

# MDRR/WRED auf dem Cisco Internet Router der Serie 12000 verstehen und konfigurieren

## Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[MDRR-Übersicht](#)

[Prioritätswarteschlange der MDRR](#)

[MDRR-Beispiel](#)

[MDRR-Unterstützung nach Modultyp](#)

[WRED-Übersicht](#)

[Verwenden der älteren CoS-Syntax für die Konfiguration](#)

[Modular QoS CLI \(MQC\) für Konfiguration verwenden](#)

[Befehle zur Überwachung von Überlastungsmanagement und -vermeidung](#)

[Befehl show interfaces](#)

[Der Befehl show interfaces {number} random](#)

[Der exec-Steckplatz \(y\) zeigt Controller-Fab-Warteschlange {port} Befehl.](#)

[Der exec-Steckplatz \(y\) zeigt den Controller-Fab-QM-Stat-Befehl.](#)

[Überwachung des eingehenden Überlastungsmanagements](#)

[Befehl show interfaces](#)

[Der exec-Steckplatz \(x\) zeigt Controller-ToFab-Warteschleife-Befehl](#)

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

[Der exec-Steckplatz \(x\) zeigt den Controller-Befehl tofab QM stat Command.](#)

[Anwenderbericht](#)

[Zugehörige Informationen](#)

## Einführung

In diesem Dokument wird beschrieben, wie Sie die Funktionen zur Überlastungsverwaltung und zur Überlastungsvermeidung der Cisco IOS® Software auf dem Cisco Internet Router der Serie 12000 konfigurieren.

Nachdem Sie dieses Dokument gelesen haben, müssen Sie in der Lage sein,

- Sie wissen, warum es wichtig ist, Modified Deficit Round Robin (MDRR) und Weighted Random Early Detection (WRED) in Ihrem Kernnetzwerk zu konfigurieren.

- Machen Sie sich mit der Implementierung vertraut, die MDRR und WRED auf der Cisco Serie 12000 zugrunde liegt.
- Konfigurieren Sie MDRR und WRED mithilfe der Syntax der Legacy Class of Service (CoS) und der Modular QoS CLI (MQC).

## Voraussetzungen

### Anforderungen

Die Leser dieses Dokuments sollten folgende Themen kennen:

- Eine allgemeine Kenntnis der Architektur der Cisco Internet Router der Serie 12000.
- Insbesondere ein Bewusstsein für die Warteschlangenarchitektur und folgende Begriffe: ToFab (Towards the Fabric): Diese Eigenschaft beschreibt die Empfangs-Seitenwarteschlangen einer eingehenden Linecard. FrFab (Aus der Fabric): Diese Eigenschaft beschreibt die übertrageseitigen Warteschlangen auf einer ausgehenden Linecard.

**Hinweis:** Es wird empfohlen, auch [im Register Ausgabe des Show-Controllers](#) nach [Gewusst wie Lesen der Ausgabe zu suchen. | Tab Queue Commands auf einem Cisco Internet Router der Serie 1200](#).

### Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Alle Cisco 12000-Plattformen, einschließlich der 12008, 12012, 12016, 12404, 12406, 12410 und 12416.
- Cisco IOS Software Release 12.0(24)S1.

**Hinweis:** Obwohl die Konfigurationen in diesem Dokument mit der Cisco IOS Software Release 12.0(24)S1 getestet wurden, kann jede Cisco IOS-Softwareversion, die den Cisco Internet Router der Serie 12000 unterstützt, verwendet werden.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netz Live ist, überprüfen Sie, ob Sie die mögliche Auswirkung jedes möglichen Befehls verstehen.

### Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps von Cisco zu Konventionen).

## Hintergrundinformationen

Warteschlangenmethoden definieren den Paketplanungsmechanismus oder die Reihenfolge, in der Pakete zur Übertragung über die physische Leitung an die Schnittstelle in die Warteschlange gestellt werden. Je nach der Reihenfolge und Anzahl der Wartungsarbeiten für eine Warteschlange durch eine Scheduler-Funktion unterstützen Warteschlangenmethoden auch

garantierte Mindestbandbreite und niedrige Latenzen.

Es muss sichergestellt werden, dass ein Paketplanungsmechanismus die Switching-Architektur unterstützt, auf der er implementiert wird. Weighted Fair Queuing (WFQ) ist der bekannte Scheduling-Algorithmus für die Ressourcenzuweisung auf Cisco Router-Plattformen mit einer busbasierten Architektur. Auf dem Cisco Internet Router der Serie 12000 wird diese Funktion jedoch nicht unterstützt. Herkömmliche Prioritätswarteschlangen und benutzerdefinierte Warteschlangen für die Cisco IOS Software werden ebenfalls nicht unterstützt. Stattdessen verwendet die Cisco Serie 1200 eine spezielle Warteschlangenform namens Modified Deficit Round Robin (MDRR), die relative Bandbreitengarantien sowie eine Warteschlange mit niedriger Latenz bietet. M von MDRR steht für "modifiziert"; wird die Prioritätswarteschlange im Vergleich zur DRR hinzugefügt, wenn keine Prioritätswarteschlange vorhanden ist. Weitere Informationen zur MDRR finden Sie im Abschnitt [MDRR Overview \(MDRR-Übersicht\)](#).

Darüber hinaus unterstützt die Cisco Serie 12000 Weighted Random Early Detection (WRED) als Drop-Policy in den MDRR-Warteschlangen. Dieser Überlastungsvermeidungsmechanismus stellt eine Alternative zum Standard-Tail-Drop-Mechanismus dar. Die Überlastung kann durch kontrollierte Tropfen vermieden werden.

Überlastungsvermeidungs- und Managementmechanismen wie WRED und MDRR sind besonders wichtig für FrFab-Warteschlangen von ausgehenden Schnittstellen mit relativ niedriger Geschwindigkeit, z. B. Channelized Line Cards (LCs). Die Hochgeschwindigkeits-Switch-Fabric kann Pakete sehr viel schneller an die Kanalgruppen senden, als die Kanalgruppen sie übertragen können. Wenn Warteschlangen/Puffer auf physischer Portebene verwaltet werden, kann der Backdruck auf einem Kanal alle anderen Kanäle dieses Ports beeinträchtigen. Aus diesem Grund ist es sehr wichtig, diesen Rückdruck über WRED/MDRR zu verwalten, der die Backdruck-Auswirkungen auf die betreffenden Kanäle begrenzt. Weitere Informationen zum Verwalten der Überbelegung ausgehender Schnittstellen finden Sie unter [Fehlerbehebung bei ignorierten Paketen und Speicherverlusten auf dem Cisco Internet Router der Serie 12000](#).

## [MDRR-Übersicht](#)

Dieser Abschnitt bietet eine Übersicht über MDRR (Modified Deficit Round Robin).

Wenn MDRR als Warteschlangenstrategie konfiguriert ist, werden nicht leere Warteschlangen nacheinander in Round-Robin-Form bereitgestellt. Bei jeder Bereitstellung einer Warteschlange wird eine bestimmte Datenmenge in die Warteschlange gestellt. Der Algorithmus verarbeitet dann die nächste Warteschlange. Wenn eine Warteschlange bereitgestellt wird, verfolgt die MDRR die Anzahl der Byte von Daten, die über den konfigurierten Wert hinausgezögert wurden. Bei der nächsten Übergabe, wenn die Warteschlange erneut bereitgestellt wird, werden weniger Daten in die Warteschlange gestellt, um die zuvor überzähligen Daten auszugleichen. Daher liegt die durchschnittliche Menge der pro Warteschlange in die Warteschlange gestellten Daten nahe am konfigurierten Wert. Darüber hinaus unterhält der MDRR eine Prioritätswarteschlange, die bevorzugt verarbeitet wird. In diesem Abschnitt wird die MDRR ausführlicher erläutert.

Jede Warteschlange in MDRR wird durch zwei Variablen definiert:

- Quantenwert - Dies ist die durchschnittliche Anzahl der Bytes, die in jeder Runde bereitgestellt werden.
- Defizitzähler - Hiermit wird verfolgt, wie viele Bytes eine Warteschlange in jeder Runde übertragen hat. Sie wird mit dem Quantenwert initialisiert.

Pakete in einer Warteschlange werden bereitgestellt, solange der Defizitzähler größer als Null ist. Jedes gesendete Paket verringert den Defizitzähler um einen Wert, der seiner Länge in Byte entspricht. Eine Warteschlange kann nicht mehr bedient werden, wenn der Defizitzähler null oder negativ wird. In jeder neuen Runde wird der Defizitzähler jeder nicht leeren Warteschlange um den Quantenwert erhöht.

**Hinweis:** Im Allgemeinen darf die Quantengröße für eine Warteschlange nicht kleiner als die maximale Übertragungseinheit (Maximum Transmission Unit, MTU) der Schnittstelle sein. Dadurch wird sichergestellt, dass der Scheduler immer mindestens ein Paket aus jeder nicht leeren Warteschlange bedient.

