

Troubleshooting de Quedas de Saída com Filas de Prioridade

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Quedas com prioridade ip rtp e LLQ](#)

[Quedas com enfileiramento de prioridade legada](#)

[Medição de tráfego com um token bucket](#)

[Passos de Troubleshooting para Diagnosticar Quedas](#)

[Etapa 1 – Coletar dados](#)

[Passo 2 - Garanta largura de banda suficiente](#)

[Etapa 3 – Garanta um tamanho de intermitência suficiente](#)

[Etapa 4 – debug priority](#)

[Outras causas de quedas](#)

[Queda de filas de prioridade e frame relay](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento fornece dicas de como resolver problemas relacionados a quedas de saída que resultam de uma configuração do mecanismo de prioridade de enfileiramento em uma interface do roteador.

[Pré-requisitos](#)

[Requisitos](#)

Os leitores deste documento devem ser familiares com estes conceitos:

- o **prioridade-grupo** ou o **prioridade-grupo do Frame Relay** permitem o mecanismo de enfileiramento de prioridade do legado Cisco. Apoios até quatro níveis das filas de prioridade.
- **IP RTP Priority** ou **Frame Relay IP RTP Priority** - Os fósforos em números de porta UDP para o Real-Time Protocol (RTP) traficam encapsulando pacotes voip e colocam estes pacotes em uma fila de prioridade.
- **prioridade** - Permite a característica do low latency queueing de Cisco (LLQ) e usa a estrutura de comando do QoS Command-Line Interface da Qualidade de Serviço modular (CLI).

Um roteador pode relatar quedas de emissor quando alguns métodos estão configurados, mas há algumas diferenças funcionais importantes entre os métodos e a razão para gotas em cada caso.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se você estiver trabalhando em uma rede ativa, assegure-se de compreender o impacto potencial de qualquer comando antes de utilizá-lo.

[Componentes Utilizados](#)

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se você estiver trabalhando em uma rede ativa, certifique-se de que entende o impacto potencial de qualquer comando antes de utilizá-lo.

[Convenções](#)

Para obter mais informações sobre as convenções de documento, refira-se a [convenções usadas nos dicas técnicas da Cisco](#).

[Quedas com prioridade ip rtp e LLQ](#)

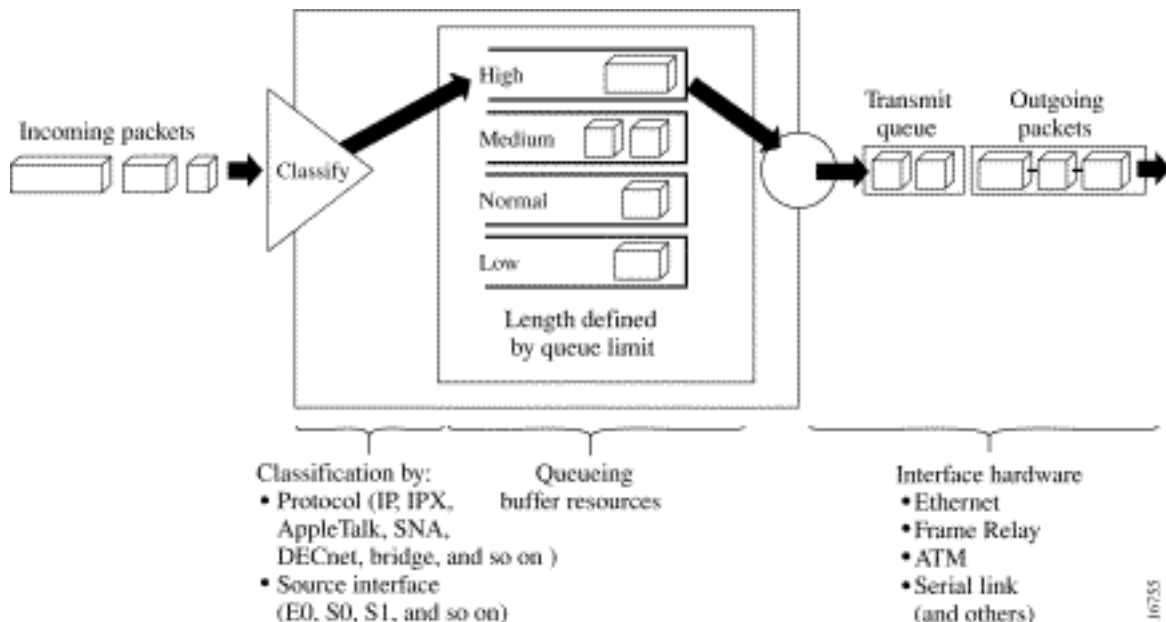
O guia de configuração do IOS da Cisco adverte contra quedas de emissor com estes mecanismos de enfileiramento de prioridade:

