

Понимание и настройка MDRR/WRED на маршрутизаторе Интернета Cisco 12000 Series

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Обзор MDRR](#)

[Очередь по приоритету на MDRR](#)

[Пример MDRR](#)

[Поддержка MDRR типом модуля](#)

[Обзор WRED](#)

[Используйте синтаксис legacy CoS для конфигурации](#)

[Используйте командную строку Modular QoS CLI \(MQC\) для конфигурации](#)

[Команды для мониторинга управления перегрузками сети и предотвращения](#)

[Команда "show interfaces"](#)

[Show interfaces {номер} случайная Команда](#)

[Очередь frrfab покажите контроллер слота ehex \(y\) {порт} Команда](#)

[Слот ehex \(y\) показывает Команду статистики QM controller frrfab](#)

[Контролируйте входящее управление перегрузками сети](#)

[Команда "show interfaces"](#)

[Команда show controller tofab queue слота ehex \(x\)](#)

[Очередь tofab покажите контроллер слота ehex \(x\) \(слот\) \(порт\) Команда](#)

[Слот ehex \(x\) показывает Команду статистики QM tofab контроллера](#)

[Примеры практического применения](#)

[Дополнительные сведения](#)

Введение

В данном документе описывается, как настроить функции управления перегрузкой и предотвращения перегрузок в ПО Cisco IOS® Интернет-маршрутизатора Cisco серии 12000.

После чтения этого документа необходимо быть в состоянии:

- Поймите, почему важно настроить Modified Deficit Round Robin (MDRR) и Взвешенное

- произвольное раннее обнаружение (WRED) в вашей базовой сети.
- Поймите реализацию, которая лежит в основе MDRR и WRED на Серии Cisco 12000.
 - Настройте MDRR и WRED с помощью синтаксиса унаследованного класса обслуживания (CoS) и модульного QoS CLI (MQC).

Предварительные условия

Требования

Читатели данного документа должны обладать знаниями по следующим темам:

- Глубокие знания с архитектурой Интернет-маршрутизатора Cisco 12000 серии.
- В частности осведомленность об архитектуре организации очереди и этих сроках: ToFab (К матрице) – который описывает очереди стороны получателя на карте входной линии. FrFab (От матрицы) – который описывает очереди стороны передачи на исходящей линейной плате.

Примечание: Также рекомендуется, что вы поиск [Как Считать Выходные данные Команд show controller frfab | tofab queue](#) на Интернет-маршрутизаторе Cisco 12000 серии.

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Все платформы Cisco 12000, которые включают 12008, 12012, 12016, 12404, 12406, 12410, и 12416.
- Программное обеспечение Cisco IOS версии 12.0 (24) S1.

Примечание: Несмотря на то, что конфигурации в этом документе были протестированы на программном обеспечении Cisco IOS версии 12.0 (24) S1, любой Cisco IOS Software Release, который поддерживает Интернет-маршрутизатор Cisco 12000 серии, может использоваться.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

Общие сведения

Методы организации очередей определяют пакетный механизм планирования или заказ, в котором пакеты исключены из очереди к интерфейсу для передачи на физическом проводе. На основе заказа и числа раз, что очередь обслуживается функцией scheduler, методы организации очередей также поддерживают минимальные пропускные способности и низкие задержки.

Важно гарантировать, что пакетный механизм планирования поддерживает архитектуру коммутации, на которой это внедрено. Обслуживание очередей на основе равнодоступности (WFQ) является известным алгоритмом планирования для выделения ресурсов на платформах маршрутизаторов Cisco с на основе шины архитектура. Однако это не поддерживается на Интернет-маршрутизаторе Cisco 12000 серии. Традиционная постановка в очередь с установлением приоритета программного обеспечения Cisco IOS и настраиваемая организация очереди также не поддерживаются. Вместо этого Серия Cisco 12000 использует специальную форму организации очереди, названной Modified Deficit Round Robin (MDRR), который предоставляет гарантии относительной полосы пропускания, а также очередь с низкой задержкой. M MDRR обозначает "модифицированный"; это добавляет очередь с приоритетами по сравнению с DRR, где не присутствует никакая очередь с приоритетами. Для получения дополнительной информации на MDRR, посмотрите раздел [Обзора MDRR](#).