Jeder MDRR-Warteschlange kann ein relatives Gewicht zugewiesen werden, wobei eine der Warteschlangen in der Gruppe als Prioritätswarteschlange definiert ist. Die Gewichtungen weisen bei Überlastung der Schnittstelle relative Bandbreite für jede Warteschlange zu. Der MDRR-Algorithmus leitet Daten aus jeder Warteschlange in Round-Robin-Warteschlangen ein, wenn Daten in der Warteschlange gesendet werden sollen.

Wenn alle regulären MDRR-Warteschlangen Daten enthalten, werden sie wie folgt gewartet:

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

Während jedes Zyklus kann eine Warteschlange eine Menge entsprechend ihrem konfigurierten Gewicht dewarteten. Bei Line Cards der Engine 0 und Engine 2 entspricht ein Wert von 1 der Angabe, dass die Schnittstelle ein Gewicht ihrer MTU erhält. Für jede Erhöhung über 1 erhöht sich das Gewicht der Warteschlange um 512 Byte. Wenn beispielsweise die MTU einer bestimmten Schnittstelle 4470 beträgt und das Gewicht einer Warteschlange auf 3 konfiguriert ist, können durch jede Drehung  $4470 + (3-1)*512 = 5494$  Byte in die Warteschlange gestellt werden. Wenn zwei normale DRR-Warteschlangen, Q0 und Q1, verwendet werden, wird Q0 mit einem Gewicht von 1 konfiguriert, und Q1 ist mit einem Gewicht von 9 konfiguriert. Wenn beide Warteschlangen überlastet sind, darf Q0 jedes Mal durch die Rotation 4470 Byte senden, und Q1 darf  $[4470 + (9-1)*512] = 8566$  Byte senden. Dadurch wird etwa ein Drittel der Bandbreite für den Datenverkehr in Q0 benötigt, und der Datenverkehr, der in Q1 fließt, macht etwa zwei Drittel der Bandbreite aus.

**Hinweis:** Die standardmäßige De-Queue-Formel für die Berechnung der MDRR-Bandbreitenzuweisung lautet  $D = MTU + (weight-1)*512$ . In Versionen vor der Cisco IOS Software Release 12.0(21) S/ST verwendeten die Line Cards der Engine 4 eine andere Dequeue-Formel. Um das MDRR-Gewicht für eine richtige Bandbreitenzuweisung zu konfigurieren, stellen Sie sicher, dass Sie eine Cisco IOS Software Release später als 12.0(21) S/ST ausführen.

**Hinweis:** Die Quantenformel für die Line Cards der Engine 4+ ist (für die Richtung toFab ist dies für FrFab nicht gültig)  $Quantum = BaseWeight + \{BaseWeight * (QueueWeight - 1) * 512\} / MTU$ . Das Basisgewicht wird mit folgender Formel berechnet:  $BaseWeight = \{(MTU + 512 - 1) / 512\} + 5$

**Hinweis:** Alle Berechnungen werden abgerundet. d. h. alle Dezimalstellen werden ignoriert.

**Hinweis:** Informationen darüber, ob eine bestimmte Engine-Linecard MDRR unterstützt, finden Sie unter [MDRR-Unterstützung nach Modultyp](#).

## **Prioritätswarteschlange der MDRR**

Die Cisco Serie 12000 unterstützt eine Prioritätswarteschlange (PQ) in der MDRR. Diese Warteschlange bietet die geringe Verzögerung und geringen Jitter, die für zeitkritischen

Datenverkehr wie Voice over IP (VoIP) erforderlich sind.

Wie oben erwähnt, unterstützt die Cisco Serie 12000 kein Weighted Fair Queueing (WFQ). Der PQ innerhalb der MDRR arbeitet daher anders als die LLQ-Funktion (Low Latency Queueing) der Cisco IOS-Software für andere Plattformen.

Ein wesentlicher Unterschied besteht darin, wie der MDRR-Scheduler so konfiguriert werden kann, dass er den PQ in einem von zwei Modi bedient, wie in [Tabelle 1](#) aufgeführt:

**Tabelle 1: Konfigurieren des MDRR-Schedulers für den Service des PQ in zwei Modi**

	<b>Alternativer Modus</b>	<b>Prioritätsmodus strikt</b>
<b>Vorteile</b>	Hier wird der PQ zwischen den anderen Warteschlangen gewartet. Mit anderen Worten, der MDRR-Scheduler bietet alternativ Services für den PQ und alle anderen konfigurierten Warteschlangen.	Hier wird der PQ immer gewartet, wenn er nicht leer ist. Dadurch wird die geringstmögliche Verzögerung bei der Zuordnung von Datenverkehr erreicht.
<b>Nachteile</b>	Dieser Modus kann im Vergleich zum Modus mit strikter Priorität Jitter und Verzögerungen verursachen.	Dieser Modus kann andere Warteschlangen aushöhlen, insbesondere wenn die Zuordnungsströme aggressive Absender sind.

Der alternative Modus bietet weniger Kontrolle über Jitter und Verzögerungen. Wenn der MDRR-Scheduler beginnt, Pakete mit MTU-Größe aus einer Datenwarteschlange zu warten, und dann ein Sprachpaket im PQ eintrifft, bedient der Scheduler im alternativen Modus die Warteschlange ohne Priorität vollständig, bis der Defizitzähler null erreicht. Während dieser Zeit wird der PQ nicht gewartet, und die VoIP-Pakete werden verzögert.

Im strikten Prioritätsmodus hingegen bietet der Scheduler nur das aktuelle Paket ohne Priorität und wechselt dann zum PQ. Der Scheduler beginnt erst dann eine Warteschlange ohne Priorität zu bedienen, wenn der PQ vollständig leer ist.

Es ist zu beachten, dass die Prioritätswarteschlange im alternativen Prioritätsmodus mehrmals in einem Zyklus gewartet wird und somit mehr Bandbreite benötigt als andere Warteschlangen mit demselben Nenngewicht. Wie viel mehr hängt davon ab, wie viele Warteschlangen definiert sind. Bei drei Warteschlangen wird beispielsweise die Warteschlange mit niedriger Latenz doppelt so oft wie die anderen Warteschlangen behandelt und sendet das doppelte Gewicht pro Zyklus. Wenn acht Warteschlangen definiert sind, wird die Warteschlange mit niedriger Latenz siebenmal häufiger bearbeitet, und das effektive Gewicht ist siebenmal höher. Die Bandbreite, die die Warteschlange beanspruchen kann, hängt davon ab, wie oft sie pro Round-Robin bedient wird, was wiederum davon abhängt, wie viele Warteschlangen insgesamt definiert werden. Im alternativen Prioritätsmodus wird die Prioritätswarteschlange aus diesem speziellen Grund in der Regel mit einem geringen Gewicht konfiguriert.

Nehmen Sie beispielsweise an, dass vier Warteschlangen definiert sind: 0, 1, 2 und die

Prioritätswarteschlange. Im alternativen Prioritätsmodus werden alle Warteschlangen, wenn sie überlastet sind, wie folgt gewartet: **0, llq, 1, llq, 2, llq, 0, llq, 1, ....** wobei "llq" für "Low Latency Queue" steht.

Jedes Mal, wenn eine Warteschlange gewartet wird, kann sie bis zu ihrem konfigurierten Gewicht senden. Aus diesem Grund kann die Warteschlange mit niedriger Latenz mindestens folgende Bandbreite bieten:

- WL = Gewicht der Warteschlange mit niedriger Latenz.
- W0, W1, ... Wn = Gewichtungen der regulären DRR-Warteschlangen.
- n = Anzahl der regulären DRR-Warteschlangen, die für diese Schnittstelle verwendet werden.
- BW = Bandbreite der Verbindung.

Für den alternativen Prioritätsmodus muss die Mindestbandbreite der Warteschlange mit niedriger Latenz =  $BW * n * WL / (n * WL + \text{Sum}(W0, Wn))$ .

Das Gewicht ist der einzige variable Parameter in MDRR, der konfiguriert werden kann. Sie beeinflusst die relative Bandbreite, die eine Datenverkehrsklasse verwenden kann, und den Anteil des Datenverkehrs, der in einer Reihe gesendet wird. Die Verwendung größerer Gewichtungen bedeutet, dass der gesamte Zyklus länger dauert und möglicherweise die Latenz erhöht.

