

Entender e configurar o MDRR/WRED em 12000 Series Routers

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Informações de Apoio](#)

[Visão geral de MDRR](#)

[Fila de prioridade do MDRR](#)

[Exemplo de MDRR](#)

[Suporte MDRR por tipo de mecanismo](#)

[Visão geral do WRED](#)

[Usar sintaxe de CoS legada para configuração](#)

[Usar CLI de QoS modular \(MQC\) para configuração](#)

[Comandos para monitorar o gerenciamento e a prevenção de congestionamentos](#)

[O comando show interfaces](#)

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

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

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

[Monitorar gerenciamento de congestionamento de entrada](#)

[O comando show interfaces](#)

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

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

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

[Casos Práticos](#)

[Informações Relacionadas](#)

[Introduction](#)

Este documento avalia como configurar o gerenciamento de congestionamento do Cisco IOS® Software e os recursos para evitar congestionamento no Cisco 12000 Series Internet Router.

Depois de ler este documento, você deve ser capaz de:

- Entenda por que é importante configurar o Rodízio de Déficit Modificado (MDRR - Modified Deficit Round Robin) e a Detecção Antecipada Aleatória Ponderada (WRED - Weighted Random Early Detection) na sua rede central.

- Compreenda a implementação subjacente ao MDRR e ao WRED no Cisco 12000 Series.
- Configure MDRR e WRED usando a sintaxe de legado de CoS (Class of Service) e MQC (Modular QoS CLI).

Prerequisites

Requirements

Os leitores deste documento devem estar cientes destes tópicos:

- Uma familiaridade geral com a arquitetura do Cisco 12000 Series Internet Router.
- Em particular, uma percepção da arquitetura de enfileiramento e estes termos: ToFab (em direção à estrutura) - que descreve as filas do lado de recepção em uma placa de linha de entrada. FrFab (Da estrutura) - que descreve as filas do lado da transmissão em uma placa de linha de saída.

Observação: também é recomendável que você pesquise [como ler a saída do comando show controller frfab | Comandos tofab queue em um Cisco 12000 Series Internet Router](#).

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Todas as plataformas Cisco 12000, que incluem as plataformas 12008, 12012, 12016, 12404, 12406, 12410 e 12416.
- Software Cisco IOS versão 12.0(24)S1.

Observação: embora as configurações neste documento tenham sido testadas no Cisco IOS Software Release 12.0(24)S1, qualquer versão do Cisco IOS Software que suporte o Cisco 12000 Series Internet Router pode ser usada.

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.

Conventions

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

Informações de Apoio

Os métodos de enfileiramento definem o mecanismo de agendamento de pacotes ou a ordem na qual os pacotes são removidos da fila para a interface para transmissão no fio físico. Com base na ordem e no número de vezes que uma fila é atendida por uma função de agendador, os métodos de enfileiramento também suportam garantias mínimas de largura de banda e baixas latências.

É importante garantir que um mecanismo de programação de pacotes suporte a arquitetura de comutação na qual ele é implementado. O Weighted Fair Queuing (WFQ) é o conhecido algoritmo de programação para alocação de recursos em plataformas de roteadores da Cisco com uma

arquitetura baseada em barramento. No entanto, ele não é suportado no Cisco 12000 Series Internet Router. O enfileiramento de prioridade do software Cisco IOS tradicional e o enfileiramento personalizado também não são suportados. Em vez disso, o Cisco 12000 Series usa uma forma especial de enfileiramento chamada Rodízio de Déficit Modificado (MDRR - Modified Deficit Round Robin), que oferece garantias de largura de banda relativas, bem como uma fila de baixa latência. M de MDRR significa "modificado"; adiciona a fila de prioridade em comparação ao DRR onde nenhuma fila de prioridade está presente. Para obter detalhes sobre o MDRR, consulte a seção [Visão geral do MDRR](#).

Além disso, o Cisco 12000 Series suporta WRED (Detecção antecipada aleatória ponderada) como política de queda dentro das filas MDRR. Esse mecanismo de prevenção de congestionamento fornece uma alternativa ao mecanismo de queda traseira padrão. O congestionamento pode ser evitado por eliminações controladas.

Mecanismos de prevenção e gerenciamento de congestionamento, como WRED e MDRR, são particularmente importantes nas filas FrFab de interfaces de saída de velocidade relativamente baixa, como placas de linha canalizadas (LCs). A matriz de comutação de alta velocidade pode entregar pacotes para os grupos de canais muito mais rápido do que os grupos de canais podem transmitir. Como o enfileiramento/buffers são gerenciados no nível de porta física, a pressão de retorno em um canal pode afetar todos os outros canais nessa porta. Assim, é muito importante gerenciar essa contrapressão por meio do WRED/MDRR, que limita o impacto da contrapressão nos canais em questão. Para obter detalhes sobre como gerenciar a sobreassinatura da interface de saída, consulte [Troubleshooting de Ignored Packets and No Memory Drops no Cisco 12000 Series Internet Router](#).

[Visão geral de MDRR](#)

Esta seção fornece uma visão geral do Rodízio de Déficit Modificado (MDRR - Modified Deficit Round Robin).

Com o MDRR configurado como a estratégia de enfileiramento, as filas não vazias são atendidas uma após a outra, de modo round-robin. Cada vez que uma fila é servida, uma quantidade fixa de dados é removida da fila. O algoritmo então atende à próxima fila. Quando uma fila é servida, o MDRR controla o número de bytes de dados que foram removidos da fila além do valor configurado. Na próxima passagem, quando a fila for atendida novamente, menos dados serão removidos da fila para compensar o excesso de dados que foram atendidos anteriormente. Como resultado, a quantidade média de dados removidos da fila por fila estará próxima do valor configurado. Além disso, o MDRR mantém uma fila prioritária que é atendida de forma preferencial. O MDRR é explicado com mais detalhes nesta seção.

Cada fila dentro do MDRR é definida por duas variáveis:

- Valor quântico - Esse é o número médio de bytes servidos em cada rodada.
- Contador de déficit - É usado para rastrear quantos bytes uma fila transmitiu em cada rodada. É inicializado para o valor quântico.

Os pacotes em uma fila são servidos desde que o contador de déficit seja maior que zero. Cada pacote servido reduz o contador de déficit de um valor igual a seu comprimento em bytes. Uma fila não poderá mais ser atendida depois que o contador de déficit ficar zerado ou negativo. Em cada nova rodada, o contador de déficit de cada fila não vazia é aumentado pelo seu valor quântico.

Observação: em geral, o tamanho quântico para uma fila não deve ser menor que a MTU (Maximum Transmission Unit, unidade máxima de transmissão) da interface. Isso garante que o agendador sempre atenda a pelo menos um pacote de cada fila não vazia.