- **ip rtp priority**: Porque o **comando ip rtp priority** dá a prioridade absoluta sobre o outro tráfego, deve ser usado com cuidado. No caso de um congestionamento, se o tráfego exceder a largura de banda configurada, todo o tráfego em excesso será desconectado.
- **comando priority** e LLQ: Quando você especifica o **comando priority** para uma classe, toma um argumento de largura de banda que dê a largura de banda máxima. No caso da congestão, policier está usado para deixar cair pacotes quando a largura de banda é excedida.

Esses dois mecanismos usam um vigilante interno para medir os fluxos de tráfego. A finalidade do vigilante é assegurar que as outras filas sejam atendidas pelo agendador de enfileiramento. No recurso original de enfileiramento de prioridades da Cisco, que utiliza os comandos `priority-group` e `priority-list`, o programador sempre serviu primeiramente a fila com prioridade mais alta. Se havia sempre um tráfego na fila de alta prioridade, as filas de baixa prioridade foram morridas de fome da largura de banda e dos pacotes que vão às filas sem prioridade.

[Quedas com enfileiramento de prioridade legada](#)

Enfileiramento de Prioridades (PO) é o mecanismo de enfileiramento de prioridades herdadas da Cisco. Conforme ilustrado abaixo, o PQ suporta até quatro níveis de filas: alto, médio, normal e baixo.



Habilitar a fila de prioridades em uma interface altera a exibição da fila de Saída, conforme ilustrado abaixo. Antes do enfileiramento de prioridades, a interface Ethernet está usando uma fila de espera de saída única com o tamanho de fila padrão de 40 pacotes.

```
R6-2500# show interface ethernet0 Ethernet0 is up, line protocol is up Hardware is Lance,
address is 0000.0c4e.59b1 (bia 0000.0c4e.59b1) Internet address is 42.42.42.2/24 MTU 1500 bytes,
BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255 Encapsulation ARPA, loopback not set,
keepalive set (10 sec) ARP type: ARPA, ARP Timeout 04:00:00 Last input 00:00:03, output
00:00:02, output hang never Last clearing of "show interface" counters never Queueing strategy:
fifo Output queue 0/40, 0 drops; input queue 0/75, 0 drops 5 minute input rate 0 bits/sec, 0
packets/sec 5 minute output rate 0 bits/sec, 0 packets/sec 239407 packets input, 22644297 bytes,
0 no buffer Received 239252 broadcasts, 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0
frame, 0 overrun, 0 ignored, 0 abort 0 input packets with dribble condition detected 374436
packets output, 31095372 bytes, 0 underruns 0 output errors, 1 collisions, 13 interface resets 0
babblers, 0 late collision, 8 deferred 0 lost carrier, 0 no carrier 0 output buffer failures, 0
output buffers swapped out
```

Após ter permitido o PQ a interface Ethernet agora está usando quatro filas de prioridade com limites de fila de variação, segundo as indicações da saída abaixo:

```
R6-2500(config)# interface ethernet0 R6-2500(config-if)# priority-group 1 R6-2500(config-if)#
end R6-2500# show interface ethernet 0 Ethernet0 is up, line protocol is up Hardware is Lance,
address is 0000.0c4e.59b1 (bia 0000.0c4e.59b1) Internet address is 42.42.42.2/24 MTU 1500 bytes,
BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255 Encapsulation ARPA, loopback not set,
keepalive set (10 sec) ARP type: ARPA, ARP Timeout 04:00:00 Last input 00:00:03, output
00:00:03, output hang never Last clearing of "show interface" counters never Input queue: 0/75/0
(size/max/drops); Total output drops: 0 Queueing strategy: priority-list 1 Output queue (queue
priority: size/max/drops): high: 0/20/0, medium: 0/40/0, normal: 0/60/0, low: 0/80/0 5 minute
input rate 0 bits/sec, 0 packets/sec 5 minute output rate 0 bits/sec, 0 packets/sec 239411
packets input, 22644817 bytes, 0 no buffer Received 239256 broadcasts, 0 runts, 0 giants, 0
throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort 0 input packets with
dribble condition detected 374440 packets output, 31095658 bytes, 0 underruns 0 output errors, 1
collisions, 14 interface resets 0 babblers, 0 late collision, 8 deferred 0 lost carrier, 0 no
carrier 0 output buffer failures, 0 output buffers swapped out
```