### Konfigurationsrichtlinien

- Es ist besser, das Gewicht der Klasse mit der niedrigsten Bandbreitenanforderung 1 zu konfigurieren, um die Verzögerung und den Jitter so gering wie möglich unter den anderen Klassen zu halten.
- Wählen Sie möglichst niedrige Gewichtungswerte aus. Beginnen Sie mit einem Gewicht von 1 für die Klasse mit der niedrigsten Bandbreite. Wenn Sie z. B. zwei Klassen mit 50 % Bandbreite für jede Klasse verwenden, müssen Sie 1 und 1 konfigurieren. Die Verwendung von 10 und 10 ist nicht sinnvoll, da die Leistung bei Auswahl von 1 nicht beeinträchtigt wird. Außerdem führt ein höheres Gewicht zu mehr Jitter.
- Ein niedriger Gewichtswert für die LLQ ist sehr wichtig, insbesondere im alternativen Modus, um den anderen Klassen nicht zu viel Verzögerung oder Jitter hinzuzufügen.

### [MDRR-Beispiel](#)

Das Beispiel in diesem Abschnitt stammt aus der *Cisco IOS® Software Architecture*, Cisco Press.

Nehmen wir an, wir haben drei Warteschlangen:

- **Warteschlange 0:** hat eine Menge von 1500 Byte; Es ist die Warteschlange mit niedriger Latenz, die für den Betrieb im alternativen Modus konfiguriert ist.
- **Warteschlange 1** - hat eine Menge von 3000 Byte.
- **Warteschlange 2** - hat eine Menge von 1.500 Byte.

[Abbildung 1](#) zeigt den Anfangsstatus der Warteschlangen sowie einige Pakete, die empfangen und in die Warteschlange gestellt wurden.

### Abbildung 1: Erstzustand des 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	

Die Warteschlange 0 wird zuerst bedient, ihr Quantum wird dem Defizitzähler hinzugefügt, Packet 1 mit 250 Byte wird übertragen, und seine Größe wird vom Defizitzähler abgezogen. Da der Defizitzähler der Warteschlange 0 immer noch größer als 0 ( $1500 - 250 = 1250$ ) ist, wird auch Paket 2 übertragen, und seine Länge wird vom Defizitzähler abgezogen. Der Defizitzähler der Warteschlange 0 ist jetzt  $-250$ , also wird Warteschlange 1 als Nächstes bedient. [Abbildung 2](#) zeigt diesen Zustand.

Abbildung 2: MDRR-Folgezustand

Queues					Deficit Counters			
Queue 0	-----				3/250	-250		
Queue 1	-----				6/1500	5/1500	4/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250		0	

Der Defizitzähler der Warteschlange 1 ist auf 3000 ( $0 + 3000 = 3000$ ) festgelegt, und die Pakete 4 und 5 werden übertragen. Bei jedem übertragenen Paket wird die Größe des Pakets vom Defizitzähler abgezogen, sodass der Defizitzähler der Warteschlange 1 auf 0 reduziert wird. [Abbildung 3](#) veranschaulicht diesen Zustand.

Abbildung 3: MDRR-Status, wenn der Zähler für das Defizit der Warteschlange 1 Null ist

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

Wir müssen vom alternativen Prioritätsmodus in die Service-Warteschlange 0 zurückkehren. Auch hier wird der Quantenwert zum aktuellen Defizitzähler hinzugefügt, und der Defizitzähler der Warteschlange 0 ist auf das Ergebnis festgelegt ( $-250 + 1500 = 1250$ ). Paket 3 wird jetzt übertragen, da der Defizitzähler größer als 0 und die Warteschlange 0 jetzt leer ist. Wenn eine Warteschlange geleert wird, ist ihr Defizitzähler auf 0 gesetzt, wie in [Abbildung 4](#) gezeigt.

Abbildung 4: MDRR-Status beim Leeren einer Warteschlange

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

Als Nächstes wird die Warteschlange 2 gewartet. Der Defizitzähler ist auf 1500 ( $0 + 1500 = 1500$ ) festgelegt. Die Pakete 7 bis 10 werden übertragen, sodass der Defizitzähler bei 500 ( $1500 - (4 \cdot 250) = 500$ ) liegt. Da der Defizitzähler immer noch größer als 0 ist, wird auch Paket 11 übertragen.

Wenn Paket 11 übertragen wird, ist die Warteschlange 2 leer, und der Defizitzähler ist auf 0 festgelegt, wie in [Abbildung 5](#) gezeigt.

Abbildung 5: MDRR-Status bei Übertragung von Paket 11

Queues		Deficit Counters
Queue 0		0
Queue 1	6/1500	0
Queue 2		0

Als Nächstes wird die Warteschlange 0 wieder gewartet (da wir uns im alternativen Prioritätsmodus befinden). Da es leer ist, werden als Nächstes die Warteschlange 1 und Paket 6 übertragen.

