 9: Criterios de coincidencia para cada clase**

Nombre de clase	Criterios correspondientes
Cola prioritaria - Tráfico de Voz	Precedencia 5
Cola comercial	Precedencia 4
Mejor cola de esfuerzo	Precedencia 0

Las tarjetas de línea OC48 pueden colocar en cola una gran cantidad de paquetes en las colas ToFab. Por lo tanto, es importante configurar el MDRR/WRED en las colas ToFab, especialmente cuando la interfaz de egreso es una interfaz de alta velocidad como OC48. La estructura sólo puede conmutar tráfico a la tarjeta de línea receptora a una velocidad máxima hipotética de 3Gbps (paquetes de 1500 bytes). Si la cantidad total de tráfico enviado es mayor de lo que el entramado de conmutación puede llevar a su tarjeta receptora, muchos paquetes se pondrán en cola en las colas 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

```

Se espera que cuanto mayor sea el tráfico VOIP que tenga, mayor será el tráfico empresarial que tenga que esperar antes de que se le atienda. Sin embargo, esto no es un problema porque el SLA estricto no requiere descartes de paquetes, y muy baja latencia y fluctuación para la cola de prioridad.

## [Información Relacionada](#)

- [Cómo leer el resultado del comando show controller frfab | Comandos tofab queue en un router de Internet de la serie 12000 de Cisco](#)
- [Solución de problemas de paquetes ignorados e interrupciones por falta de memoria en el router de Internet de la serie Cisco 12000](#)
- [Resolución de problemas de caídas de entradas en el router de Internet de la serie 12000 de Cisco](#)
- [Detección temprana aleatoria ponderada en el router de Cisco serie 12000](#)
- [Descripción General de la Interfaz de Línea de Comandos de Calidad de Servicio Modular](#)
- [Página de Soporte de Routers de Internet de la Serie 12000](#)
- [Soporte Técnico - Cisco Systems](#)