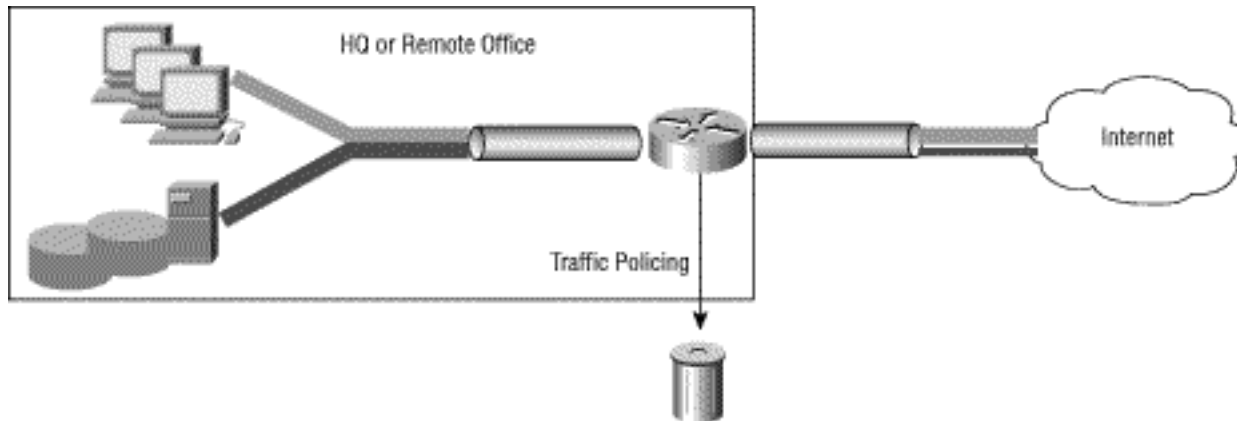
[O comando priority-list {list-number}](#) é usado atribuir fluxos de tráfego a uma fila específica.

Conforme os pacotes chegam a uma interface, as filas de prioridade nessa interface são varridas por pacotes em uma ordem decrescente de prioridade. A fila de alta prioridade é feita a varredura primeiramente, então a fila de prioridade média, e assim por diante. O pacote na cabeça da fila a mais prioritária é escolhido para a transmissão. Esse procedimento é repetido toda vez que um pacote precisa ser enviado.

Cada fila é definida por um tamanho máximo ou pelo número máximo de pacotes que a fila pode reter. Quando um pacote chegando faria com que a profundidade de fila atual excedesse o limite de fila configurado, o pacote é deixado cair. Por isso, como foi observado acima, as quedas de saída com PQ normalmente acontecem devido ao estouro do limite de fila e não a um vigilante interno, como é o caso típico com o LLQ. [O comando priority-list list-number queue-limit altera o tamanho de uma fila de prioridade.](#)

Medição de tráfego com um token bucket

As prioridades LLQ e IP RTP implementam o vigilante interno usando um token bucket como um sistema de medida de tráfego. Esta seção aborda o conceito de token bucket.



Um token bucket propriamente dito não possui política de descarte ou prioridade. A metáfora do token bucket funciona nas seguintes linhas:

- Os tokens são colocados em um bucket a uma certa taxa.
- Cada token significa a permissão para que a fonte envie um determinado número de bit na rede.
- Para enviar um pacote, o regulador de tráfego deve ser capaz de remover vários tokens do bucket, iguais em representação ao tamanho do pacote.
- Se não houver tokens suficientes no bucket para enviar um pacote, o pacote aguardará até que o bucket tenha tokens suficientes (no caso de um formador) ou o pacote seja descarregado ou marcado (no caso de um vigilante).
- O bucket propriamente dito possui uma capacidade específica. Se o bucket for totalmente preenchido, os tokens recém-chegados serão descartados e não estarão disponíveis para pacotes futuros. Assim, a qualquer momento, o maior surto que um aplicativo pode enviar na rede é proporcional ao tamanho do pacote. Um token bucket permite intermitência, mas a limita.

Vamos considerar um exemplo usando pacotes e uma CIR (taxa de informações consolidadas) de 8.000 bps.