Um peso relativo pode ser atribuído a cada fila de MDRR, com uma das filas no grupo definida como uma fila de prioridade. Os pesos atribuem largura de banda relativa para cada fila quando a interface está congestionada. O algoritmo MDRR retira os dados de cada fila de forma round-robin se houver dados na fila a ser enviada.

Se todas as filas MDRR regulares possuírem dados, elas obterão serviços da seguinte maneira:

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

Durante cada ciclo, uma fila pode retirar um quantum com base no peso configurado. Nas placas de linha Engine 0 e Engine 2, um valor de 1 é equivalente a dar à interface um peso de sua MTU. Para cada incremento acima de 1, o peso da fila aumenta em 512 bytes. Por exemplo, se o MTU de uma determinada interface é 4470 e o peso de uma fila for configurado para 3, a cada tempo através da rotação, é permitido que $4470 + (3-1)*512 = 5494$ bytes sejam desenfileirados. Se duas filas DRR normais, Q0 e Q1, forem usadas, Q0 será configurado com um peso de 1 e Q1 será configurado com um peso de 9. Se ambas as filas estiverem congestionadas, de cada vez na rotação, Q0 teria permissão para enviar 4470 bytes e Q1 teria permissão para enviar $[4470 + (9-1)*512] = 8566$ bytes. Isso daria ao tráfego que vai para Q0 aproximadamente 1/3 da largura de banda, e ao tráfego que passa por Q1 cerca de 2/3 da largura de banda.

Observação: a fórmula padrão de desenfileiramento usada para calcular a atribuição de largura de banda de MDRR é $D = MTU + (weight-1)*512$. Em versões anteriores ao software Cisco IOS versão 12.0(21) S/ST, as placas de linha Engine 4 usavam uma fórmula de fila diferente. Para configurar o peso de MDRR para uma atribuição de largura de banda correta, certifique-se de executar uma versão do Cisco IOS Software posterior a 12.0(21) S/ST.

Observação: a fórmula quântica para as placas de linha Engine 4+ é (para a direção toFab, isso não é válido para FrFab) $Quantum = BaseWeight + \{BaseWeight * (QueueWeight - 1) * 512\} / MTU$. O BaseWeight é obtido com esta fórmula: $PesoBase = \{(MTU + 512 - 1) / 512\} + 5$

Nota: Todos os cálculos são arredondados; ou seja, todos os decimais são ignorados.

Observação: para saber se uma placa de linha de mecanismo específica suporta MDRR, consulte [Suporte MDRR por Tipo de Mecanismo](#).

[Fila de prioridade do MDRR](#)

O Cisco 12000 Series suporta uma fila de prioridade (PQ) dentro do MDRR. Essa fila fornece o atraso baixo e o atraso baixo de sincronismo exigidos pelo tráfego sensível ao tempo, como Voz sobre IP (VoIP).

Conforme observado acima, a Série Cisco 12000 não suporta o enfileiramento moderado ponderado (WFQ). Sendo assim, o PQ dentro de MDRR opera de maneira diferente do recurso de enfileiramento de baixa latência (LLQ) do software Cisco IOS, disponível para outras plataformas.

Uma diferença importante é como o agendador MDRR pode ser configurado para atender ao PQ em um dos dois modos, conforme listado na [tabela 1](#):

Tabela 1 - Como configurar o agendador MDRR para atender ao PQ em dois modos

	Alternate Mode (Modo Alternado)	Modo de prioridade estrita
Vantagens	Aqui, o PQ é atendido entre as outras filas. Em outras palavras, o agendador MDRR serve, alternativamente, o PQ e quaisquer outras filas configuradas.	Aqui o PQ é atendido sempre que não está vazio. Isso fornece o menor retardo possível para correspondência de tráfego.
Desvantagens	Esse modo pode introduzir instabilidade e retardo quando comparado ao modo de prioridade estrita.	Esse modo pode sobrecarregar outras filas, especialmente se os fluxos correspondentes forem remetentes agressivos.

O modo alternativo pode exercer menos controle sobre o jitter e o atraso. Se o agendador MDRR começar a atender quadros de tamanho MTU de uma fila de dados e, em seguida, um pacote de voz chegar ao PQ, o agendador em modo alternativo atenderá completamente à fila de não prioridade até que seu contador de déficit chegue a zero. Durante esse período, o PQ não é atendido e os pacotes VoIP são atrasados.

Por outro lado, no modo de prioridade estrita, o agendador atende apenas o pacote não prioritário atual e, em seguida, muda para o PQ. O agendador começa a atender a uma fila sem prioridade somente depois que o PQ fica completamente vazio.

É importante observar que a fila de prioridade no modo de prioridade alternativa é atendida mais de uma vez em um ciclo e, portanto, requer mais largura de banda do que outras filas com o mesmo peso nominal. Quanto mais é uma função de quantas filas são definidas. Por exemplo, com três filas, a fila de baixa latência é atendida duas vezes mais frequentemente que as outras filas e envia o dobro de peso por ciclo. Se oito filas forem definidas, a fila de latência baixa será atendida sete vezes mais frequentemente e o peso efetivo será sete vezes maior. Dessa forma, a largura de banda que a fila pode tomar está relacionada à frequência com que é atendida por round robin, que por sua vez depende de quantas filas são definidas como um todo. No modo de prioridade alternativa, a fila de prioridade é normalmente configurada com um peso pequeno para esse motivo específico.

Como exemplo, suponha que quatro filas estejam definidas: 0, 1, 2 e a fila de prioridade. No modo de prioridade alternativa, se todas as filas estiverem congestionadas, elas serão atendidas da seguinte maneira: **0, llq, 1, llq, 2, llq, 0, llq, 1,** em que llq significa fila de latência baixa.

Cada vez que uma fila é ocupada, ela pode enviar até o peso configurado para ela. Portanto, a largura de banda mínima que a fila de baixa latência pode ter é de:

- WL = peso da fila de baixa latência.
- W0, W1, ... Wn = pesos das filas DRR regulares.
- n = número de filas de DRR regulares usadas para esta interface.
- BW = Largura de banda do link.

Para o modo de prioridade alternada, a largura de banda mínima da fila de latência mínima = BW

$$* n * WL / (n * WL + \text{Sum}(W0, Wn)).$$

O peso é o único parâmetro de variável no MDRR que pode ser configurado. Ele influencia a quantidade relativa de largura de banda que uma classe de tráfego pode usar e a quantidade de tráfego enviado de uma vez. A utilização de pesos mais elevados significa que o ciclo global demora mais e, possivelmente, aumenta a latência.