## [MDRR-Unterstützung nach Modultyp](#)

Die Cisco Serie 12000 unterstützt fünf Linecard-Modelle mit einzigartigen Layer 3 (L3) Forwarding Engine-Architekturen. Die Unterstützung für MDRR variiert je nach L3-Modultyp und Kartentyp. Beispielsweise wird MDRR und WRED auf Engine 0 ATM Line Cards nicht unterstützt. Mit dem Befehl **show diag** können Sie den L3-Modultyp Ihrer installierten Linecards ermitteln:

```
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 auf ToFab \(Rx\)-Warteschlangen](#)

Sie können entweder die "Legacy CoS-Syntax" oder die "Modular QoS Command Line Interface"

verwenden, um MDRR für die Cisco Serie 12000 zu konfigurieren. In den späteren Abschnitten dieses Dokuments wird die Konfiguration von MDRR mit Legacy-CoS oder modularer QoS erläutert. Sie müssen die ToFab-Warteschlangen nur mit der Legacy-CoS-Syntax konfigurieren, da sie die neuere Syntax des MQC nicht unterstützen. Einzelheiten finden Sie in [Tabelle 2](#).

**Tabelle 2: Details zur MDRR in ToFab (Rx)-Warteschlangen**

	Wo implementiert	ToFab-MDRR	ToFab Alternative PQ	ToFab Strict PQ	ToFab WRED
Eng0	Software	Nein**	Nein**	Ja	Ja
Eng1	-	Nein	Nein	Nein	Nein
Eng2	Hardware	Ja	Ja	Ja	Ja
Eng3	Hardware	Nein	Ja	Ja	Ja
Eng4	Hardware	Ja	Ja	Ja	Ja
Eng4+	Hardware	Ja	Ja	Ja	Ja

\*\* MDRR wird auf Engine 0-LCs in ToFab-Richtung (Rx) unterstützt, jedoch nur im Modus mit strikter Priorität, nicht im alternativen Prioritätsmodus. Die sieben verbleibenden Warteschlangen werden wie gewohnt unterstützt.

An den eingehenden Schnittstellen wird eine separate virtuelle Ausgabewarteschlange pro Ziel-LC verwaltet. Wie diese Warteschlangen implementiert werden, hängt vom L3-Modultyp ab.

- Engine 0 - Eingehende LCs verwalten acht Warteschlangen pro Zielsteckplatz. Die maximale Anzahl der Warteschlangen beträgt  $16 \times 8 = 128$ . Jede Warteschlange kann separat konfiguriert werden.
- Engines 2, 3, 4 und 4+: Eingehende LCs verwalten acht Warteschlangen pro Zielschnittstelle. Bei 16 Zielsteckplätzen und 16 Schnittstellen pro Steckplatz beträgt die maximale Anzahl von Warteschlangen  $16 \times 16 \times 8 = 2048$ . Alle Schnittstellen in einem Zielsteckplatz müssen dieselben Parameter verwenden.

### [MDRR auf FrFab \(Tx\)-Warteschlangen](#)

Die MDRR in den FrFab-Warteschlangen funktioniert unabhängig von der Hardware oder Software konsistent. Alle L3-Modultypen unterstützen acht Klassenwarteschlangen für jede ausgehende Schnittstelle. Einzelheiten finden Sie in [Tabelle 3](#).

**Tabelle 3: Details zur MDRR in FrFab (Tx)-Warteschlangen**

	Wo implementiert	FrFab Alternative PQ	FrFab Strict PQ	FrFab WRED
Eng0	Software <sup>1</sup>	Nein	Ja	Ja
Eng1	-	Nein	Nein	Nein

Eng2	Hardware	Ja <sup>2</sup>	Ja	Ja
Eng3	Hardware	Nein	Ja	Ja
Eng4	Hardware	Ja	Ja	Ja
Eng4+	Hardware	Ja	Ja	Ja

<sup>1</sup> Unterstützung für MDRR in FrFab-Warteschlangen von Engine 0-LCs wird in den folgenden Versionen der Cisco IOS-Software eingeführt:

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

<sup>2</sup> Sie müssen alternative MDRR in FrFab-Richtung mit der vorhandenen CoS-Syntax konfigurieren.

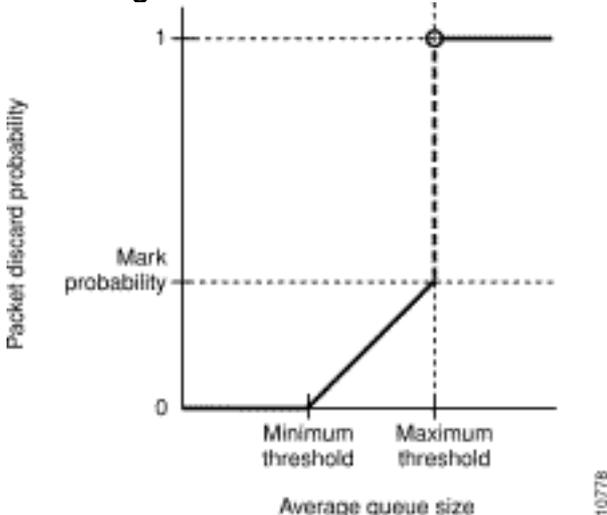
**Hinweis:** Der 3xGE LC unterstützt MDRR in den ToFab-Warteschlangen und, wie in der Cisco IOS Software Release 12.0(15)S, in den FrFab-Warteschlangen mit zwei Einschränkungen, nämlich einem festen Quantum und einer einzigen CoS-Warteschlange für jede Schnittstelle. Die Prioritätswarteschlange unterstützt einen Quantenwert, der konfiguriert werden kann, sowie den strikten und den alternativen Prioritätsmodus. Alle drei Schnittstellen verwenden einen einzigen PQ.

**Hinweis:** In den Versionshinweisen für Cisco Router der Serie 12000 finden Sie aktuelle Informationen zu den unterstützten QoS-Funktionen der Cisco LCs der Serie 12000.

## WRED-Übersicht

Weighted Random Early Detection (WRED) wurde entwickelt, um die schädlichen Auswirkungen von Schnittstellenüberlastungen auf den Netzwerkdurchsatz zu verhindern.

Abbildung 6: WRED-Paketverlust-Wahrscheinlichkeit



Eine Erläuterung der WRED-Parameter finden Sie unter [Weighted Random Early Detection auf dem Cisco Router der Serie 12000](#). Die Mindest-, Höchstwert- und Kennzeichnungsparameter beschreiben die tatsächliche ROTE-Kurve (Random Early Detection). Liegt der gewichtete Warteschlangendurchschnitt unter dem Mindestwert, werden keine Pakete verworfen. Liegt der gewichtete Warteschlangendurchschnitt über dem maximalen Warteschlangenschwellenwert,

werden alle Pakete verworfen, bis der durchschnittliche Wert unter den maximalen Schwellenwert fällt. Wenn der Durchschnitt zwischen dem Mindest- und dem Höchstwert liegt, kann die Wahrscheinlichkeit, dass das Paket verworfen wird, durch eine gerade Linie vom Mindestschwellenwert (die Wahrscheinlichkeit des Verfalls beträgt 0) bis zum maximalen Grenzwert (die Verlustrisiko ist gleich dem 1/Mark Wahrscheinlichkeitsbezeichner) berechnet werden.

Der Unterschied zwischen ROT und WRED besteht darin, dass WRED Datenverkehr mit niedrigerer Priorität selektiv verwerfen kann, wenn die Schnittstelle überlastet wird, und differenzierte Leistungsmerkmale für verschiedene CoS-Klassen bereitstellen kann. Standardmäßig verwendet WRED für jedes Gewicht ein anderes ROT-Profil (IP-Rangfolge: 8 Profile). Es verwirft weniger wichtige Pakete aggressiver als wichtigere Pakete.

Die Anpassung der WRED-Parameter zur Verwaltung der Warteschlangentiefe ist eine komplexe Herausforderung. Sie hängt von zahlreichen Faktoren ab, darunter:

- Datenverkehrslast und -profil wurden angeboten.
- Verhältnis von Last und verfügbarer Kapazität
- Verhalten des Datenverkehrs bei Überlastung.

Diese Faktoren variieren je nach Netzwerk und hängen wiederum von den angebotenen Services und den Kunden ab, die diese Services nutzen. Daher können wir keine Empfehlungen für spezifische Kundenumgebungen abgeben. In [Tabelle 4](#) werden jedoch allgemein empfohlene Werte basierend auf der Bandbreite der Verbindung beschrieben. In diesem Fall unterscheiden wir nicht zwischen den verschiedenen Serviceklassen die Verwerfungsmerkmale.

**Tabelle 4 - Empfohlene Werte basierend auf der Bandbreite der Verbindung**

Bandbreite	Theoretische Bandbreite (Kbit/s)	Physische Bandbreite <sup>1</sup> (Kbit/s)	Mindestschwelle	Maximaler Grenzwert
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

<sup>1</sup> physischer SONET-Satz

Es gibt mehrere Einschränkungen, die bei der Berechnung der oben genannten Schwellenwerte berücksichtigt werden. Wenn beispielsweise die Verbindungsauslastung maximiert und die durchschnittliche Warteschlangentiefe minimiert wird, muss der Unterschied zwischen Maximum und Minimum eine Leistung von zwei betragen (aufgrund von Hardware-Beschränkungen). Basierend auf Erfahrung und Simulation beträgt die maximale momentane Tiefe einer von RED gesteuerten Warteschlange weniger als 2 MaxTh. Für OC48 und höher, 1 MaxTh usw. Die genaue Bestimmung dieser Werte geht jedoch über den Rahmen dieses Dokuments hinaus.

**Hinweis:** Der Wert für die exponentielle Gewichtungskonstante muss nicht auf Line Cards der Engine 2 und höher konfiguriert werden, da stattdessen die Abfrage von Hardware-Warteschlangen verwendet wird. Für LCs der Engine 0 können folgende Werte konfiguriert werden:

- D3 - 9

- oc3-10
- Oktober 2012 - 12

**Hinweis:** WRED wird von Engine 1-LCs nicht unterstützt.

Wie in den folgenden Abschnitten erläutert, können Sie WRED sowohl mit der älteren CoS-Syntax als auch mit der neueren MQC-Syntax konfigurieren.

## Verwenden der älteren CoS-Syntax für die Konfiguration

Die Syntax der Cisco 12000 Class of Service (CoS) verwendet eine **cos-queue-group**-Vorlage, um eine Reihe von CoS-Definitionen zu definieren. Anschließend wenden Sie die Vorlage den ToFab- bzw. FrFab-Warteschlangen an ein- bzw. ausgehenden Schnittstellen an.

### Schritt 1: Definieren einer cos-queue-group

Der Befehl **cos-queue-group** erstellt eine benannte Vorlage mit MDRR- und WRED-Parametern. Die verfügbaren Konfigurationsparameter der CLI sind wie folgt:

```
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

Mit MDRR können Sie die IP-Priorität MDRR-Warteschlangen zuordnen und die spezielle Warteschlange mit niedriger Latenz konfigurieren. Sie können den Rangfolgeparameter unter dem Befehl **cos-queue-group** für Folgendes verwenden:

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

Sie können einer normalen MDRR-Warteschlange (Warteschlange 0 bis 6) eine bestimmte IP-Priorität zuordnen oder sie der Prioritätswarteschlange zuordnen. Mit dem obigen Befehl können Sie derselben Warteschlange mehrere IP-Präzedenzfälle zuordnen.

**Hinweis:** Es wird empfohlen, für die Warteschlange mit niedriger Latenz die Priorität 5 zu verwenden. Für Routing-Updates wird die Rangfolge 6 verwendet.

Sie können jeder MDRR-Warteschlange ein relatives Gewicht zuweisen, wobei eine der Warteschlangen in der Gruppe als Prioritätswarteschlange definiert ist. Sie können den **Befehl queue** unter der **cos-queue-group** dazu verwenden:

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

Verwenden Sie den Befehl **cos-queue-group**, um beliebige WRED-Parameter zu definieren:

```
random-detect-label
```

Hier ein Beispiel für eine **cos-queue-group** mit dem Namen **oc12**. Sie definiert drei Datenverkehrsklassen: Warteschlange 0, 1 und niedrige Latenz. Er ordnet der Warteschlange 0 bis 3, der Warteschlange 0 die Prioritätswerte 4, 6 und 7 der Warteschlange 1 und der Warteschlange mit niedriger Latenz die Prioritätswerte 5 zu. Warteschlange 1 wird mehr Bandbreite zugewiesen.