- Neste exemplo, os token buckets iniciais começam com 1000 bytes.
- Quando um pacote de 450 bytes chega, ele entra em conformidade porque bytes suficientes estão disponíveis no token bucket de conformidade. O pacote é enviado e 450 bytes são removidos do token bucket, deixando 550 bytes.
- Quando o próximo pacote chegar 0,25 segundos mais tarde, 250 bytes serão adicionados ao token bucket $((0,25 * 8000)/8)$, deixando 700 bytes no token bucket. Se o próximo pacote é 800 bytes, o pacote excede, e está deixado cair. Nenhum byte foi retirado do token bucket.

Passos de Troubleshooting para Diagnosticar Quedas

Etapa 1 – Coletar dados

As etapas para recolher dados são mostradas abaixo.

1. Execute os comandos a seguir várias vezes e determine a rapidez e a frequência do aumento de quedas. Use a saída para estabelecer uma linha de base de seus testes padrão de tráfego e níveis de tráfego. Calcule qual é a taxa de queda normal na interface.
show queueing interface
router# **show queueing interface hssi 0/0/0** Interface Hssi0/0/0 queueing strategy: priority Output queue utilization (queue/count) high/12301 medium/4 normal/98 low/27415
show interface - Monitore o valor da carga exibido na saída. Além disso, garanta que a soma das contas de queda por fila na saída da interface **show** seja equivalente à conta de quedas de saída. As gotas das **saídas de interface da mostra** contrárias devem indicar o agregado total de todas as gotas na saída, incluindo o descarte WRED, o descarte devido à falha de buffer (erros do “sem buffer”), e mesmo os descartes na memória de adaptador de porta a bordo.
router# **show interface serial 4/1/2** Serial4/1/2 is up, line protocol is up Hardware is cyBus Serial Description: E1 Link to 60W S9/1/2 Backup Internet address is 169.127.18.228/27 MTU 1500 bytes, BW 128 Kbit, DLY 21250 usec, rely 255/255, **load 183/255** Encapsulation HDLC, loopback not set, keepalive set (10 sec) Last input 00:00:00, output 00:00:00, output hang never Last clearing of "show interface" counters 5d10h Input queue: 0/75/0 (size/max/drops); Total output drops: **68277** **Queueing strategy: priority-list 7 Output queue: high 0/450/0, medium 0/350/143, normal 0/110/27266, low 0/100/40868** 5 minute input rate 959000 bits/sec, 419 packets/sec 5 minute output rate 411000 bits/sec, 150 packets/sec 144067307 packets input, 4261520425 bytes, 0 no buffer Received 0 broadcasts, 0 runts, 0 giants, 0 throttles 42 input errors, 34 CRC, 0 frame, 0 overrun, 1 ignored, 8 abort 69726448 packets output, 2042537282 bytes, 0 underruns 0 output errors, 0 collisions, 0 interface resets 0 output buffer failures, 46686454 output buffers swapped out 0 carrier transitions **Nota: Algumas interfaces exibem valores "txload" e "rxload" separados.**
Hssi0/0/0 is up, line protocol is up
Hardware is cyBus HSSI
MTU 1500 bytes, BW 7500 Kbit, DLY 200 usec,
reliability 255/255, **txload 138/255, rxload 17/255** Encapsulation FRAME-RELAY IETF, crc 16, loopback not set Keepalive set (5 sec) LMI enq sent 4704, LMI stat recvd 4704, LMI upd recvd 0, DTE LMI up LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0 LMI DLCI 1023 LMI type is CISCO frame relay DTE Broadcast queue 0/256, broadcasts sent/dropped 8827/0, interface broadcasts 7651 Last input 00:00:00, output 00:00:00, output hang never Last clearing of "show interface" counters 06:31:58 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 84 **Queueing strategy: priority-list 1 Output queue (queue priority: size/max/drops): high: 0/20/0, medium: 0/40/0, normal: 0/60/0, low: 0/80/84** 5 minute input rate 524000 bits/sec, 589 packets/sec 5 minute output rate 4080000 bits/sec, 778 packets/sec 11108487 packets input, 1216363830 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 15862186 packets output, 3233772283 bytes, 0 underruns 0 output errors, 0 applique, 1 interface resets 0 output buffer failures, 2590 output buffers swapped out 0 carrier transitions LC=down CA=up TM=down LB=down TA=up LA=down
show policy-map interface interface-name – Procure um valor diferente de zero para o contador de "pkts discards".
Router# **show policy-map interface s1/0** Serial1/0.1: DLCI 100 - output : mypolicy Class voice Weighted Fair Queueing **Strict Priority** Output Queue: Conversation 72 Bandwidth 16 (kbps) Packets Matched 0 (**pkts discards/bytes discards**) 0/0 Class immediate-data Weighted Fair Queueing Output Queue: Conversation 73 Bandwidth 60 (%) Packets Matched 0 (**pkts discards/bytes discards/tail drops**) 0/0/0 mean queue depth: 0 drops: class random tail min-th max-th mark-prob 0 0 0 64 128 1/10 1 0 0 71 128 1/10 2 0 0 78 128 1/10 3 0 0 85 128 1/10 4 0 0 92 128 1/10 5 0 0 99 128 1/10 6 0 0 106 128 1/10 7 0 0 113 128 1/10 rsvp 0 0 120 128 1/10 **Nota: A saída de exemplo seguinte indica valores correspondentes para os “pacotes” e contadores**