Кроме того, Cisco серии 12000 поддерживает взвешенное случайное раннее обнаружение (WRED) в качестве политики отбрасывания внутри очередей MDRR. Этот механизм предотвращения перегрузок предоставляет альтернативу механизму отбрасывания остатка по умолчанию. Можно избежать перегрузки с помощью управляемых сбросов.

Предотвращение перегрузки и механизмы управления, такие как WRED и MDRR особенно важны на очередях FrFab относительно низкоскоростных исходящих интерфейсов, таковы как канализируемые линейные карты (LC). Высокоскоростная коммутационная фабрика может отправить пакеты группам каналов намного быстрее, чем группы каналов могут передать их. Как организация очереди / буферами управляют на уровне физического порта, противодействие на один канал может повлиять на все другие каналы на том порту. Таким образом очень важно управлять тем противодействием через WRED/MDRR, который ограничивает влиянием противодействия к рассматриваемому каналу (каналам). Для получения дополнительной информации о том, как управлять превышением нормы для исходящего интерфейса, посмотрите [Пропущенные пакеты Устранения проблем и Никакие Отбрасывания Памяти на Интернет-маршрутизаторе Cisco 12000 серии](#).

[Обзор MDRR](#)

Этот раздел предоставляет обзор Modified Deficit Round Robin (MDRR).

С MDRR, настроенным как стратегия организации очереди, непустые очереди подаются один за другим в порядке круговой очереди. Каждый раз, когда очередь подается, фиксированный размер данных исключен из очереди. Алгоритм тогда обслуживает следующую очередь. Когда очередь подается, MDRR отслеживает количество байтов данных, которые были исключены из очереди сверх установленного значения. В следующем проходе, когда очередь подается снова, меньше данных будет исключено из очереди для компенсации избыточные данные, которые подавались ранее. В результате средняя величина данных, исключенных из очереди на очередь, будет близко к установленному значению. Кроме того, MDRR поддерживает очередь с приоритетами, которая подается на привилегированном образе. MDRR объяснен более подробно в этом разделе.

Каждая очередь в MDRR определена двумя переменными:

- Количественное значение – Это - среднее количество байтов, подаваемых в каждом раунде.
- Счетчик дефицита – Это используется для отслеживания, сколько байтов очередь

передала в каждом раунде. Это инициализируется к количественному значению. Пакетами в очереди служат долго, поскольку счетчик дефицита больше, чем нуль. Каждый обслуженный пакет уменьшает значение счетчика дефицита на значение, равное его размеру в байтах. Очередь больше не может обслуживаться после того, как значение счетчика дефицита станет равным нулю или отрицательным. В каждом новый раунд счетчик дефицита каждой непустой очереди увеличен ее количественным значением.

Примечание: В целом квантовый размер для очереди не должен быть меньшим, чем максимальный размер передаваемого блока данных (MTU) интерфейса. Это гарантирует, что планировщик всегда служит по крайней мере одному пакету от каждой непустой очереди.

Каждой очереди MDRR может быть назначен относительный вес, причем одна из очередей группы определяется в качестве приоритетной. Когда интерфейс переполнен, веса назначают относительную полосу пропускания для каждой очереди. Если существуют данные в очереди, чтобы быть переданными, алгоритм MDRR исключает данные из очереди от каждой очереди в порядке круговой очереди.

Если все стандартные очереди MDRR имеют данные, они обслуживаются следующим образом:

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

Во время каждого цикла очередь может исключить квант из очереди на основе своего настроенного веса. На Механизме 0 и линейных картах Engine 2, значение 1 эквивалентно предоставлению интерфейса веса его MTU. Для каждого шага увеличения больше, чем на 1, весовой коэффициент очереди увеличивается на 512 байт. Например, если для конкретного интерфейса установлено значение максимального размера блока данных, равное 4470, и вес очереди, равный 3, при каждом циклическом сдвиге допускается удаление $4470 + (3-1) * 512 = 5494$ байт из очереди. Если две обычных очереди DRR, Q0 и Q1, используются, Q0 настроен с весом 1, и Q1 настроен с весом 9. Если бы обе очереди переполнены, каждый раз посредством вращения, Q0 позволили бы передать 4470 байтов, и Q1 позволят передать $[4470 + (9-1) * 512] = 8566$ байтов. Это дало бы трафик, который переходит к Q0 приблизительно 1/3 пропускной способности и трафика, который проходит Q1 о 2/3 пропускной способности.