Diretrizes de configuração

- É melhor configurar o peso da classe que tem o requisito de largura de banda mais baixo para 1 para manter o atraso e o jitter o mais baixo possível entre as outras classes.
- Selecione os valores de peso mais baixos possíveis. Comece com um peso de 1 para a classe com a menor largura de banda. Por exemplo, quando você usa duas classes com 50% de largura de banda para cada classe, você deve configurar 1 e 1. Não faz sentido usar 10 e 10, porque não há impacto no desempenho quando você escolhe 1. Além disso, um peso maior apresenta mais tremulação.
- Um valor de baixo peso para LLQ é muito importante, especialmente no modo alternativo para não adicionar muito atraso ou jitter às outras classes.

Exemplo de MDRR

O exemplo nesta seção é extraído da *Inside Cisco IOS® Software Architecture*, Cisco Press.

Suponha que tenhamos três filas:

- **Fila 0** - tem um quantum de 1500 bytes; É a fila de baixa latência, configurada para operar em modo alternado.
- **Fila 1** - tem um quantum de 3000 bytes.
- **Fila 2** - tem um quantum de 1500 bytes.

[A Figura 1](#) ilustra o estado inicial das filas, juntamente com alguns pacotes que foram recebidos e enfileirados.

Figura 1: Estado inicial do 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

A fila 0 é servida primeiro, seu quantum é adicionado ao contador de déficit, o Pacote 1, que é de 250 bytes, é transmitido e seu tamanho é subtraído do contador de déficit. Como o contador de déficit da fila 0 ainda é maior que 0 (1500 - 250 = 1250), o pacote 2 também é transmitido e seu comprimento subtraído do contador de déficit. O contador de déficit da fila 0 agora é -250, portanto, a fila 1 será atendida em seguida. [A Figura 2](#) indica esse estado.

Figura 2: Estado subsequente do 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

O contador de déficit da fila 1 é definido como 3000 ($0 + 3000 = 3000$) e os pacotes 4 e 5 são transmitidos. Com cada pacote transmitido, subtraia o tamanho do pacote do contador de déficit, para que o contador de déficit da fila 1 seja reduzido para 0. [A Figura 3](#) ilustra esse estado.

Figura 3: Estado MDRR quando o contador de déficit da fila 1 é zero

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

Precisamos retornar do modo de prioridade alternativa para a fila de serviço 0. Mais uma vez, o quantum é adicionado ao contador do déficit atual e o contador do déficit da fila 0 é ajustado ao resultado ($-250 + 1500 = 1250$). O pacote 3 é transmitido agora porque o contador de déficit é maior que 0 e a fila 0 está vazia agora. Quando uma fila é esvaziada, o contador de déficit é configurado para 0, como mostrado na Figura 4.

Figura 4: Estado MDRR quando uma fila está vazia

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

A fila 2 é servida em seguida; seu contador de déficit está definido como 1500 ($0 + 1500 = 1500$). Os pacotes 7 a 10 são transmitidos, o que deixa o contador de déficit em 500 ($1500 - (4 \times 250) = 500$). Como o contador de déficit ainda é maior que 0, o pacote 11 também é transmitido.

Quando o pacote 11 é transmitido, a fila 2 está vazia e seu contador de déficit definido como 0, como mostra a Figura 5.

Figura 5: Estado MDRR quando o pacote 11 é transmitido

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

A fila 0 é atendida novamente em seguida (porque estamos no modo de prioridade alternada). Como está vazio, servimos a fila 1 em seguida e transmitimos o pacote 6.

[Suporte MDRR por tipo de mecanismo](#)

O Cisco 12000 Series é compatível com cinco modelos de placas de linha com arquiteturas de Engine de encaminhamento da Camada 3 (L3). O suporte para MDRR varia com o tipo de mecanismo L3 e com o tipo de placa. Por exemplo, não há suporte para MDRR e WRED em placas de linha ATM Engine 0. Use o comando show diag para determinar o tipo de L3 Engine das placas de linha instaladas.

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

[MDRR em filas ToFab \(Rx\)](#)

Você pode usar a "Sintaxe de CoS legada" ou a "Interface de linha de comando de QoS modular" para configurar o MDRR na série Cisco 12000. As seções posteriores deste documento discutem como configurar o MDRR com CoS legado ou QoS modular. Você deve configurar as filas ToFab com a sintaxe de CoS legada somente porque elas não suportam a sintaxe mais recente do MQC. Consulte a [tabela 2](#) para obter detalhes.

Tabela 2 - Detalhes sobre MDRR em filas ToFab (Rx)

	Onde foram implementados	ToFab MDRR	PQ alternativo para ToFab	ToFab Strict PQ	ToFab WRED
Eng0	Software	Não**	Não**	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
Eng4+	Hardware	Yes	Yes	Yes	Yes

	e				
--	---	--	--	--	--

** O MDRR é suportado no Mecanismo 0 LCs na direção ToFab (Rx), mas somente no modo prioridade estrita, não no modo prioridade alternativa. As sete filas restantes são suportadas normalmente.

As interfaces de entrada mantêm uma fila de saída virtual separada por LC de destino. A maneira como essas filas são implementadas depende do tipo de Engine L3.

- Engine 0 - As LCs de entrada mantêm oito filas por slot de destino. Portanto, o número máximo de filas é $16 \times 8 = 128$. Cada fila pode ser configurada separadamente.
- Engines 2, 3, 4 e 4+ – LCs de entrada mantêm oito filas por interface de destino. Com 16 slots de destino e 16 interfaces por slot, o número máximo de filas é $16 \times 16 \times 8 = 2048$. Todas as interfaces em um slot de destino devem usar os mesmos parâmetros.

[MDRR em Filas FrFab \(Tx\)](#)

O MDRR nas filas FrFab opera de forma consistente, seja implementado em hardware ou software. Todos os tipos de mecanismo L3 suportam oito filas de classe para cada interface de saída. Consulte a [tabela 3](#) para obter detalhes.

Tabela 3 - Detalhes sobre MDRR em filas FrFab (Tx)

	Onde foram implementados	PQ alternativo de FrFab	PQ Rígido para FrFab	FrFab WRED
Eng0	Software ¹	No	Yes	Yes
Eng1	-	No	No	No
Eng2	Hardware	Sim ²	Yes	Yes
Eng3	Hardware	No	Yes	Yes
Eng4	Hardware	Yes	Yes	Yes
Eng4+	Hardware	Yes	Yes	Yes

¹ O suporte para MDRR em filas FrFab de LCs Engine 0 é apresentado nestas versões do software Cisco IOS:

- Software Cisco IOS versão 12.0(10)S - 4xOC3 e 1xOC12 POS, 4xOC3 e 1xCHOC12/STM4.
- Software Cisco IOS versão 12.0(15)S - 6xE3 e 12xE3.
- Software Cisco IOS versão 12.0(17)S - 2xCHOC3/STM1.

² Você deve configurar MDRR alternativo na direção FrFab com a sintaxe de CoS legada.