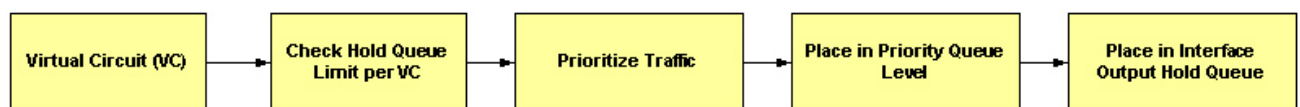
de "pacotes compatíveis". Esta condição indica que um grande número de pacotes está sendo comutado ou que a interface está passando por um congestionamento enorme.

Essas duas condições podem gerar um limite excedente de fila de classes e devem ser

investigadas.
router# show policy-map interface Serial4/0 Service-policy output: policy1
Class-map: class1 (match-all) 189439 packets, 67719268 bytes 5 minute offered rate 141000 bps, drop rate 0 bps Match: access-group name ds-class-af3 Weighted Fair Queueing Output Queue: Conversation 265 Bandwidth 50 (%) Max Threshold 64 (packets) (pkts matched/bytes matched) 189439/67719268 (depth/total drops/no-buffer drops) 0/0/0

2. Caracterize os fluxos de tráfego e os pacotes nesses fluxos. Que é o tamanho médio do pacote? Para qual direção os quadros de tamanho MTU estão fluindo? Muitos fluxos de tráfego são assíncronos no que diz respeito à carga. Por exemplo, com download via FTP, a maioria dos pacotes de tamanho de MTU flui do servidor FTP para o cliente. Os pacotes do cliente FTP para o servidor são ACKs TCP simples. Os pacotes estão usando TCP ou UDP? O TCP permite que cada fluxo envie um número autorizado de pacotes antes que a fonte precise de suspender a transmissão e esperar o destino para reconhecer os pacotes transmitido.
3. Com o Frame Relay, determine se os descartes estão ocorrendo na fila de interface ou em uma fila por VC. O diagrama a seguir ilustra o fluxo dos pacotes por um circuito virtual de Frame

Relay:



4. O enfileiramento de prioridade suporta até quatro filas de saída, uma por nível de prioridade de fila, e cada fila é definida por um limite. O sistema de enfileiramento verifica o tamanho da fila em relação ao limite de fila configurado, colocando o pacote em uma fila. Se a fila selecionada estiver cheia, o roteador desconectará o pacote. Tente aumentar o tamanho da fila com o comando `priority-list {-} queue-limit` e recomece monitorar.

Passo 2 - Garanta largura de banda suficiente

Com LLQ, policiar permite o tratamento justo de outros pacotes de dados no outro Class-Based Weighted Fair Queuing (CBWFQ) ou nas filas WFQ. Para evitar quedas de pacote, aloque um total ideal de largura de banda para a fila de prioridade, levando em consideração o tipo de codec usado e as características da interface. O IP RTP Priority não permitirá o tráfego além da quantidade atribuída.

É sempre mais seguro alocar um pouco mais que a quantidade exigida conhecida de largura de banda para a fila de prioridade. Por exemplo, supõe que você atribuiu 24 larguras de banda dos kbps, a quantidade padrão exigida para a transmissão de voz, à fila de prioridade. Esta alocação parece segura porque a transmissão de pacotes de voz ocorre em uma taxa de bit constante. Contudo, porque a rede e o roteador ou o interruptor podem usar alguma da largura de banda para produzir o tremor e o atraso, atribuir levemente mais do que a quantidade de largura de banda requerida (tal como 25 kbps) assegura a constância e a Disponibilidade.