### Konfigurationsbeispiel

```
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.
```

## Schritt 2: Erstellen von Steckplatztabelle für ToFab-Warteschlangen

Um eine Head-of-Line-Blockierung zu vermeiden, verwalten eingehende Schnittstellen der Cisco Serie 12000 pro Zielsteckplatz eine virtuelle Ausgabewarteschlange. Gehen Sie mit dem Attach-Befehl zu einer Linecard, und führen Sie den Befehl `show controller tofab queue` aus (oder geben Sie direkt den Befehl `show controller tofab queue` im Execute-on-Steckplatz 0 ein), um diese virtuellen Ausgabewarteschlangen anzuzeigen. Nachfolgend finden Sie Beispiele für die direkt von der LC-Konsole erfasste Ausgabe. Siehe [How To Read the Output of the show controller tofab Queue Commands auf einem Cisco Internet Router der Serie 1200](#).

```
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:**

Slot	Dest			
0	0	0	0	65535
1	0	0	0	65535
2	0	0	0	65535
3	0	0	0	65535
4	0	0	0	65535
5	0	17282	0	65535
6	0	0	0	65535
7	0	75	0	65535
8	0	0	0	65535
9	0	0	0	65535
10	0	0	0	65535
11	0	0	0	65535
12	0	0	0	65535
13	0	0	0	65535
14	0	0	0	65535
15	0	0	0	65535
<b>Multicast</b>	0	0	0	65535

LC-Slot1#

Verwenden Sie den Befehl **slot-table-cos**, um eine benannte **cos-queue-Gruppe** einer virtuellen Ausgabewarteschlange zuzuordnen. Sie können eine eindeutige **cos-queue-group-Vorlage** für jede Ausgabewarteschlange konfigurieren.

```
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
```

**Hinweis:** In der obigen Konfiguration werden drei Vorlagen mit den Namen oc48, oc12 und oc3 verwendet. Die Konfiguration für die **cos-queue-group** mit dem Namen oc12 ist in Schritt 1 dargestellt. Konfigurieren Sie oc3 und oc48. Es wird empfohlen, je nach Bandbreite und Anwendung eine eindeutige Vorlage auf eine Reihe von Schnittstellen anzuwenden.

### Schritt 3: Anwenden eines Steckplatztabellencodes auf eine eingehende Schnittstelle

Verwenden Sie den Befehl **rx-cos-slot**, um eine **Steckplatztabelle-cos** auf eine LC-Einheit anzuwenden.

```
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
```

### Schritt 4: Anwenden einer cos-queue-group auf eine ausgehende Schnittstelle

Die Cisco Serie 12000 verwaltet eine separate Warteschlange pro ausgehender Schnittstelle. Um diese Warteschlangen anzuzeigen, fügen Sie sie an die Linecard-CLI an. Verwenden Sie den Befehl **Attach**, und führen Sie dann den Befehl **show controller fab queue** aus, wie hier gezeigt:

```
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
```

Verwenden Sie den Befehl **tx-cos**, um eine **cos-queue-group**-Vorlage auf eine Schnittstellenwarteschlange anzuwenden. Wie hier gezeigt, wenden Sie den Parameter direkt auf die Schnittstelle an. Es sind keine Tabellen erforderlich. In diesem Beispiel ist *pos48* der Name eines Parametersatzes.

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

Bestätigen Sie Ihre Konfiguration mit dem Befehl **show cos**:

```
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-erfab !--- 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
```

**Hinweis:** Die Legacy-CLI verwendet auch die Rangfolge-Syntax für MPLS-Datenverkehr (Multiprotocol Label Switching). Der Router behandelt die MPLS-Bits wie IP Type of Service (ToS)-Bits und legt die entsprechenden Pakete in die richtigen Warteschlangen. Dies trifft bei MQC überhaupt nicht zu. MPLS QoS wird in diesem Dokument nicht behandelt.

## [Modular QoS CLI \(MQC\) für Konfiguration verwenden](#)

Das Ziel der modularen QoS-CLI (MQC) von Cisco besteht darin, alle verschiedenen QoS-Funktionen logisch miteinander zu verbinden, um die Konfiguration der Quality of Service (QoS)-Funktionen der Cisco IOS-Software zu vereinfachen. Die Klassifizierung erfolgt beispielsweise getrennt von Warteschlangen, Richtlinien und Shaping. Es bietet ein einheitliches, vorlagenbasiertes Konfigurations-Framework für QoS. Im Folgenden sind einige Punkte zur MQC-Konfiguration aufgeführt:

- Sie kann problemlos auf eine Schnittstelle angewendet und aus dieser entfernt werden.
- Sie kann problemlos wiederverwendet werden (dieselbe Richtlinie kann auf mehrere Schnittstellen angewendet werden).
- Es bietet ein zentrales Konfigurations-Framework für QoS, das die einfache Bereitstellung, Überwachung und Fehlerbehebung ermöglicht.
- Sie bietet eine höhere Abstraktionsstufe.
- Sie ist plattformunabhängig.

Auf der Cisco Serie 12000 können MQC-Befehle anstelle der älteren CoS-Syntax (Class of Service) verwendet werden.

Die MQC-Unterstützung für die Cisco Serie 12000 bedeutet nicht, dass das gleiche QoS-Feature-Set auf einer anderen Plattform wie der Cisco Serie 7500 jetzt auch für die Cisco Serie 12000 verfügbar ist. Die MQC stellt eine gemeinsame Syntax bereit, bei der ein Befehl zu einer gemeinsamen Funktion oder einem gemeinsamen Verhalten führt. Beispielsweise implementiert der Befehl **bandwidth** eine garantierte Mindestbandbreite. Die Cisco Serie 12000 verwendet

MDRR als Planungsmechanismus für die Bandbreitenreservierung, während die Cisco Serie 7500 WFQ verwendet. Der Hauptalgorithmus ergänzt die jeweilige Plattform.

Wichtig ist, dass nur die FrFab-Warteschlangen die Konfiguration von QoS-Funktionen über den MQC unterstützen. Da es sich bei den ToFab-Warteschlangen um virtuelle Ausgabewarteschlangen und nicht um echte Eingangswarteschlangen handelt, werden sie vom MQC nicht unterstützt. Sie müssen mit älteren CoS-Befehlen konfiguriert werden.

[In Tabelle 5](#) ist die Unterstützung für MQC pro L3-Modultyp aufgeführt.

**Tabelle 5: Unterstützung von MQC für L3-Modultypen**

Typ des L3-Moduls	Modul 0	Modul 1	Modul 2	Modul 3	Modul 4	Modul 4+
MQC-Unterstützung	Ja	Nein	Ja	Ja	Ja	Ja
IOS-Version	12,0(15)S	-	12,0(15)S <sup>1</sup>	12,0(21)S	12,0(22)S	12,0(22)S

<sup>1</sup>Beachten Sie folgende Ausnahmen mit MQC-Unterstützung für Line Cards der Engine 0 und 2 (LC):

- 2xCHOC3/STM1 - Einführung in 12.0(17)S.
- 1xOC48 DPT - Einführung in 12.0(18)S.
- 8xOC3 ATM - geplant für 12.0(22)S.

Der MQC verwendet die folgenden drei Schritte, um eine QoS-Richtlinie zu erstellen:

1. Definieren Sie eine oder mehrere Datenverkehrsklassen mit dem Befehl **class-map**.
2. Erstellen Sie eine QoS-Richtlinie mit dem Befehl **policy-map** und weisen Sie QoS-Aktionen wie **Bandbreite** oder **Priorität** einer benannten Datenverkehrsklasse zu.
3. Verwenden Sie den Befehl **service-policy**, um eine Richtlinienzuordnung an die FrFab-Warteschlange einer ausgehenden Schnittstelle anzuhängen.

Verwenden Sie den Befehl **show policy-map interface**, um Ihre Richtlinie zu überwachen.

Weitere Informationen finden Sie unter [Übersicht über die modulare Quality of Service-Befehlszeilenschnittstelle](#).

### [Schritt 1: Definieren von Klassenzuordnungen](#)

Mit dem Befehl **class-map** werden Verkehrsklassen definiert. Intern weist der Befehl **class-map** auf der Cisco Serie 12000 einer bestimmten CoS-Warteschlange auf der Linecard eine Klasse zu (weitere Einzelheiten siehe [Schritt 4](#)).

Der Befehl **class-map** unterstützt "match-any", bei dem Pakete, die einer der Übereinstimmungsanweisungen entsprechen, in die Klasse eingefügt werden, und "match-all", bei

dem Pakete nur dann in diese Klasse eingefügt werden, wenn alle Anweisungen true sind. Diese Befehle erstellen eine Klasse mit dem Namen "Prec\_5" und klassifizieren alle Pakete mit einer IP-Priorität von 5 für diese Klasse:

```
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
```

[In Tabelle 6](#) sind die für jeden L3-Modultyp unterstützten Anpassungskriterien aufgeführt.

**Tabelle 6 - Unterstützte Übereinstimmungskriterien für L3-Engines**

	Engine 0, 2	Modul 3	Modul 4	Modul 4+
IP-Rangfolge	Ja	Ja	Ja	Ja <sup>1</sup>
Zugriffsgruppe	Nein	Ja	Nein	Nein
MPLS Exp	Nein	Ja	Nein	Ja (12.0.26S)
ip dscp	Nein	Ja	Nein	Ja (12.0.26S)
QoS-Gruppe	Nein	Ja	Nein	Nein
Übereinstimmung-Eingangsschnittstelle POS x/y	Nein	Ja (nur Empfangsrichtlinie)	Nein	Nein

<sup>1</sup> Eingang/Ausgang seit 12.0.26S

### Schritt 2 - Erstellen einer Richtlinienzuordnung

Der Befehl **policy-map** wird verwendet, um einer oder mehreren definierten Klassen Paketverwaltungsrichtlinien oder -aktionen zuzuweisen. Beispiel: Wenn Sie eine Bandbreitenreservierung zuweisen oder ein zufälliges Drop-Profil anwenden.

Die Cisco Serie 12000 unterstützt eine Teilmenge von MQC-Funktionen, die auf der

Hochgeschwindigkeits-Architektur der L3 Engines basieren. [In Tabelle 7](#) sind die unterstützten Befehle aufgelistet:

**Tabelle 7 - Unterstützte Befehle**

Command	Beschreibung
Bandbreite	Garantiert eine Mindestbandbreite in Zeiten von Überlastungen. Sie wird als Prozentsatz der Verbindungsgeschwindigkeit oder als absoluter Wert angegeben. Wenn eine Klasse keine Bandbreite verwendet oder benötigt, die der reservierten Kbit/s entspricht, kann die verfügbare Bandbreite von anderen Bandbreitenklassen verwendet werden.
Polizei, Form	Begrenzt die Datenverkehrsmenge, die eine Klasse übertragen kann. Diese Befehle unterscheiden sich in der Funktion geringfügig. Der Befehl <b>Police</b> identifiziert den Datenverkehr, der die konfigurierte Bandbreite überschreitet, und sendet oder bemerkt ihn. Der <b>shape</b> -Befehl puffert jeglichen überzähligen Datenverkehr und plant die Übertragung mit konstanter Geschwindigkeit, verwirft jedoch keine Änderungen oder Anmerkungen.
Warteschlangenlimit	Weist einer bestimmten Datenverkehrsklasse eine festgelegte Warteschlangenlänge zu. Sie können dies als Anzahl von Paketen angeben, die in der Warteschlange gespeichert werden können.
Priorität	Identifiziert eine Warteschlange als Warteschlange mit niedriger Latenz. MQC unterstützt den strikten Modus nur für einen PQ. Der alternative Modus wird durch MQC nicht unterstützt. Verwenden Sie den Befehl <b>priority</b> ohne Prozentwert, um den Modus strict priority zu aktivieren. <b>Hinweis:</b> Die Implementierung des <b>Prioritätsbefehls</b> auf der Cisco Serie 12000 unterscheidet sich von der Implementierung auf anderen Routern, auf denen Cisco IOS-Software ausgeführt wird. Auf dieser Plattform ist der Prioritätsverkehr nicht auf den konfigurierten Kbit/s-Wert in Zeiten von Überlastung beschränkt. Daher müssen Sie auch den <b>Polizei</b> -Befehl konfigurieren, um zu begrenzen, wie viel Bandbreite eine Prioritätsklasse nutzen kann, und um eine angemessene Bandbreite für andere Klassen sicherzustellen. Derzeit wird der

	<b>Polizeibefehl</b> nur auf Line Cards der Engine 3 unterstützt. Auf den anderen Engine Line Cards ist nur die Standardklasse zulässig, wenn Sie eine Prioritätsklasse konfigurieren.
zufällig erkennen	Weist ein WRED-Profil zu. Verwenden Sie den Befehl <b>random-detect priority</b> , um nicht standardmäßige WRED-Werte pro IP-Rangfolgewert zu konfigurieren.

Auf Engine 3-LCs müssen Sie die FrFab-Warteschlangen mit der Modular QoS CLI (MQC) konfigurieren. Die ältere Befehlszeilenschnittstelle (CLI) wird nicht unterstützt.

Wenn Sie den Befehl **bandwidth** konfigurieren, beachten Sie, dass die LCs Engine 0 und 2 nur sechs Bandbreitenklassen unterstützen. Eine siebte Klasse kann für Dienste mit niedriger Latenz verwendet werden, und eine achte Klasse (Class-Default) wird für den gesamten nicht übereinstimmenden Verkehr verwendet. Sie haben also insgesamt acht Warteschlangen. Klassenstandard wird nicht als Prioritätsklasse verwendet.

Auf Engine 3-LCs wird der Befehl **bandwidth percent** in einen Kbit/s-Wert umgewandelt, der je nach der zugrunde liegenden Verbindungsrate variiert und dann direkt in der Warteschlange konfiguriert wird. Die Genauigkeit dieser garantierten Mindestbandbreite beträgt 64 Kbit/s.

Obwohl mit dem Befehl **bandwidth** keine Umwandlung in einen Quantenwert vorgenommen wird, haben alle Warteschlangen einen Quantenwert. Bei Modul-3-LCs wird der Quantenwert intern basierend auf der maximalen Übertragungseinheit (Maximum Transmission Unit, MTU) der Schnittstelle festgelegt und für alle Warteschlangen gleichermaßen festgelegt. Es gibt keinen MQC-CLI-Mechanismus zum Ändern dieses Quantenwerts, weder direkt noch indirekt. Der Quantenwert muss größer oder gleich der MTU der Schnittstelle sein. Intern liegt der Quantenwert in Einheiten von 512 Byte. Bei einer MTU von 4470 Byte muss der minimale Quantenwert der MTU also 9 sein.

### [MDRR auf Engine 3 LC](#)

Dieser Abschnitt enthält Konfigurationshinweise für die Implementierung von WRED und MDRR auf Modul 3-LCs.

- Die in der CLI konfigurierte MDRR-Bandbreite wird in eine Menge umgewandelt, die L2 entspricht (z. B. wird der L1-Overhead entfernt). Dieser Betrag wird dann auf die nächsten 64 Kbit/s aufgerundet und in Hardware programmiert.
- Für eine Klasse werden drei verschiedene WRED-Profile unterstützt.
- Der WRED (Maximum threshold - minimum threshold) ist ungefähre Leistung von 2. Der Mindestschwellenwert wird dann automatisch angepasst, während der Höchstwert unverändert bleibt.
- Der Wert für die Mark-Wahrscheinlichkeit 1 wird unterstützt.
- Exponentielle Konfiguration von Gewichtungskonstanten wird nicht unterstützt.
- IP Precedence, MPLS EXP-Bits und DSCP-Werte werden unterstützt.

**Hinweis:** Jedem Port oder Kanal auf der Tetra (4GE-SFP-LC= ) oder CHOC12/DS1-IR-SC= Frostbite Linecards sind standardmäßig vier Warteschlangen zugewiesen. Die vier Warteschlangen bestehen aus:

- Eine Klasse für Prioritätswarteschlangen (LLQ)
- Eine Standard-Warteschlangenklasse
- Zwei normale, nicht priorisierte Klassen

Wenn eine Dienststrichlinie angewendet wird, die mehr als diese vier Klassen (1 HPQ, 2 LPQs und Klassenstandard) auf die Schnittstelle enthält, wird folgender Fehler gemeldet:

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

**% Verfügbare Warteschlangenressourcen reichen nicht aus, um die Anforderung zu erfüllen.**

Ab 12.0(26)S wurde für die 4GE-SFP-LC= Tetra-Linecard ein Befehl hinzugefügt, mit dem anstelle von vier acht Warteschlangen/VLANs konfiguriert werden können. Die acht Warteschlangen bestehen aus:

- Eine LLQ
- Eine Standard-Warteschlange
- Sechs normale Warteschlangen

Für die Verwendung dieses Befehls ist ein Neuladen der Linecard mit Mikrocode erforderlich, sodass anstelle von 1022 nur 508 VLANs konfiguriert werden können. Die Befehlssyntax ist wie folgt:

**[no] hw-module-Steckplatz <steckplatz#> qos-Schnittstellenwarteschlangen 8**

Beispiele:

**Router(config)#hw-module-Steckplatz 2 QoS-Schnittstellenwarteschlangen 8**

**Warnung: Laden Sie die Linecard neu, damit dieser Befehl wirksam wird.**

**Router(config)#Microcode reload 2**

Dieser Befehl ist für die CHOC12/DS1-IR-SC= Frostbite-Linecard in 12.0(32)S verfügbar.

### **Beispiel 1: Bandbreitenprozentsatz-Befehl**

In diesem Beispiel werden 20 Prozent der verfügbaren Bandbreite dem Datenverkehr der Klasse Prec\_4 und 30 Prozent dem Datenverkehr der Klasse Prec\_3 zugewiesen. Die restlichen 50 Prozent verbleiben bei der Klassenstandardklasse.

Darüber hinaus wird WRED als Drop-Mechanismus für alle Datenklassen konfiguriert.

#### **Beispiel 1: Bandbreitenprozentsatz**

```

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
```

## Beispiel 2: Befehl bandwidth {kbps}

In diesem Beispiel wird veranschaulicht, wie der Bandbreitenbefehl als absoluter Kbit/s-Wert statt als Prozentsatz angewendet wird.

### Beispiel 2: Bandbreite {Kbit/s}