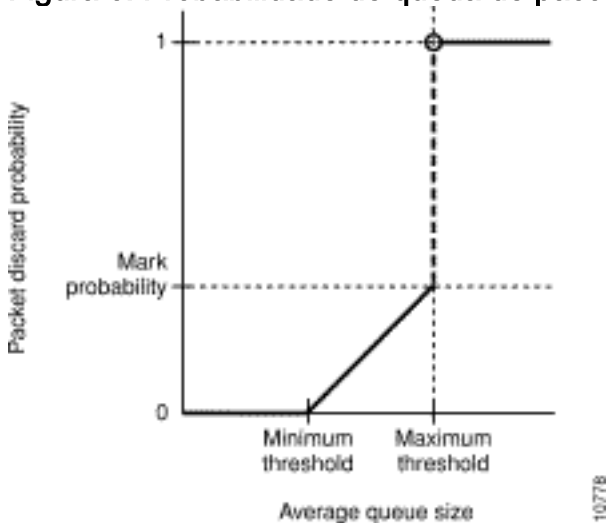
Observação: o LC 3xGE suporta MDRR nas filas ToFab e, a partir do Cisco IOS Software Release 12.0(15)S, nas filas FrFab com duas restrições, a saber, um quantum fixo e uma única fila de CoS para cada interface. A fila de prioridade suporta um quantum que pode ser configurado e os modos de prioridade estrita e alternativa. Todas as três interfaces compartilham um único PQ.

Observação: consulte as Notas de versão dos Cisco 12000 Series Routers para obter as informações mais recentes sobre os recursos de QoS suportados nos Cisco 12000 Series LCs.

Visão geral do WRED

A WRED (Detecção antecipada aleatória ponderada) foi desenvolvida para impedir os efeitos nocivos da congestão da interface no throughput da rede.

Figura 6: Probabilidade de queda de pacote WRED



Consulte [Detecção Antecipada Aleatória Ponderada no Cisco 12000 Series Router](#) para obter uma explicação dos parâmetros WRED. Os parâmetros de probabilidade mínima, máxima e de marca descrevem a curva real da Detecção Antecipada Aleatória (RED - Random Early Detection). Quando a média ponderada da fila estiver abaixo do limite mínimo, nenhum pacote será descartado. Quando a média ponderada da fila estiver acima do limite máximo da fila, todos os pacotes serão descartados até que a média caia abaixo do limite máximo. Quando a média está entre os limiares mínimo e máximo, a probabilidade de o pacote ser descartado pode ser calculada por uma linha reta desde o limite mínimo (a probabilidade de queda será 0) até o limite máximo (a probabilidade de queda é igual ao denominador de probabilidade de 1/marca).

A diferença entre o VERMELHO e o WRED é que o WRED pode descartar seletivamente o tráfego de prioridade mais baixa quando a interface começa a ficar congestionada e pode fornecer características de desempenho diferenciadas para diferentes classes de serviço (CoS). Por padrão, o WRED usa um perfil RED diferente para cada peso (precedência de IP - 8 perfis). Ele desconecta pacotes menos importantes de forma mais agressiva do que pacotes mais importantes.

É um desafio complexo ajustar os parâmetros WRED para gerenciar a profundidade da fila e depende de muitos fatores, que incluem:

- Carga de tráfego e perfil oferecidos.
- Relação entre carga e capacidade disponível.
- Comportamento do tráfego na presença de congestionamento.

Esses fatores variam de rede para rede e, por sua vez, dependem dos serviços oferecidos e dos clientes que usam esses serviços. Assim, não podemos fazer recomendações que se aplicam a ambientes específicos do cliente. No entanto, a [tabela 4](#) descreve os valores geralmente recomendados com base na largura de banda do link. Nesse caso, não diferenciamos as características da queda segundo as diferentes classes de serviço.

Tabela 4 - Valores recomendados com base na largura de banda do link

Largura	Largura	BW1	Limite	Limiar
---------	---------	-----	--------	--------

de banda	de Banda Teórica (kbps)	Físico (kbps)	mínimo	Máximo
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

¹ Taxa de SONET física

Existem várias restrições que são levadas em conta quando se calculam os valores limiares acima. Por exemplo, a maximização da utilização do link enquanto minimiza a profundidade média da fila, a diferença entre o Máximo e o Mínimo deve ser uma potência de dois (devido à limitação do hardware). Com base na experiência e na simulação, a profundidade instantânea máxima de uma filha controlada por RED é menor que 2 MaxTh. Para o OC48 ou maior, 1 MaxTh e assim por diante. No entanto, a determinação exata desses valores está além do escopo deste documento.

Observação: o valor da constante de ponderação exponencial não precisa ser configurado nas placas de linha Engine 2 e acima, já que a amostragem da fila de hardware é usada. Para LCs do Engine 0, esses valores podem ser configurados:

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

Observação: o WRED não é suportado nos LCs do Engine 1.

Como as seções a seguir explicam, você pode usar a sintaxe de CoS legada e a sintaxe de MQC mais recente para configurar o WRED.

Usar sintaxe de CoS legada para configuração

A sintaxe de Classe de serviço (CoS) herdada do Cisco 12000 Series utiliza um modelo cos-queue-group para definir um conjunto de definições de CoS. Em seguida, você aplica o modelo às filas ToFab e FrFab em interfaces de entrada ou saída, respectivamente.

Passo 1: Definir um cos-queue-group

O comando **cos-queue-group** cria um modelo nomeado de parâmetros MDRR e WRED. Aqui estão os parâmetros de configuração disponíveis na 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

Com o MDRR, você pode mapear a precedência de IP para filas MDRR e configurar a fila especial de baixa latência. Você pode usar o parâmetro de precedência no comando **cos-queue-group** para isso:

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

Você pode mapear uma precedência de IP particular para uma fila MDRR comum (fila de 0 a 6) ou pode mapeá-la para uma fila de prioridade. O comando acima permite que você mapeie diversas precedências de IP na mesma fila.

Observação: é recomendável usar a precedência 5 para a fila de baixa latência. A Precedência 6 é usada para atualizações de roteamento.

Você pode atribuir um peso relativo a cada fila MDRR, com uma das filas do grupo definidas como uma fila de prioridade. Você pode usar o comando **queue** sob o **cos-queue-group** para fazer isso:

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

Use o comando **cos-queue-group** para definir os parâmetros da WRED:

```
random-detect-label
```

Aqui está um exemplo de um **cos-queue-group** chamado **oc12**. Ele define três classes de tráfego: fila 0, 1 e baixa latência. Ele mapeia os valores de precedência de IP 0 - 3 para a fila 0, os valores de precedência 4, 6 e 7 para a fila 1 e a precedência 5 para a fila de baixa latência. Mais largura de banda foi atribuída à fila 1.

Exemplo de configuração

```
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.