Примечание: Стандартная формула двухсторонней очереди, используемая для вычислений присвоения пропускной способности MDRR, является $D = MTU + (\text{вес } 1) * 512$. В версиях ранее, чем Cisco IOS Software Release 12.0 (21) US / ST, Механизм 4 линейных карты использовали другую формулу двухсторонней очереди. Для настройки веса MDRR для корректного назначения полосы пропускания гарантируйте выполнение Cisco IOS Software Release позже, чем 12.0 (21) S/ST.

Примечание: Квантовая формула для Механизма 4 + линейные карты (для направления toFab, это не допустимо для FrFab), $\text{Quantum} = \text{BaseWeight} + \{\text{BaseWeight} * (\text{QueueWeight} - 1) * 512\} / \text{MTU}$. BaseWeight получен с этой формулой: $\text{BaseWeight} = \{(\text{MTU} + 512 - 1) / 512\} + 5$

Примечание: Все вычисления округлены; т.е. все десятичные числа проигнорированы.

Примечание: Чтобы знать, поддерживает ли определенная линейная карта механизма MDRR, посмотрите [Поддержку MDRR Типом модуля](#).

Очередь по приоритету на MDRR

Серия Cisco 12000 поддерживает очередь приоритетов (PQ) в MDRR. Эта очередь предоставляет низкую задержку и малые колебания задержки, требуемые чувствительным к времени трафиком, таким как Передача голоса по IP (VoIP).

Как было сказано ранее, маршрутизаторы серии Cisco 12000 не поддерживают механизм взвешенной равноправной очередности (WFQ). Таким образом приоритетная очередность в MDRR отличается от очередей с малой задержкой в программном обеспечении Cisco IOS, доступных на других платформах.

Основное различие - то, как планировщик MDRR может быть настроен для обслуживания PQ в одном из двух режимов, как перечислено в [таблице 1](#):

Таблица 1 – как настроить планировщика MDRR для обслуживания PQ в двух режимах

	Альтернативный режим	Режим строгого соблюдения приоритета
Преимущества	Здесь, PQ обслуживается промежуточные другие очереди. Другими словами, планировщик MDRR альтернативно обслуживает PQ и любые другие настроенные очереди.	Здесь PQ обслуживается каждый раз, когда это не пусто. Это предоставляет самую низкую задержку аналогичного трафика.
Недостатки	Этот режим может представить дрожание и задержаться когда по сравнению с режимом строгого соблюдения приоритета.	Если соответствующие потоки являются агрессивными отправителями, этот режим может исчерпать ресурсы другие очереди, особенно.

Альтернативный режим может осуществить меньше контроля на дрожании и задержке. Если планировщик MDRR начинает обслуживать кадры размера MTU от очереди данных, и затем голосовой пакет поступает в PQ, планировщик в альтернативном режиме полностью служит неприоритетной последовательности, пока ее счетчик дефицита не достигает нуля. В это время PQ не обслуживается, и Пакеты VoIP задержаны.

В отличие от этого, в режиме строгого соблюдения приоритета, сервисы планирования только текущий неприоритет пакета и затем переключается на PQ. Планировщик начинает обслуживать неприоритетную последовательность только после того, как PQ станет абсолютно пустым.

Следует отметить, что очередь с приоритетами в альтернативном режиме с приоритетами обслуживается несколько раз в цикле, и таким образом берет больше пропускной

способности, чем другие очереди с тем же номинальным весом. Сколько еще функция того, сколько очередей определено. Например, с тремя очередями, очередь с низкой задержкой обслуживается вдвое более часто, чем другие очереди, и она передает дважды свой вес на цикл. Если заданы восемь очередей, то очередь с низкой задержкой будет обслуживаться в семь раз чаще, а эффективная нагрузка будет в семь раз выше. Таким образом пропускная способность, которую может взять очередь, отнесена к тому, как часто это подается на циклический выбор, который в свою очередь зависит от того, сколько очередей определено в целом. В альтернативном режиме с приоритетами очередь с приоритетами обычно настраивается с маленьким весом по этой определенной причине.

Как пример, предположите, что определены четыре очереди: 0, 1, 2 и очередь с приоритетами. В альтернативном режиме с приоритетами, если бы все очереди переполнены, они были бы обслужены следующим образом: 0, llq, 1, llq, 2, llq, 0, llq, 1.... где llq означает очередь с низкой задержкой.