A largura de banda atribuída para uma fila de prioridade inclui sempre o cabeçalho de encapsulamento da camada 2. Não inclui a verificação de redundância cíclica (CRC). (Refira [que bytes são contados pelo enfileiramento do IP to ATM CoS?](#) para mais informação.) Embora seja somente alguns bytes, as imposições de CRC um impacto crescente como fluxos de tráfego incluem um número mais alto de pacotes pequenos.

Além, em interfaces ATM, a largura de banda atribuída para uma fila de prioridade não inclui as seguintes despesas gerais da taxa de célula ATM:

- Qualquer preenchimento pela segmentação e remontagem (SAR) para tornar a última célula de um pacote um múltiplo par de 48 bytes.
- 4-byte CRC do reboque da camada de adaptação ATM 5 (AAL5).
- Cabeçalho de célula ATM de 5 bytes.

Quando você calcular o total de largura de banda a ser alocada para uma classe de prioridades específica, deverá considerar o fato de que os cabeçalhos da Camada 2 são incluídos. Quando ATM é usado, você deve levar em consideração o fato de a sobrecarga de taxa de célula ATM não estar incluída. Você deve igualmente permitir a largura de banda para a possibilidade de tremor introduzida por dispositivos de rede no trajeto da voz. Refira a [vista geral dos recursos de enfileiramento de latência baixa](#).

Ao usar filas de prioridade para levar pacotes voip, refira a [Voz sobre o IP - pelo consumo de largura de banda por chamada](#).

Etapa 3 – Garanta um tamanho de intermitência suficiente

O tratamento de uma série de pacotes que partem de uma interface através de uma fila de prioridade varia conforme o tamanho do pacote e o número de bytes que restam no Token Bucket. É importante considerar as características do fluxo de tráfego que está sendo dirigido à fila de prioridade porque o LLQ usa um vigilante, não um shaper. Um vigilante utiliza um bucket token da seguinte forma:

- O bucket é preenchido com tokens baseados na taxa de classe até o máximo do o parâmetro de burst.
- Se o número de tokens é superior ou igual a tamanho do pacote, o pacote está enviado, e o Token Bucket é decrescido. Caso contrário, o pacote será perdido.

O valor de intermitência padrão do medidor de tráfego de bucket de token para LLQs é calculado como 200 milissegundos de tráfego na taxa de largura de banda configurada. Em alguns casos, o valor padrão é inadequado, particularmente quando o tráfego TCP está indo para a fila de prioridade. Os fluxos TCP normalmente são do tipo burst e podem exigir um tamanho de burst maior do que o padrão atribuído pelo sistema de filas, em especial em enlaces lentos.

O seguinte exemplo de saída foi gerado em um ATM PVC com uma taxa de célula sustentada dos kbps 128. O sistema de enfileiramento ajusta o tamanho de intermitência à medida que o valor especificado com o comando de prioridade é alterado.

```
7200-17# show policy-map int atm 4/0.500 ATM4/0.500: VC 1/500 - Service-policy output: drops
Class-map: police (match-all) 0 packets, 0 bytes 5 minute offered rate 0 bps, drop rate 0 bps
Match: any Weighted Fair Queueing Strict Priority Output Queue: Conversation 24 Bandwidth 90 (%)
Bandwidth 115 (kbps) Burst 2875 (Bytes) !--- Burst value of 2875 bytes is assigned when !--- the
reserved bandwidth value is 115 kbps. (pkts matched/bytes matched) 0/0 (total drops/bytes drops)
0/0 Class-map: class-default (match-any) 0 packets, 0 bytes 5 minute offered rate 0 bps, drop
rate 0 bps Match: any 7200-17# show policy-map int atm 4/0.500 ATM4/0.500: VC 1/500 - Service-
policy output: drops Class-map: police (match-all) 0 packets, 0 bytes 5 minute offered rate 0
bps, drop rate 0 bps Match: any Weighted Fair Queueing Strict Priority Output Queue:
Conversation 24 Bandwidth 50 (%) Bandwidth 64 (kbps) Burst 1600 (Bytes) !--- Burst value changes
to 1600 bytes when the !--- reserved bandwidth value is changed to 64 kbps. (pkts matched/bytes
matched) 0/0 (total drops/bytes drops) 0/0 Class-map: class-default (match-any) 0 packets, 0
bytes 5 minute offered rate 0 bps, drop rate 0 bps Match: any
```

A funcionalidade do LLQ foi estendida para permitir um tamanho de Bc (Committed Burst)

configurável com o recurso Configuring Burst Size in Low Latency Queueing (Configurando o Tamanho do Burst no Enfileiramento de Baixa Latência). Com essa nova funcionalidade, a rede pode agora acomodar intermitências temporárias de tráfego e lidar com o tráfego de rede de maneira mais eficiente.