```
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
```

## Beispiel 3: priority Command

Dieses Beispiel wurde für Service Provider entwickelt, die den Cisco Router der Serie 12000 als MPLS-PE-Router (Provider Edge) verwenden und eine QoS-Dienstrichtlinie für die Verbindung zwischen dem PE-Router und dem CE-Router (Customer Edge) konfigurieren müssen. Er platziert IP-Rangfolge-5-Pakete in einer Prioritätswarteschlange und beschränkt die Ausgabe dieser Warteschlange auf 64 Mbit/s. Anschließend wird den Bandbreitenklassen ein Teil der verbleibenden Bandbreite zugewiesen.

Alle Klassenwarteschlangen, die keine Priorität haben, werden mit dem Befehl **random-detect** konfiguriert, um WRED als Drop-Richtlinie zu aktivieren. Für alle Bandbreitenklassen und Klassenstandardwerte muss WRED explizit konfiguriert sein.

### Beispiel 3 - Priorität

```
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
```

### Schritt 3 - Zuweisen einer Richtlinienzuordnung zu einer ausgehenden Schnittstellenwarteschlange

Wie oben erwähnt, funktioniert der MQC nur mit FrFab-Warteschlangen an einer ausgehenden Schnittstelle. Um eine definierte Richtlinienzuordnung anzuwenden, verwenden Sie den Befehl **service-policy output**, wie hier gezeigt:

```
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
```

### Schritt 4: Überwachen und Überprüfen der Service-Richtlinie

Verwenden Sie den Befehl **show policy-map interface**, um die Anwendung einer Richtlinie anzuzeigen. Der Befehl **show policy-map interface** (Richtlinienzuordnung anzeigen) zeigt Folgendes an:

- Konfigurierte Bandbreite und Prioritätsklassen sowie die Anpassungskriterien.
- Alle WRED-Profilen.
- Form und Richtlinienparameter.
- Datenverkehrsabrechnung und -raten
- Die interne CoS-Warteschlange, der eine bestimmte Klasse zugeordnet ist. Auf diese Warteschlangen wird derselbe Index verwiesen, der in der Ausgabe des Befehls **show controller fab queue** verwendet wird.

Im Folgenden finden Sie ein Beispiel für eine vollständige Konfiguration und die Befehle **show** zur Überwachung der Richtlinie:

#### **Vollständige Konfiguration**

```
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.
```

Verwenden Sie den Befehl **show policy-map interface**, um die auf der Schnittstelle konfigurierte Richtlinie zusammen mit allen konfigurierten Klassen anzuzeigen. Hier ist die Ausgabe des Befehls:

```
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
```

## Befehle zur Überwachung von Überlastungsmanagement und -vermeidung

In diesem Abschnitt werden die Befehle aufgelistet, mit denen Sie Ihre Richtlinie für das Überlastungsmanagement und die Vermeidung von Überlastungen überwachen können.

[In Tabelle 8](#) sind die relevanten Befehle für die Eingangs- und Ausgangs-Linecards aufgeführt.

**Tabelle 8 - Befehle für die Linecards**

Ingress Line Card	Ausgangs-Linecard
<ul style="list-style-type: none"> <li>• Anzeigen von Schnittstellen</li> <li>• exec-Steckplatz &lt;x&gt; sh-Controller-ToFab-Warteschlange</li> <li>• exec-Steckplatz &lt;x&gt; Show Controller-ToFab-Warteschlange</li> </ul>	<ul style="list-style-type: none"> <li>• Anzeigen von Schnittstellen</li> <li>• show interfaces &lt;y&gt; random</li> <li>• exec-Steckplatz &lt;y&gt; Show-Controller-Fab-Warteschlange</li> <li>• exec-Steckplatz &lt;y&gt;</li> </ul>

<p>&lt;Steckplatz&gt; &lt;Port&gt;</p> <ul style="list-style-type: none"> <li>• exec-Steckplatz &lt;x&gt;</li> </ul> <p>show-Controller tofab qm stat</p>	<p>show-Controller-Fab- Warteschlange &lt;Port&gt;</p> <ul style="list-style-type: none"> <li>• exec-Steckplatz &lt;y&gt;</li> </ul> <p>show-Controller-Fab QM-Status</p>
---	---

Diese Befehle werden in diesem Abschnitt erläutert.

## [Befehl show interfaces](#)

Bevor Sie diesen Befehl verwenden, bestätigen Sie die richtige "Warteschlangenstrategie". Wenn die Ausgabe First In, First Out (FIFO) anzeigt, stellen Sie sicher, dass der Befehl **service-policy** in der aktuellen Konfiguration angezeigt wird (wenn MQC zum Konfigurieren der MDRR verwendet wurde).

Überwachen Sie die Anzahl der Ausgabeabbrüche. Dies stellt die Gesamtzahl der WRED FrFab-Verwerfungen dar, die bei ausgehendem Datenverkehr an dieser Schnittstelle aufgetreten sind. Die Anzahl der Ausgabeverwerfen in der Befehlsausgabe **show interfaces** muss gleich oder größer sein als die Anzahl der Ausgabeverwerfen in der Ausgabe von **show interfaces <number> random** command.

**Hinweis:** Auf dem Cisco Router der Serie 12000 werden die Ausgabeverwerfen der Schnittstelle aktualisiert, nachdem die WRED-Drops aktualisiert wurden. Wenn Sie ein Tool zum Abfragen beider Drop-Zähler verwenden, besteht die geringe Wahrscheinlichkeit, dass die Schnittstellenverluste noch nicht aktualisiert werden.

```
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
```

## [Der Befehl show interfaces {number} random](#)

Wenn Sie diesen Befehl verwenden, müssen Sie:

- Überprüfen Sie, ob die richtige **cos-queue-group**-Vorlage auf diese Schnittstelle angewendet wird.
- Überprüfen Sie die MDRR-Gewichtungen. Für jede MDRR-Warteschlange können Sie den gewichteten Durchschnitt der Warteschlangenlänge und den höchsten erreichten Wert (in Paketen) überprüfen. Die Werte werden als gewichteter Durchschnitt berechnet und müssen nicht die tatsächlich erreichte maximale Warteschlangentiefe widerspiegeln.
- Überprüfen Sie die Mindest- und Höchstwerte für WRED.
- Überprüfen Sie die Anzahl der zufälligen Verwerfen und Verwerfen von Schwellenwerten für jedes ROTE-Label ("To Fabric"-Verwerfen bedeutet, dass auf allen Linecards die gesamte Anzahl der Verwerfen dieses Labels abgelegt ist).
- Der Zähler "TX-queue-limit drop" wird nur für LCs der Engine 1 verwendet, die WRED nicht unterstützen. Mit Engine 1-Karten können Sie die Grenze der MDRR-Warteschlangen mit dem Befehl **TX-queue-limit interface** festlegen. Wenn WRED unterstützt wird, bestimmen die WRED-Schwellenwerte die Tiefe der MDRR-Warteschlangen.

```
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
```

## Der exec-Steckplatz (y) zeigt Controller-Fab-Warteschlange {port} Befehl.

Dieser Befehl zeigt die momentane Warteschlangentiefe für einen bestimmten Port an einem bestimmten Steckplatz an. In der Beispielausgabe in diesem Abschnitt wird die MDRR-Warteschlange an der Schnittstelle POS 4/1 angezeigt. Sie sehen eine Warteschlangentiefe für die MDRR-Warteschlange 1 von 1964-Paketen. Das Gewicht ist die Anzahl der Bytes, die in dieser Warteschlange verarbeitet werden können. Dieses Gewicht bestimmt den Prozentsatz der Bandbreite, die Sie dieser Warteschlange geben möchten. Das Defizit ist der Wert, der dem DRR-Algorithmus mitteilt, wie viele Pakete noch bedient werden müssen. Wie Sie sehen, befinden sich in der LLQ-Warteschlange (DRR-Warteschlange 7) keine Pakete in der Warteschlange.

```
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
```

Mit diesem Befehl wird insbesondere die Tiefe der Prioritätswarteschlange der Ausgangs-Linecard überwacht. Wenn Sie sehen, dass Pakete bei dieser LLQ zu warten beginnen, ist es ein guter Hinweis, dass Sie einen Teil des Voice over IP (VOIP)-Datenverkehrs an andere Ausgangs-Linecards umleiten müssen. Bei einem guten Design sollte die Länge immer 0 oder 1 sein. In einem realen Netzwerk wird selbst bei Sprachdaten ein Datenverkehrsaufkommen von Spitzenzeiten verursacht. Die zusätzliche Verzögerung wird schwerwiegender, wenn die gesamte Sprachlast für kurze Zeit 100 % der Ausgangsbandbreite überschreitet. Der Router kann nicht mehr Datenverkehr auf die Leitung übertragen, als zugelassen ist. Der Sprachverkehr wird daher in die Warteschlange mit eigener Priorität gestellt. Dadurch entstehen durch den Burst des Sprachdatenverkehrs selbst Sprachlatenz und Sprachjitter.