```

Etapa 2 - Criar um slot-table-cos para as filas ToFab

Para evitar bloqueio de cabeçalho de linha, as interfaces de entrada no Cisco 12000 Series mantêm uma fila de saída virtual por slot de destino. Vá para uma placa de linha usando o comando attach e execute o comando execute-on show controller tofab queue (ou insira diretamente o comando execute-on slot 0 show controllers tofab queue) para ver essas filas de saída virtuais. A saída de exemplo capturada diretamente do console LC é fornecida abaixo. Consulte [Como ler a saída do comando show controller tofab | Comandos tofab queue em um Cisco 12000 Series Internet Router](#).

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
Multicast	0	0	0	65535

LC-Slot1#

Use o comando **slot-table-cos** para mapear um **cos-queue-group** nomeado para uma fila de saída virtual de destino. Você pode configurar um modelo **cos-queue-group** exclusivo para cada fila de saída

```
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
```

Observação: a configuração acima usa três modelos, chamados oc48, oc12 e oc3. A configuração para o **cos-queue-group** chamado oc12 é como mostrado na Etapa 1. Da mesma forma, configure o oc3 e o oc48. É recomendável aplicar um modelo exclusivo a um conjunto de interfaces com base na largura de banda e no aplicativo.

[Etapa 3 - Aplicar um slot-table-cos a uma interface de entrada](#)

Use o comando **rx-cos-slot** para aplicar um **slot-table-cos** a um 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
```

[Etapa 4 - Aplicar um cos-queue-group a uma interface de saída](#)

O Cisco 12000 Series mantém uma fila separada por interface de saída. Para visualizar essas filas, conecte-se à CLI da placa de linha. Use o comando **attach** e execute o comando **show controller frfab queue**, conforme ilustrado aqui:

```
LC-Slot1#show controller frfab queue
===== Line Card (Slot 2) =====
Carve information for FrFab buffers
SDRAM size: 16777216 bytes, address: 20000000, carve base: 2002D100
```

16592640 bytes carve size, 0 SDRAM bank(s), 0 bytes SDRAM pagesize, 2 carve(s)
max buffer data size 9248 bytes, min buffer data size 80 bytes
20052/20052 buffers specified/carved
16581552/16581552 bytes sum buffer sizes specified/carved

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

Use o comando **tx-cos** para aplicar um modelo **cos-queue-group** a uma fila de interface. Como mostrado aqui, você aplica o conjunto de parâmetros diretamente à interface; nenhuma tabela é necessária. Neste exemplo, *pos48* é o nome de um conjunto de parâmetros.

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

Use o comando **show cos** para confirmar sua configuração:

```
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
```

Observação: o CLI legado também usa a sintaxe de precedência para o tráfego Multiprotocol Label Switching (MPLS). O roteador trata os bits MPLS como se fossem bits do Tipo de Serviço (ToS - IP Type of Service) e coloca os pacotes apropriados nas filas corretas. Isso não é totalmente verdadeiro para o MQC. A QoS MPLS está fora do escopo deste documento.

[Usar CLI de QoS modular \(MQC\) para configuração](#)

O objetivo da CLI QoS modular (MQC) da Cisco é conectar todos os diferentes recursos de QoS

de forma lógica, a fim simplificar a configuração dos recursos de Qualidade de Serviço (QoS) do software Cisco IOS. Por exemplo, a classificação é feita separadamente do enfileiramento, da vigilância e da modelagem. Ele fornece uma estrutura de configuração única para QoS baseada em modelo. Aqui estão alguns pontos a serem lembrados sobre a configuração de MQC:

- Ele pode ser facilmente aplicado e removido de uma interface.
- Ela pode ser facilmente reutilizada (a mesma política pode ser aplicada a várias interfaces).
- Ele oferece uma única estrutura de configuração para QoS que permite provisionar, monitorar e solucionar problemas com facilidade.
- Ele fornece um nível mais alto de abstração.
- É independente de plataforma.

No Cisco 12000 Series, os comandos MQC podem ser usados em vez da sintaxe de Classe de Serviço (CoS - Class of Service) herdada.

O suporte de MQC na série Cisco 12000 não implica que o mesmo conjunto de recursos de QoS disponível em outra plataforma, como a série Cisco 7500, esteja agora disponível no Cisco 12000. O MQC fornece uma sintaxe comum na qual um comando resulta em uma função ou comportamento compartilhado. Por exemplo, o comando **bandwidth** implementa uma garantia de largura de banda mínima. O Cisco 12000 Series usa MDRR como mecanismo de agendamento para fazer a reserva de largura de banda, enquanto o Cisco 7500 Series usa WFQ. O algoritmo principal complementa a plataforma específica.

O importante é que apenas as filas FrFab suportam a configuração de recursos de QoS através do MQC. Como as filas ToFab são filas de saída virtuais, e não filas de entrada verdadeiras, elas não são suportadas pelo MQC. Eles devem ser configurados com comandos CoS legados.

[A Tabela 5](#) lista o suporte para o MQC por tipo de mecanismo L3.

Tabela 5 - Suporte para MQC para tipos de mecanismo L3

Tipo de L3 Engine	Mecanismo 0	Mecanismo 1	Mecanismo 2	Mecanismo 3	Mecanismo 4	Engine 4+
Suporte a MQC	Yes	No	Yes	Yes	Yes	Yes
Versão do IOS	12.0(15)S	-	12.0(15)S ¹	12.0(21)S	12.0(22)S	12.0(22)S

¹ Lembre-se destas exceções com suporte MQC em placas de linha (LCs) Engine 0 e 2:

- 2xCHOC3/STM1 - Introduzido na versão 12.0(17)S.
- 1xOC48 DPT introduzido na versão 12.0(18)S.
- 8xOC3 ATM Planejado para 12.0(22)S.

O MQC usa estas três etapas para criar uma política de QoS:

1. Defina uma ou mais classes de tráfego com o comando **class-map**.
2. Crie uma política de QoS com o comando **policy-map** e atribua ações de QoS, como largura de banda ou prioridade, a uma classe de tráfego nomeada.
3. Use o comando **service-policy** para anexar um mapa de política à fila FrFab de uma interface de saída.

Use o comando **show policy-map interface** para monitorar sua política.

Consulte [Visão Geral da Interface de Linha de Comando da Qualidade de Serviço Modular](#) para obter mais informações.

Etapa 1 - Definir mapas de classe

O comando **class-map** é usado para definir classes de tráfego. Internamente, no Cisco 12000 Series, o comando **class-map** atribui uma classe a uma fila CoS específica na placa de linha (consulte a [Etapa 4](#) para obter detalhes).

O comando **class-map** suporta "match-any", que coloca pacotes que correspondem a qualquer uma das instruções de correspondência na classe, e "match-all", que coloca pacotes nessa classe somente quando todas as instruções são verdadeiras. Esses comandos criam uma classe chamada "MOL_5" e classificam todos os pacotes com uma precedência de IP de 5 para esta classe:

```
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
```

[A Tabela 6](#) lista os critérios de correspondência suportados para cada tipo de Mecanismo L3.

Tabela 6 - Critérios de correspondência suportados para mecanismos L3

	Mecanismo o 0, 2	Mecanismo o 3	Mecanismo o 4	Engine 4+
precedência de ip	Yes	Yes	Yes	Sim ¹
access-group	No	Yes	No	No
mpls exp	No	Yes	No	Sim (12.0.26S)
ip dscp	No	Yes	No	Sim (12.0.26S)
qos-group	No	Yes	No	No
match input-interface POS x/y	No	Sim (somente como política de	No	No

		recepção)		
--	--	-----------	--	--

¹ entrada/saída desde 12.0.26S

Etapa 2 - Criar um mapa de políticas

O comando **policy-map** é usado para atribuir políticas ou ações de tratamento de pacotes a uma ou mais classes definidas. Por exemplo, quando você atribui uma reserva de largura de banda ou aplica um perfil de queda aleatório.

A série Cisco 12000 suporta um subconjunto de recursos MQC, com base na arquitetura de alta velocidade dos mecanismos L3. [A Tabela 7](#) lista os comandos que são suportados:

Tabela 7 - Comandos suportados

Comando	Descrição
largura de banda	Fornecer uma garantia de largura de banda mínima durante períodos de congestionamento. É especificado como uma porcentagem da velocidade do link ou como um valor absoluto. Se uma classe não usar ou precisar de largura de banda igual aos kbps reservados, a largura de banda disponível poderá ser usada por outras classes de largura de banda.
police, shape	Limita a quantidade de tráfego que uma classe pode transmitir. Esses comandos são ligeiramente diferentes em função. O comando police identifica o tráfego que excede a largura de banda configurada e o descarta ou observa. O comando shape coloca em buffer qualquer excesso de tráfego e o agenda para transmissão em uma taxa constante, mas não descarta ou marca novamente.
Limite de fila	Atribui um comprimento fixo à fila de uma determinada classe de tráfego. Você pode especificar isso no número de pacotes que podem ser mantidos na fila.
prioridade	Identifica uma fila como uma fila de baixa latência. O MQC suporta modo estrito somente para um PQ. O modo alternativo não é suportado através do MQC. Use o comando priority sem um valor percentual para ativar o modo de prioridade estrita. Observação: a implementação do comando priority no Cisco 12000 Series difere da implementação em outros roteadores que executam o software Cisco IOS. Nesta plataforma, o tráfego prioritário não está limitado ao valor de kbps configurado durante períodos de congestionamento. Assim, você

	também deve configurar o comando police para limitar a largura de banda que uma classe de prioridade pode usar e garantir a largura de banda adequada para outras classes. No momento, o comando police só é suportado em placas de linha do Engine 3. Nas outras placas de linha do mecanismo, somente class-default é permitido quando você configura uma classe de prioridade.
random-detect	Atribui um perfil WRED. Use o comando random-detect precedence para configurar valores WRED não padrão por valor de precedência IP.

Nas LCs do Engine 3, você deve configurar as filas FrFab com a CLI de QoS modular (MQC); a CLI (Command Line Interface, interface de linha de comando) legada não é suportada.

Ao configurar o comando **bandwidth**, observe que os LCs do Engine 0 e 2 suportam apenas seis classes de largura de banda. Uma sétima classe pode ser usada para serviço de baixa latência e uma oitava classe, que é padrão de classe, é usada para todo o tráfego não correspondente. Portanto, você tem um total de oito filas. O padrão da classe não é utilizado como classe de prioridade.

Nas LCs do Engine 3, o comando **bandwidth percent** é convertido em um valor kbps, que varia com a taxa de link subjacente, e então configurado diretamente na fila. A precisão dessa garantia de largura de banda mínima é de 64 Kbps.

Embora nenhuma conversão para um valor quântico seja feita com o comando **bandwidth**, todas as filas têm um quantum. Nas LCs do Engine 3, o valor quântico é definido internamente com base na MTU (Maximum Transmission Unit, unidade máxima de transmissão) da interface e é definido igualmente para todas as filas. Não existe mecanismo CLI MQC para modificar esse valor quântico, direta ou indiretamente. O valor quântico deve ser maior ou igual ao MTU da interface. Internamente, o valor quântico está em unidades de 512 bytes. Assim, com um MTU de 4.470 bytes, o valor quântico mínimo de MTU deve ser 9.

[MDRR na LC do Engine 3](#)

Esta seção fornece notas de configuração para implementar WRED e MDRR em LCs do Engine 3.

- A largura de banda de MDRR configurada na CLI é convertida em uma quantidade correspondente a L2 (por exemplo, a sobrecarga de L1 é removida). Em seguida, essa quantidade é arredondada para os próximos 64 kbps e programada no hardware.
- Três perfis WRED diferentes são suportados para uma classe.
- O WRED (limiar máximo - limiar mínimo) é aproximado à potência mais próxima de 2. O limiar mínimo é então ajustado automaticamente enquanto o limiar máximo é mantido inalterado.
- Há suporte para o valor de probabilidade Mark 1.
- A configuração constante de ponderação exponencial não é suportada.
- Precedência de IP, bits MPLS EXP e valores DSCP são suportados.

Observação: cada porta ou canal no Tetra (4GE-SFP-LC=) ou CHOC12/DS1-IR-SC= As placas

de linha Frostbite têm quatro filas alocadas por padrão. As quatro filas consistem no seguinte:

- Uma classe de fila de prioridade (LLQ)
- Uma classe de fila padrão
- Duas classes não prioritárias normais

Ao aplicar uma política de serviço que contenha mais de quatro classes (1 HPQ, 2 LPQs e class-default) à interface, o seguinte erro será relatado:

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

% Não há recursos de enfileiramento suficientes disponíveis para atender à solicitação.

A partir de 12.0(26)S, um comando foi adicionado para a placa de linha 4GE-SFP-LC= Tetra que permite a configuração de oito filas/VLAN em vez de quatro. As oito filas consistem no seguinte:

- Um LLQ
- Uma fila padrão de classe
- Seis filas normais

O uso desse comando exigirá uma recarga de microcódigo da placa de linha e resultará na capacidade de configurar apenas 508 VLANs em vez de 1022. A sintaxe do comando é:

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

Por exemplo:

```
Router(config)#hw-module slot 2 qos interface queues 8
```

aviso: Faça o microcarregamento da placa de linha para que este comando entre em vigor

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

Este comando estará disponível para a placa de linha CHOC12/DS1-IR-SC= Frostbite em 12.0(32)S

Exemplo 1 - bandwidth percent Command

Este exemplo aloca 20% de largura de banda para class Prec_4 traffic e 30% para traffic of class Prec_3 traffic. Deixa os 50% restantes para a classe padrão de classe.

Além disso, ele configura a WRED como o mecanismo de queda em todas as classes de dados.

Exemplo 1 - porcentagem de largura de banda