Utilize o parâmetro de burst com o comando `priority` para aumentar o valor de burst de 1600 bytes para 3200 bytes.

```
policy-map AV
  class AV
    priority percent 50 3200
```

Nota: Um alto valor aumenta a largura de banda efetiva que a classe de prioridade pode usar e pode dar a aparência que as classes de prioridade estão obtendo a mais do que sua parte justa da largura de banda.

Além disso, o sistema de enfileiramento atribuiu originalmente um limite de fila interno de 64 pacotes para a fila de baixa latência. Em alguns casos, quando um burst de 64 pacotes chegou à fila de prioridade, o medidor de tráfego determina que o burst corresponda à taxa configurada, mas o número de pacotes excedeu o limite de fila. Em consequência, alguns pacotes Tail-foram deixados cair. A identificação de bug Cisco [CSCdr51979](#) ([clientes registrados somente](#)) resolve este problema permitindo que o tamanho da fila de prioridade cresça tão profundo quanto permitido pelo medidor de tráfego.

A seguinte saída foi capturada em um PVC do Frame Relay configurado com um CIR de 56 kbps. No primeiro conjunto de saída de exemplo, a taxa oferecida combinada das classes c1 e c2 é de 76 kbps. A razão é que os valores calculados das taxas oferecida menos taxas da gota não representam as taxas de transmissão real e não estão incluindo os pacotes que se sentam no shaper antes que estejam transmitidos.

```
router# show policy-map int s2/0.1 Serial2/0.1: DLCI 1000 - Service-policy output: p Class-map:
c1 (match-all) 7311 packets, 657990 bytes 30 second offered rate 68000 bps, drop rate 16000 bps
Match: ip precedence 1 Weighted Fair Queueing Strict Priority Output Queue: Conversation 24
Bandwidth 90 (%) Bandwidth 50 (kbps) Burst 1250 (Bytes) (pkts matched/bytes matched) 7311/657990
(total drops/bytes drops) 2221/199890 Class-map: c2 (match-all) 7311 packets, 657990 bytes 30
second offered rate 68000 bps, drop rate 44000 bps Match: ip precedence 2 Weighted Fair Queueing
Output Queue: Conversation 25 Bandwidth 10 (%) Bandwidth 5 (kbps) Max Threshold 64 (packets)
(pkts matched/bytes matched) 7310/657900 (depth/total drops/no-buffer drops) 64/6650/0 Class-
map: class-default (match-any) 2 packets, 382 bytes 30 second offered rate 0 bps, drop rate 0
bps Match: any
```

Neste exato segundo da saída, o comando `show policy-map interface counters` foi normalizado. No PVC de 56 kbps, a classe c1 está enviando cerca de 50 kbps e a classe c2 está enviando cerca de 6 kbps.

```
router# show policy-map int s2/0.1 Serial2/0.1: DLCI 1000 - Service-policy output: p Class-map:
c1 (match-all) 15961 packets, 1436490 bytes 30 second offered rate 72000 bps, drop rate 21000
bps Match: ip precedence 1 Weighted Fair Queueing Strict Priority Output Queue: Conversation 24
Bandwidth 90 (%) Bandwidth 50 (kbps) Burst 1250 (Bytes) (pkts matched/bytes matched)
15961/1436490 (total drops/bytes drops) 4864/437760 Class-map: c2 (match-all) 15961 packets,
1436490 bytes 30 second offered rate 72000 bps, drop rate 66000 bps Match: ip precedence 2
Weighted Fair Queueing Output Queue: Conversation 25 Bandwidth 10 (%) Bandwidth 5 (kbps) Max
Threshold 64 (packets) (pkts matched/bytes matched) 15960/1436400 (depth/total drops/no-buffer
drops) 64/14591/0 Class-map: class-default (match-any) 5 packets, 1096 bytes 30 second offered
rate 0 bps, drop rate 0 bps Match: any
```

[Etapa 4 – debug priority](#)

[O comando debug priority exibe a saída da fila de prioridade quando os pacotes são provenientes](#)

[dessa fila.](#)

Cuidado: Antes de utilizar **comandos debug**, consulte [Informações Importantes sobre Comandos Debug](#). O comando **debug priority** pode imprimir uma grande quantidade de resultado do debug disruptivo em um roteador de produção. A quantidade depende do nível de congestionamento.