Каждый раз во время обслуживания очереди она может заполниться до ее сконфигурированного веса. Минимальная пропускная способность очереди с небольшой задержкой:

- WL = вес очереди с низкой задержкой.
- W0, W1... Wn = веса обычных очередей DRR.
- n = количество обычных очередей DRR используется для этого интерфейса.
- BW = Пропускная способность ссылки.

В альтернативном режиме приоритетов минимальная пропускная способность очереди с низкой задержкой = $BW * n * WL / (n * WL + \text{Sum}(W0, Wn))$.

Вес является единственным переменным параметром в MDRR, который может быть настроен. Это влияет на относительное значение пропускной способности, которое класс трафика может использовать, и сколько трафика передается в одном повороте. Использование больших весов означает, что полный цикл занимает больше времени, и возможно увеличивает задержку.

Указания по настройке

- Лучше настроить вес класса, который имеет самые низкие требования пропускной способности к 1 для хранения задержки и дрожания максимально низко среди других классов.
- Выберите минимальное возможное значение веса. Начните с веса 1 для класса с наименьшей пропускной способностью. Например, при использовании двух классов с 50%-й пропускной способностью для каждого класса необходимо настроить 1 и 1. Это не целесообразно использовать 10 и 10, потому что нет никакого влияния на производительность при выборе 1. Кроме того, большой вес порождает большие колебания задержки.
- Низкое значение веса для LLQ очень важно, особенно в альтернативном режиме для не добавления слишком большого количества задержки или дрожания к другим классам.

Пример MDRR

Пример в этом разделе взят из *Архитектуры программного обеспечения Cisco IOS*, Cisco Press.

Предположите, что у нас есть три очереди:

- У очереди 0 – есть квант 1500 байтов; это очередь с низким временем ожидания настроена для работы в альтернативном режиме.
- У очереди 1 – есть квант 3000 байтов.
- Очередь 2 в “Б” содержит кванты, равные 1500 байт.

[Рисунок 1](#) иллюстрирует первоначальное состояние очередей, наряду с некоторыми пакетами, которые были получены и помещены в очередь.

Рисунок 1 – первоначальное состояние MDRR

Очередь 0 обслуживается сначала, ее квант добавлен к ее счетчику дефицита, Пакет 1, который составляет 250 байтов, передан, и его размер вычтен из счетчика дефицита. Поскольку счетчик дефицита очереди 0 еще больше, чем 0 ($1500 - 250 = 1250$), пакет 2 передан также, и его длина, вычтенная из счетчика дефицита. Счетчик дефицита очереди 0 теперь -250, таким образом, очередь 1 обслуживается затем. [Рисунок 2](#) указывает на это состояние.

Рисунок 2 – MDRR последующее Состояние

Счетчик дефицита очереди 1 установлен в 3000 ($0 + 3000 = 3000$), и пакеты 4 и 5 переданы. С каждым переданным пакетом вычтите размер пакета от счетчика дефицита, таким образом, счетчик дефицита очереди 1 уменьшен до 0. [Рисунок 3](#) иллюстрирует это состояние.

Рисунок 3 – Состояние MDRR, когда Счетчик дефицита Очереди 1 является Нулем

Мы должны возвратиться от альтернативного режима с приоритетами до очереди обслуживания 0. Снова, квант добавлен к текущему счетчику дефицита, и счетчик дефицита очереди 0 установлен в результат ($-250 + 1500 = 1250$). Пакет 3 теперь передан, так как счетчик дефицита больше 0, а очередь 0 теперь пуста. [После опустошения очереди для счетчика дефицита устанавливается значение 0, как показано на рисунке 4.](#)

Рисунок 4 – Состояние MDRR, Когда Освобождена Очередь

Очередь 2 обслуживается затем; ее счетчик дефицита выставлен на 1500 ($0 + 1500 = 1500$). Пакеты 7 - 10 переданы, который оставляет счетчик дефицита в 500 ($1500 - (4 * 250) = 500$). Поскольку счетчик дефицита еще больше, чем 0, пакет 11 также передан.

[При передаче пакета 11 очередь 2 опустошается, а ее счетчик дефицита выставляется на 0, как показано на рисунке 5.](#)

Рисунок 5 – Состояние MDRR, Когда Передан Пакет 11

Затем очередь 0 обслуживается снова (потому что мы рассматриваем режим с альтернативными приоритетами). Поскольку это пусто, мы очередь обслуживания 1 следующее, и пакет передачи 6.