```
policy-map GSR_EXAMPLE
class Prec_4
  bandwidth percent 20
  random-detect
  random-detect precedence 4 1498 packets 9690 packets 1
!--- All data classes should have WRED configured. class
Prec_3 bandwidth percent 30 random-detect random-detect
precedence 3 1498 packets 9690 packets 1 class class-
default !--- Class-default uses any leftover bandwidth.
random-detect random-detect precedence 2 1498 packets
```

```
9690 packets 1 random-detect precedence 1 1498 packets
9690 packets 1 random-detect precedence 0 1498 packets
9690 packets 1
```

Exemplo 2 - bandwidth {kbps} Command

Este exemplo ilustra como aplicar o comando bandwidth como um valor absoluto de kbps em vez de uma porcentagem.

Exemplo 2 - largura 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
```

Exemplo 3 - comando priority

Este exemplo foi projetado para provedores de serviços que usam o roteador Cisco 12000 Series como um roteador de borda de provedor (PE) MPLS e precisam configurar uma política de serviço de QoS no link entre o roteador PE e o roteador de borda de cliente (CE). Ele coloca pacotes IP precedence 5 em uma fila prioritária e limita a saída dessa fila para 64 Mbps. Em seguida, atribui uma parte da largura de banda restante às classes de largura de banda.

Todas as filas de classe não prioritárias são configuradas com o comando **random-detect** para ativar o WRED como a política de queda. Toda classe de largura de banda e classe padrão devem ter WRED configurado explicitamente.

Exemplo 3 - prioridade

```
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
```

[Etapa 3 - Atribuir um mapa de política a uma fila de interface de saída](#)

Como mencionado acima, o MQC funciona somente com as filas FrFab em uma interface de saída. Para aplicar um mapa de política definido, use o comando **service-policy output**, como mostrado aqui:

```
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
```

[Etapa 4 - Monitorar e verificar a política de serviço](#)

Use o comando **show policy-map interface** para exibir a aplicação de uma política. O comando **show policy-map interface** exibe o seguinte:

- Classes de largura de banda e prioridade configuradas e os critérios de correspondência.
- Qualquer perfil WRED.
- Forma e parâmetros policiais.
- Tarifação e taxas de tráfego.
- A fila de CoS interna para a qual uma determinada classe é mapeada. Essas filas são referenciadas pelo mesmo índice usado na saída do comando **show controller frfab queue**.

Aqui está um exemplo de uma configuração completa e os comandos **show** para monitorar a política:

Configuração completa

```
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.
```

Use o comando **show policy-map interface** para exibir a política configurada na interface, juntamente com todas as classes configuradas. Aqui está a saída do 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 monitorar o gerenciamento e a prevenção de congestionamentos

Esta seção lista os comandos que você pode usar para monitorar o gerenciamento de congestionamento e a política de prevenção.

[A Tabela 8](#) lista os comandos relevantes para as placas de linha de entrada e saída.

Tabela 8 - Comandos das placas de linha

Placa de linha de entrada	Placa de Saída
<ul style="list-style-type: none">• show interfaces• exec slot <x> sh controller tofab queue• exec slot <x> show controller tofab queue <slot> <port>• exec slot <x> show controller tofab qm stat	<ul style="list-style-type: none">• show interfaces• show interfaces <y> random• exec slot <y> show controller frfab queue• exec slot <y> show controller frfab queue <port>• exec slot <y> show controller frfab QM stat

Esses comandos são explicados nesta seção.

O comando show interfaces

Antes de usar esse comando, confirme a "Estratégia de enfileiramento" correta. Se a saída exibir First In, First Out (FIFO), verifique se o comando **service-policy** aparece na configuração em execução (se o MQC tiver sido usado para configurar o MDRR).

Monitore o número de quedas de saída, que representa o número total de quedas FrFab da WRED ocorridas para o tráfego de saída nessa interface. O número de quedas de saída na saída do comando **show interfaces** deve ser igual ou superior ao número de quedas de saída na saída do comando **show interfaces <number> random**.

Observação: no Cisco 12000 Series Router, as quedas de saída da interface são atualizadas depois que as quedas de WRED são atualizadas. Há uma pequena chance de que se você usar uma ferramenta para consultar ambos os contadores de queda, as quedas de interface ainda não sejam atualizadas.

```
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
```

O Comando show interfaces {number} random

Ao usar esse comando, você deve:

- Verifique se o modelo **cos-queue-group** correto está aplicado a esta interface.
- Verifique os pesos de MDRR. Para cada fila MDRR, você pode verificar a média ponderada do comprimento da fila e o valor mais alto alcançado (em pacotes). Os valores são calculados como uma média ponderada e não precisam refletir a profundidade máxima real da fila jamais alcançada.
- Verifique os limiares mínimo e máximo de WRED.

- Verifique o número de perdas aleatórias e perdas de limiar para cada rótulo RED (perdas "To Fabric" indicam a quantidade total de perdas para esse rótulo em todas as placas de ingresso).
- O contador "TX-queue-limit drops" é usado somente em LCs Engine 1, que não suportam WRED. As placas Engine 1 permitem definir o limite das filas MDRR com o comando **TX-queue-limit interface**. Quando há suporte para WRED, os limiares de WRED determinam a profundidade das filas 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
0	29065142	73492	9614385	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0

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

```
Queue Lengths
```

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

Queue	Average	High Water Mark	Weight
0	0.000	2278.843	1
1	0.000	0.000	73
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
```

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

Esse comando exibe a profundidade instantânea da fila para uma determinada porta em um determinado slot. O exemplo de saída nesta seção exibe a fila MDRR na interface POS 4/1. Você vê uma profundidade de fila para a fila MDRR 1 de pacotes de 1964. O peso é o número de bytes que podem ser servidos nesta fila. Esse peso determina a porcentagem de largura de banda que você deseja dar a esta fila. O déficit é o valor que informa ao algoritmo DRR quantos pacotes ainda precisam ser servidos. Você pode ver que não há pacotes enfileirados no LLQ (fila 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

```

Esse comando é utilizado principalmente para monitorar a profundidade da Fila de prioridade da placa de linha de saída. Quando você vê que os pacotes começam a esperar nesta LLQ, é uma boa indicação de que você deve encaminhar algum tráfego de Voz sobre IP (VOIP) para outras placas de linha de saída. Em um bom design, o comprimento deve ser sempre 0 ou 1. Em uma rede real, você experimentará tráfego em surtos, mesmo para dados de voz. O retardo extra se torna mais grave quando a carga de voz total excede 100% da largura de banda de entrada para um intervalo curto. O roteador não pode colocar mais tráfego na rede do que o permitido, por isso, o tráfego de voz é enfileirado em sua própria fila de prioridade. Isso cria a latência de voz e o jitter de voz introduzidos pela intermitência do próprio tráfego 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

```

A fila 7 é o LLQ, e o comprimento informa quantos pacotes estão nesse LLQ.

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

Use este comando quando suspeitar que a memória de pacote de um LC comece a se aproximar da capacidade total. Um valor crescente para o contador "no mem drop" sugere que o WRED não está configurado ou que os limiares de WRED estão definidos muito altos. Este contador não deve aumentar em condições normais. Consulte [Troubleshooting de Ignored Packets and No Memory Drops no Cisco 12000 Series Internet Router](#) para obter mais informações.