O seguinte exemplo de saída foi gerado em um Cisco 3640.

```
r3-3640-5# debug priority Priority output queueing debugging is on r3-3640-5# ping 10.10.10.2
Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2
seconds: !!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms r3-3640-
5# 00:42:40: PQ: Serial0/1: ip -> normal 00:42:40: PQ: Serial0/1 output (Pk size/Q 104/2)
00:42:40: PQ: Serial0/1: ip -> normal 00:42:40: PQ: Serial0/1 output (Pk size/Q 104/2) 00:42:40:
PQ: Serial0/1: ip -> normal 00:42:40: PQ: Serial0/1 output (Pk size/Q 104/2) 00:42:40: PQ:
Serial0/1: ip -> normal 00:42:40: PQ: Serial0/1 output (Pk size/Q 104/2) 00:42:40: PQ:
Serial0/1: ip -> normal 00:42:40: PQ: Serial0/1 output (Pk size/Q 104/2) 00:42:40: PQ:
Serial0/1: ip -> normal 00:42:40: PQ: Serial0/1 output (Pk size/Q 104/2) 00:42:41: PQ: Serial0/1
output (Pk size/Q 13/0) r3-3640-5#no debug priority 00:42:51: PQ: Serial0/1 output (Pk size/Q
13/0) Priority output queueing debugging is off
```

Nos seguintes resultados de debug priority, 64 indicam a profundidade real da fila de prioridade então o pacote foi deixado cair.

```
*Feb 28 16:46:05.659:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.671:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.679:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.691:WFQ:dropping a packet from the priority queue 64
```

[Outras causas de quedas](#)

Os seguintes motivos para descartes de saída com LLQ foram descobertos pelo Cisco Technical Assistance Center (TAC) durante o Troubleshooting de um problema e documentados em um relatório de bugs da Cisco:

- O aumento dos valores de limiar máximo do WRED em outra classe esvaziou os buffers disponíveis e levou a quedas na fila de prioridade. Para ajudar a diagnosticar o problema, um contador "no-buffer drops" para a classe de prioridade está planejado para lançamento futuro do IOS.
- Se o limite de fila da interface de entrada for menor que o da interface de saída, o pacote cairá em deslocamento para a interface de entrada. Estes sintomas são documentados na identificação de bug Cisco [CSCdu89226](#) ([clientes registrados somente](#)). Resolva este problema fazendo sob medida a fila de entrada e a fila de saída apropriadamente para impedir caídas de entrada e permitir que o mecanismo de filas da prioridade de saída tome o efeito.
- Permitir uma característica que não seja apoiada no comutado por CEF ou o caminho fast-switched faz com que um grande número pacotes sejam comutados por processo. Com o LLQ, os pacotes comutados por processos são vigiados atualmente, independentemente de a interface estiver congestionada. Em outras palavras, mesmo que a interface não esteja congestionada, o sistema de enfileiramento mede pacotes comutados por processos e garante que a carga oferecida não exceda o valor de largura de banda configurado com o comando de prioridade. Este problema é documentado na identificação de bug Cisco [CSCdv86818](#) ([clientes registrados somente](#)).

[Queda de filas de prioridade e frame relay](#)

O Frame Relay é um caso especial no que diz respeito a policiar a fila de prioridade. [A visão geral do recurso Enfileiramento de baixa latência para frame relay](#) observa o seguinte caveat: “O PQ é policiado para assegurar-se de que as filas consideráveis não estejam morridas de fome da largura de banda. Ao configurar o PQ, especifique em kbps a quantidade máxima de largura de banda disponível para essa fila. Os pacotes que excederem esse limite máximo serão descartados. Em outras palavras, originalmente, a fila de prioridade de uma política de serviço configurada em uma classe de mapas de Frame Relay era vigiada durante períodos com ou sem congestionamento. Os IO 12.2 removem esta exceção. O PQ é policiado ainda com FRF.12, mas outros pacotes sem adequação são deixados cair somente se há uma congestão.

[Informações Relacionadas](#)

- [página de suporte de QoS](#)
- [Módulo de recurso de vigilância de tráfego do Cisco IOS Software Release 12.2](#)
- [Suporte Técnico - Cisco Systems](#)