[Поддержка MDRR типом модуля](#)

Cisco серии 12000 поддерживает пять моделей плат линии с уникальными архитектурами процессоров пересылки уровня 3 (L3). Поддержка MDRR меняется в зависимости от типа Механизма L3 и типа карты. Например, нет никакой поддержки MDRR и WRED на Механизме 0 карт линии ATM. Команда `show diag` используется для определения типа модуля L3 установленных линейных плат:

```
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 на ToFab (Rx) очереди

Можно использовать или "Синтаксис legacy CoS" или "Интерфейс командной строки Модульного QoS" для настройки MDRR на Серии Cisco 12000. Последующие разделы в этом документе обсуждают, как настроить MDRR с Устаревшим CoS или Модульным QoS. Необходимо настроить очереди ToFab с синтаксисом legacy CoS только, поскольку они не поддерживают более новый синтаксис MQC. [Подробные сведения приведены в таблице 2.](#)

Таблица 2 – детализирует на MDRR на ToFab (Rx) очереди

	Где реализовано	ToFab MDRR	PQ альтернативы ToFab	ToFab Strict PQ	ToFab WRED
Eng0	Программное обеспечение	Нет	Нет	Да	Да
Eng1	-	Нет	Нет	Нет	Нет
Eng2	Аппаратные средства	Да	Да	Да	Да
Eng3	Аппаратные средства	Нет	Да	Да	Да
Eng4	Аппаратные средства	Да	Да	Да	Да
Eng4 +	Аппаратные средства	Да	Да	Да	Да

** MDRR поддерживается на картах каналов Engine 0 в направлении ToFab (Rx), но только в режиме жесткого, а не чередующегося приоритета. Оставшиеся семь очередей поддерживаются в обычном режиме.

Входные интерфейсы поддерживают отдельную виртуальную очередь вывода на целевой LC. Порядок реализации этих очередей зависит от типа модуля L3.

- Механизм 0 – Входящие LC поддерживает восемь очередей на конечный слот. Таким образом максимальное число очередей $16 \times 8 = 128$. Каждая очередь может быть настроена отдельно.
- Engines 2, 3, 4, and 4+ - входящие линейные платы поддерживают восемь очередей на один интерфейс назначения. С 16 конечными слотами и 16 интерфейсами на слот,

максимальное число очередей $16 \times 16 \times 8 = 2048$. Все интерфейсы на конечном слоте должны использовать те же параметры.

MDRR для очередей FrFab (Tx)

MDRR на очередях FrFab последовательно работает, внедренный ли в аппаратных средствах или программном обеспечении. Все типы Механизма L3 поддерживают восемь очередей классов для каждого исходящего интерфейса. Посмотрите [таблицу 3](#) для подробных данных.

Таблица 3 – детализирует на MDRR на FrFab (Tx) очереди

	Где реализовано	Альтернативная PQ в FrFab	FrFab строгий PQ	FrFab WRED
Eng0	Software ¹	Нет	Да	Да
Eng1	-	Нет	Нет	Нет
Eng2	Аппаратные средства	Yes ²	Да	Да
Eng3	Аппаратные средства	Нет	Да	Да
Eng4	Аппаратные средства	Да	Да	Да
Eng4 +	Аппаратные средства	Да	Да	Да

¹Support для MDRR на очередях устройства FrFab 0 LC представлен в этих версиях программного обеспечения Cisco IOS:

- Программное обеспечение Cisco IOS версии 12.0(10)S - 4xOC3 и 1xOC12 POS, 4xOC3, и 1xCHOC12/STM4.
- Программное обеспечение Cisco IOS версии 12.0(15)S - 6xE3 и 12xE3.
- Программное обеспечение Cisco IOS версии 12.0(17)S - 2xCHOC3/STM1.

²You должен настроить альтернативный MDRR в направлении FrFab с синтаксисом legacy CoS.

Примечание: 3xGE LC поддерживает MDRR на очередях ToFab и, как от программного обеспечения Cisco IOS версии 12.0(15)S, на очередях FrFab с двумя ограничениями, а именно, неподвижным квантом и одиночной очередью CoS для каждого интерфейса. Очередь с приоритетами поддерживает квант, который может быть настроен, и и строгие и альтернативные режимы с приоритетами. Все три интерфейса используют общую PQ.

Примечание: См. Комментарии к выпуску маршрутизаторов Cisco серии 12000 для последней информации о поддерживаемых Характеристиках QoS на LC Серии Cisco 12000.

Обзор WRED

Взвешенное случайно раннее обнаружение (WRED) предназначено для предотвращения пагубного влияния перегрузки интерфейса на пропускную способность сети.

Рисунок 6 – вероятность отбрасывания пакетов WRED

Посмотрите [Взвешенное произвольное раннее обнаружение на маршрутизаторе Cisco серии 12000](#) для пояснения параметров WRED. Минимум, максимум и параметры вероятности метки описывают фактическую кривую Произвольного раннего обнаружения (RED). Когда взвешенное среднее число очереди ниже минимального порога, никакие пакеты не отброшены. Когда взвешенное среднее число очереди выше порога максимального размера очереди, все пакеты отброшены, пока среднее число не опускается ниже максимального порога. Когда среднее число между минимумом и максимальными порогами, вероятность, что пакет будет отброшен, может быть вычислена прямой линией из минимального порога (вероятность отбрасывания будет 0) в максимальный порог (вероятность отбрасывания равна 1/метке знаменателю вероятности).

Различие между КРАСНЫМ и WRED - то, что WRED может выборочно сбросить от трафика с более низким приоритетом, когда интерфейс начинает переполняться и может предоставить дифференцируемые характеристики производительности для других Классов обслуживания (CoS). По умолчанию WRED использует другой профиль КРАСНОГО для каждого веса (приоритет IP-трафика - 8 профилей). Менее важные пакеты отбрасываются более агрессивно, чем более важные.

Это - сложная проблема настроить параметры WRED для управления глубиной очереди и зависит от многих факторов, которые включают:

- Предлагаемый трафик и профиль.
- Соотношение загрузки к полезной мощности.
- Поведение трафика в присутствии перегрузки.

Эти факторы варьируются сеть сетью и, в свою очередь, зависят от предложенных услуг и от клиентов, которые используют те сервисы. Таким образом мы не можем сделать рекомендации, которые применяются к определенным пользовательским окружениям. Однако [таблица 4](#) обычно описывает рекомендуемые значения на основе пропускной способности ссылки. В этом случае мы не различаем характеристики отбрасывания между другими классами службы.

Таблица 4 – рекомендуемые значения на основе пропускной способности ссылки

Bandwidth	Теоретический BW (кбит/с)	Физическая BW ¹ (кбит/с)	Минимальный порог	Максимальный порог
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

¹Physical Скорость SONET

Существует несколько ограничений, которые учитываются при расчете вышеуказанных пороговых значений. Например, максимизация использования соединения при уменьшении

средней глубины очереди, различия между Максимумом и Минимумом должна быть питанием два (из-за аппаратного ограничения). На основании опыта и моделирования максимальная мгновенная длина очереди, контролируемой RED, меньше чем 2 MaxTh. Для OC48 и выше, 1 MaxTh и так далее. Однако точное определение этих значений не входит в спектр вопросов, обсуждаемых в документе.

Примечание: Значение Экспоненциальной весовой константы не должно быть настроено на Engine 2 и выше линейных карт, так как выборка очереди аппаратных ресурсов используется вместо этого. Для Механизма 0 LC могут быть настроены эти значения:

- ds3 – 9
- oc3 – 10
- oc12 – 12

Примечание: WRED не поддерживается на Engine 1 LC.

Как следующие разделы объясняют, можно использовать и синтаксис legacy CoS и более новый синтаксис MQC для настройки WRED.

[Используйте синтаксис legacy CoS для конфигурации](#)

Унаследованный синтаксис класса обслуживания (CoS) Cisco 12000 предусматривает определение CoS с использованием шаблона CoS-очередь-группа. Вы тогда применяете шаблон к очередям ToFab и FrFab на входящем или исходящих интерфейсах, соответственно.

[Шаг 1: Определите cos-queue-group](#)

Команда **cos-queue-group** создает именованный шаблон MDRR и параметров WRED. Вот доступные параметры конфигурации в CLI:

```
Router(config)#cos-queue-group oc12 Router(config-cos-que)#? Static cos queue commands: default
Set a command to its defaults dscp Set per DSCP parameters, Engine 3 only exit Exit from COS
queue group configuration mode exponential-weighting-constant Set group's RED exponential weight
constant. (Not used by engine 0, 1 or 2 line cards) no Negate a command or set its defaults
precedence Set per precedence parameters queue Set individual queue parmeters random-detect-
label Set RED drop criteria traffic-shape Enable Traffic Shaping on a COS queue group
```

С MDRR можно сопоставить приоритет IP-трафика с очередями MDRR и настроить специальную очередь с низкой задержкой. Можно использовать параметр приоритета при команде **cos-queue-group** для этого:

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

Можно сопоставить определенный приоритет IP со стандартной очередью MDRR (очередь от 0 до 6) или с очередью приоритетов. Приведенная выше команда позволяет сопоставить несколько приоритетов IP той же самой очереди.

Примечание: Рекомендуется использовать приоритеты 5 для очереди с низкой задержкой. Приоритет 6 используется для маршрутизации обновлений.

Можно дать каждой очереди MDRR относительный вес с одной из очередей в группе, определенной как очередь с приоритетами. Можно использовать команду **очереди** под **cos-queue-group**, чтобы сделать это:

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

Используйте команду `cos-queue-group` для определения параметров WRED:

```
random-detect-label <label> <minimum-threshold> <maximum-threshold> <mark-probability denominator>
```

Вот пример `cos-queue-group`, названного `oc12`. Это определяет три класса трафика: очередь 0, 1, и низкая задержка. Это сопоставляет значения приоритета IP-трафика 0 - 3 для организации очереди 0, значения приоритета 4, 6, и 7 для организации очереди 1, и приоритеты 5 к очереди с низкой задержкой. Очередь 1 предназначена для большей пропускной способности.

Пример конфигурации

```
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.
```

Шаг 2 – создание slot-table-cos для очередей ToFab

Для предотвращения головки блокирования линии входные интерфейсы на Серии Cisco 12000 поддерживают виртуальную очередь вывода на конечный слот. Перейдите к линейной карте с помощью команды присоединения и выполните команду `show controller tofab queue execute-on` (или непосредственно введите `execute-on slot 0` команд очереди ToFab `show controllers`) просмотреть эти виртуальные очереди вывода (VOQ). Пример выходных данных, перехваченный непосредственно от консоли LC, предоставлен ниже. Посмотрите, [Как Считать Выходные данные Команд show controller frfab | tofab queue на Интернет-маршрутизаторе Cisco 12000 серии](#).

```
LC-Slot1#show controllers tofab queues Carve information for ToFab buffers SDRAM size: 33554432
bytes, address: 30000000, carve base: 30029100 33386240 bytes carve size, 4 SDRAM bank(s), 8192
bytes SDRAM pagesize, 2 carve(s) max buffer data size 9248 bytes, min buffer data size 80 bytes
40606/40606 buffers specified/carved 33249088/33249088 bytes sum buffer sizes specified/carved
Qnum Head Tail #Qelem LenThresh ---- ---- ---- ----- 5 non-IPC free queues:
20254/20254 (buffers specified/carved), 49.87%, 80 byte data size 1 17297 17296 20254 65535
12152/12152 (buffers specified/carved), 29.92%, 608 byte data size 2 20548 20547 12152 65535
6076/6076 (buffers specified/carved), 14.96%, 1568 byte data size 3 32507 38582 6076 65535
```

```

1215/1215 (buffers specified/carved), 2.99%, 4544 byte data size 4 38583 39797 1215 65535
809/809 (buffers specified/carved), 1.99%, 9248 byte data size 5 39798 40606 809 65535 IPC
Queue: 100/100 (buffers specified/carved), 0.24%, 4112 byte data size 30 72 71 100 65535 Raw
Queue: 31 0 17302 0 65535 ToFab Queues: Dest Slot 0 0 0 0 65535 1 0 0 0 65535 2 0 0 0 65535 3 0
0 0 65535 4 0 0 0 65535 5 0 17282 0 65535 6 0 0 0 65535 7 0 75 0 65535 8 0 0 0 65535 9 0 0 0
65535 10 0 0 0 65535 11 0 0 0 65535 12 0 0 0 65535 13 0 0 0 65535 14 0 0 0 65535 15 0 0 0 65535
Multicast 0 0 0 65535 LC-Slot1#

```

Используйте команду **slot-table-cos** для сопоставления именованного **cos-queue-group** с целевой виртуальной очередью вывода. Можно настроить уникальный шаблон **cos-queue-group** для каждой очереди вывода

```

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

```

Примечание: Вышеупомянутая конфигурация использует три шаблона, названные oc48, oc12 и oc3. Конфигурация для названного oc12 **cos-queue-group** находится как показано в Step1. Точно так же настройте oc3 и oc48. Рекомендуется применить уникальный шаблон к ряду интерфейсов на основе пропускной способности и приложения.

[Шаг 3 - Применяет slot-table-cos к входному интерфейсу](#)

Используйте команду **rx-cos-slot** для применения **slot-table-cos** к LC.