```
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
```

Warteschlange 7 ist die LLQ, und die Länge gibt an, wie viele Pakete in dieser LLQ enthalten sind.

## Der exec-Steckplatz (y) zeigt den Controller-Fab-QM-Stat-Befehl.

Verwenden Sie diesen Befehl, wenn Sie vermuten, dass der Paketspeicher eines LC die volle Kapazität erreicht. Ein zunehmender Wert für den Zähler "no mem drop" legt nahe, dass WRED nicht konfiguriert ist oder dass die WRED-Schwellenwerte zu hoch festgelegt sind. Dieser Zähler darf unter normalen Bedingungen nicht erhöht werden. Weitere Informationen finden Sie unter [Problembehandlung bei ignorierten Paketen und Speicherverlusten auf dem Cisco Internet Router der Serie 12000](#).

```
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
```

## Überwachung des eingehenden Überlastungsmanagements

In diesem Abschnitt werden die Befehle zum Überwachen des eingehenden Überlastungsmanagements beschrieben.

### Befehl show interfaces

Bevor Sie diesen Befehl ausführen, überprüfen Sie, ob der Wert im ignorierten Zähler erhöht wird. Wenn Ihnen auf der ToFab-Seite der Arbeitsspeicher ausgeht oder die Linecard die Pakete nicht schnell genug akzeptiert, werden ignorierte Pakete angezeigt. Weitere Informationen finden Sie unter [Fehlerbehebung bei Einzugseinträgen auf dem Cisco Internet Router der Serie 12000](#).

```
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

```

## Der exec-Steckplatz (x) zeigt Controller-ToFab-Warteschleife-Befehl

Diese Beispielausgabe des Befehls **exec-Steckplatz <x> show controller tofab queue** wurde erfasst, wenn eine Ausgangs-Linecard in Steckplatz 3 nicht überlastet war.

```

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

```

Die folgende Ausgabe wurde bei Überlastungen an Steckplatz 3 erfasst:

```

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

```

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

Verwenden Sie diesen Befehl, um zu bestimmen, wie viel Speicher auf der ToFab-Seite verwendet wird. Beachten Sie insbesondere die Zahl in der Spalte "#Qelem". Beachten Sie, dass

- Wenn kein Speicher verwendet wird, sind die Werte am höchsten.
- Der Wert der Spalte "#Qelem" nimmt bei der Pufferung von Paketen ab.
- Wenn die Spalte "#Qelem" null erreicht, werden alle geschnitzten Puffer verwendet. Auf Engine 2 LC können kleine Pakete Pufferspeicherplatz von größeren Paketen ausleihen.

Sie können diesen Befehl auch verwenden, um die Anzahl der in einer virtuellen Ausgabewarteschlange in Warteschlangen gespeicherten Pakete zu bestimmen. Das Beispiel hier zeigt, wie Steckplatz 14 auf die aktuelle Anzahl der Pakete in diesen Warteschlangen für Steckplatz 4, Port 1 (POS 4/1) überprüft wird. Es werden 830 Pakete in die Warteschlange 1 der MDRR gestellt.

```

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

## Der exec-Steckplatz (x) zeigt den Controller-Befehl tofab QM stat Command.

Mit diesem Befehl können Sie die Anzahl der ToFab-Drops pro Linecard anzeigen. Überprüfen Sie außerdem, ob ein Zähler für "kein Speicherverlust" inkrementiert ist. Dieser Zähler erhöht sich, wenn CoS nicht auf der ToFab-Seite konfiguriert ist.

```
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
```

## Anwenderbericht

In dieser Fallstudie wird gezeigt, wie eine typische Richtlinie für den Netzwerkkern einer Service-Provider-Umgebung konfiguriert wird. Sie wendet Warteschlangenbefehle an und ermöglicht die Verwendung von MDRR/WRED für das aktive Warteschlangenmanagement. QoS-Richtlinien in Edge-Routern verwenden normalerweise Datenverkehrsmarkierungen, Konditionierung usw., damit Router im Core den Datenverkehr anhand von IP-Rangfolge oder DiffServ Code Point (DSCP)-Werten in Klassen sortieren können. In dieser Fallstudie werden QoS-Funktionen der Cisco IOS-Software verwendet, um enge Service Level Agreements (SLAs) und unterschiedliche Service Level für Sprach-, Video- und Datenservices im selben IP-Backbone zu erfüllen.

Bei diesem Ansatz hat ein Service Provider drei Datenverkehrsklassen implementiert. Die wichtigste ist die LLQ- oder Low Latency Queueing-Klasse. Dies ist die Klasse für Sprache und Video. Diese Klasse muss eine minimale Verzögerung und Jitter aufweisen und darf niemals Paketverluste oder neu sortierte Pakete erleiden, solange die Bandbreite dieser Klasse die Verbindungsbandbreite nicht überschreitet. Diese Klasse wird in der DiffServ-Architektur als Expedited Forwarding Per-Hop Behavior (EF PHB)-Datenverkehr bezeichnet. Der Internet Service Provider (ISP) hat das Netzwerk so konzipiert, dass die durchschnittliche Last der Verbindungsbandbreite 30 % nicht überschreitet. Die beiden anderen Klassen sind die Business-Klasse und die Best-Effort-Klasse.

Im Design haben wir die Router so konfiguriert, dass die Business-Klasse immer 90 % der verbleibenden Bandbreite erhält und die Best Effort-Klasse 10 % erreicht. Diese beiden Klassen verfügen über weniger zeitkritischen Datenverkehr und können Datenverkehrsverluste, höhere Verzögerungen und Jitter verzeichnen. Im Design stehen Line Cards der Engine 2 im Mittelpunkt:

## Line Cards mit 1xOC48 rev B, 4xOC12 rev B und 8xOC3

Rev B-Linecards eignen sich aufgrund einer überarbeiteten ASIC- und Hardware-Architektur am besten für die Übertragung von VoIP-Datenverkehr, was zu einer sehr geringen Latenz führt. Mit dem überarbeiteten ASIC wird die Warteschlange für die Übertragung von FIFO durch den Line Card-Treiber auf etwa das Doppelte der größten MTU auf der Karte vergrößert. Suchen Sie ein "-B", das der Teilenummer beigefügt ist, wie OC48E/POS-SR-SC-B=.

**Hinweis:** Verwechseln Sie die Übertragungs-FIFO-Warteschlange nicht mit den FrFab-Warteschlangen, die auf Engine 0-Linecards mit dem Schnittstellenbefehl tx-queue-limit eingestellt werden können.

[In Tabelle 9](#) sind die Zuordnungskriterien für jede Klasse aufgeführt.

**Tabelle 9 - Übereinstimmende Kriterien für jede Klasse**

Klassenname	Anpassungskriterien
Prioritätswarteschlange - Sprachdatenverkehr	Rangfolge 5
Geschäftswarteschlange	Rangfolge 4
Best Effort Queue	Rangfolge 0

Die Line Cards OC48 können eine große Anzahl von Paketen in den ToFab-Warteschlangen in die Warteschlange stellen. Daher ist es wichtig, MDRR/WRED für die ToFab-Warteschlangen zu konfigurieren, insbesondere wenn die Ausgangsschnittstelle eine Hochgeschwindigkeitsschnittstelle wie OC48 ist. Die Fabric kann den Datenverkehr nur theoretisch mit einer maximalen Geschwindigkeit von 3 Gbit/s (1.500-Byte-Pakete) auf die empfangende Linecard umschalten. Wenn der gesamte gesendete Datenverkehr größer ist als der Datenverkehr, den die Switching-Fabric an die empfangende Karte übertragen kann, werden viele Pakete in die ToFab-Warteschlange gestellt.

```
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

```

Es wird davon ausgegangen, dass je mehr VOIP-Datenverkehr vorhanden ist, desto mehr Geschäftsverkehr warten muss, bis er bedient wird. Dies stellt jedoch kein Problem dar, da das enge SLA keine Paketverluste, eine sehr geringe Latenz und Jitter für die Prioritätswarteschlange erfordert.

## [Zugehörige Informationen](#)

- [Lesen der Ausgabe der Registerkarte "show controller" | Tab-Warteschlangenbefehle auf einem Cisco Internet Router der Serie 12000](#)
- [Fehlerbehebung bei nicht erkannten Paketen und Speicherverlusten beim Cisco Internet Router der Serie 12000](#)
- [Fehlerbehebung bei Eingangsverlusten auf Internetroutern der Cisco 12000 Serie](#)
- [Weighted Random Early Detection auf dem Cisco Router der Serie 12000](#)

- [Modulare Übersicht über die Quality of Service-Befehlszeilenschnittstelle](#)
- [Support-Seite für Internet Router der Serie 1200](#)
- [Technischer Support – Cisco Systems](#)