```

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
```

Monitorar gerenciamento de congestionamento de entrada

Esta seção descreve os comandos usados para monitorar o gerenciamento de congestionamento de entrada.

O comando show interfaces

Antes de emitir esse comando, verifique se o valor no contador ignorado está aumentando. Você verá pacotes ignorados se ficar sem memória no lado ToFab ou se a placa de linha não aceitar os pacotes com a rapidez necessária. Para obter mais informações, consulte Troubleshooting de Quedas de Entrada no Cisco 12000 Series Router.

```
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
```

O comando exec slot (x) show controller tofab queue

Este exemplo de saída do comando `exec slot <x> show controller tofab queue` foi capturado quando não havia congestionamento em uma placa de linha de saída no slot 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
```

A seguinte saída foi capturada quando havia congestionamento no slot 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
```

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

Use este comando para determinar quanta memória é usada no lado ToFab. Em particular, observe o número na coluna '#Qelem'. Observe que:

- Quando nenhuma memória é usada, os valores estão em seu nível mais alto.
- O valor da coluna "#Qelem" diminui à medida que os pacotes são armazenados em buffer.
- Quando a coluna "#Qelem" atingir zero, todos os buffers gravados estarão em uso. No Engine 2 LC, pequenos pacotes podem emprestar espaço de buffer de pacotes maiores.

Você também pode usar esse comando para determinar o número de pacotes enfileirados em uma fila de saída virtual. O exemplo aqui mostra como verificar o slot 14 quanto ao número instantâneo de pacotes nessas filas para o slot 4, porta 1 (POS 4/1). Vemos 830 pacotes na fila MDRR 1.

```
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
```

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

Use esse comando para ver o número de quedas ToFab por placa de ingresso. Verifique também se há um contador "sem perda de memória" incrementado. Esse contador é incrementado quando o CoS não está configurado no 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

```

Casos Práticos

Este estudo de caso mostra como configurar uma política típica para o núcleo da rede de um ambiente de provedor de serviços. Ele aplica comandos de fila e permite que você use MDRR/WRED para o gerenciamento de fila ativo. As políticas de QoS nos roteadores de borda normalmente usam marcação de tráfego, condicionamento e assim por diante, para permitir que os roteadores no núcleo ordenem o tráfego em classes com base nos valores de precedência de IP ou Ponto de Código de DiffServ (DSCP). Este estudo de caso usa os recursos de QoS do software Cisco IOS para atender a contratos de nível de serviço (SLAs) rígidos e diferentes níveis de serviço para serviços de voz, vídeo e dados no mesmo backbone IP.

Na abordagem, um provedor de serviços implementou três classes de tráfego. A mais importante é a LLQ ou a classe de enfileiramento de latência baixa. Essa é a classe para Voz e Vídeo. Essa classe deve experimentar um retardo mínimo e jitter e nunca deve experimentar perda de pacotes ou pacotes reordenados, desde que a largura de banda dessa classe não exceda a largura de banda do link. Essa classe é conhecida como tráfego EF PHB (Expedited Forwarding Per-Hop Behavior) na arquitetura DiffServ. O provedor de serviços de Internet (ISP) projetou a rede de uma maneira que essa classe não exceda 30% na carga média da largura de banda do link. As outras duas classes são a classe de negócios e a classe de melhor esforço.

No projeto, configuramos os roteadores de forma que a classe comercial sempre obtenha 90% da largura de banda restante e a classe de melhor esforço obtenha 10%. Essas duas classes têm menos tráfego sensível ao tempo e podem sofrer perda de tráfego, maior atraso e instabilidade. No projeto, o foco está nas placas de linha Engine 2: Placas de linha 1xOC48 rev B, 4xOC12 rev B e 8xOC3.

As placas de linha Rev B são mais adequadas para transportar tráfego VoIP devido a uma arquitetura revisada de ASIC e hardware, que introduz muito pouca latência. Com o ASIC revisado, a fila FIFO de transmissão é redimensionada pelo driver da placa de linha para aproximadamente duas vezes mais MTU na placa. Procure um "-B" adicionado ao número da peça, como OC48E/POS-SR-SC-B=.

Observação: não confunda a fila FIFO de transmissão com as filas FrFab que podem ser ajustadas nas placas de linha Engine 0 com o comando tx-queue-limit interface.

[A Tabela 9](#) lista os critérios de correspondência para cada classe.

Tabela 9 - Critérios de correspondência para cada classe

Nome da classe	Critérios de correspondência
----------------	------------------------------

Fila de Prioridade - Tráfego de voz	Precedência 5
Fila de negócios	Precedência 4
Fila de melhor esforço	Precedência 0

As placas de linha OC48 podem enfileirar um grande número de pacotes nas filas ToFab. Assim, é importante configurar o MDRR/WRED nas filas ToFab, especialmente quando a interface de saída é uma interface de alta velocidade, como a OC48. A tela pode comutar apenas o tráfego para a placa de linha de recepção, em uma taxa máxima teórica de 3 Gbps (pacotes de 1500 bytes). Se a quantidade total de tráfego enviado for maior do que o que a matriz de comutação pode transportar para sua placa receptora, muitos pacotes serão enfileirados nas filas 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
```

Espera-se que quanto mais tráfego VOIP você tiver, mais tráfego comercial terá que esperar antes de ser atendido. No entanto, isso não é um problema porque o SLA restrito não requer descarte de pacotes e latência e jitter muito baixos para a fila prioritária.

[Informações Relacionadas](#)

- [Como ler a saída do comando show controller frfab | Comandos tofab queue em um Cisco 12000 Series Internet Router](#)
- [Troubleshooting de Pacotes Ignorados e Quedas Sem Memória no Cisco 12000 Series Internet Router](#)
- [Troubleshooting de Quedas de Entrada no Cisco 12000 Series Internet Router](#)
- [Detecção antecipada aleatória ponderada no Cisco 12000 Series Router](#)
- [Visão geral da interface de linha de comando de qualidade de serviço modular](#)
- [Página de suporte dos roteadores de Internet da série 12000](#)
- [Suporte Técnico - Cisco Systems](#)