```

Router(config)#rx-cos-slot 0 ? WORD Name of slot-table-cos Router(config)#rx-cos-slot 0 table1
Router(config)#rx-cos-slot 2 table1

```

[Шаг 4 - Применяет cos-queue-group к исходящему интерфейсу](#)

Серия Cisco 12000 поддерживает отдельную очередь на исходящий интерфейс. Для просмотра этих очередей подключите к CLI линейной карты. Используйте команду **присоединения**, и затем выполните команду **show controller frfab queue**, как проиллюстрировано здесь:

```

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

```

Используйте команду **tx-cos** для применения шаблона **cos-queue-group** к очереди интерфейса. Как показано здесь, вы применяете набор параметров непосредственно к интерфейсу; никакие таблицы не необходимы. В данном примере *pos48* является названием набора параметров.

```

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

```

Используйте команду **show cos** для подтверждения конфигурации:

```
Router#show cos !--- Only some of the fields are visible if MDRR is configured on Inbound !---
or Outbound interfaces. Interface Queue cos Group Gi4/0 eng2-frfab !--- TX-cos has been applied.
Rx Slot Slot Table 4 table1 !--- rx-cos-slot has been applied. Slot Table Name - table1 1 eng0-
tofab 3 eng0-tofab !--- slot-table-cos has been defined. cos Queue Group - eng2-tofab !--- cos-
queue-group has been defined. Prec Red Label [min, max, prob] Drr Queue [deficit] 0 0 [6000,
15000, 1/1] 0 [10] 1 1 [10000, 20000, 1/1] 1 [40] 2 1 [10000, 20000, 1/1] 1 [40] 3 1 [10000,
20000, 1/1] 0 [10] 4 2 [15000, 25000, 1/1] 2 [80] 5 2 [15000, 25000, 1/1] 2 [80] 6 no drop low
latency 7 no drop low latency
```

Примечание: Устаревший CLI также использует синтаксис прецедента для трафика Многопротокольной коммутации по меткам (MPLS). Маршрутизатор рассматривает биты MPLS, как будто они - IP-тип сервисов (ToS) биты, и помещает соответствующие пакеты в корректные очереди. Это совсем неверно для MQC. MPLS QoS выходит за рамки этого документа.

[Используйте командную строку Modular QoS CLI \(MQC\) для конфигурации](#)

Цель Командной строки Modular QoS CLI (MQC) Cisco состоит в том, чтобы подключить все другие Характеристики QoS в логичном способе для упрощения конфигурации функций Качества обслуживания (QoS) программного обеспечения Cisco IOS. Например, классификация сделана отдельно от организации очереди, применения политик и формирования. Это служит одиночной основой конфигурации для QoS, которое основано на шаблоне. Вот некоторые точки для запоминания о конфигурации MQC:

- Этому можно легко примениться к и удалить из интерфейса.
- Это может быть легко снова использовано (та же политика может быть применена к нескольким интерфейсам).
- Это предлагает одиночную платформу конфигурации для QoS, которое позволяет легко условию, монитору и устранению неполадок.
- Это предоставляет более высокий уровень абстракции.
- Это независимо от платформы.

На Серии Cisco 12000 команды MQC могут использоваться вместо устаревшего синтаксиса Класса обслуживания (CoS).

Поддержка MQC на Серии Cisco 12000 не подразумевает, что та же Характеристика QoS, установленная доступный на другой платформе, такой как Cisco серии 7500, теперь доступна на Cisco 12000. MQC предоставляет обычный синтаксис, в котором команда приводит к совместно используемой функции или поведению. Например, команда **bandwidth** внедряет минимальную пропускную способность. В то время как Cisco серии 7500 использует WFQ, Серия Cisco 12000 использует MDRR в качестве механизма планирования для создания резервирования полосы пропускания. Основной алгоритм дополняет конкретную платформу.

Значительно, только очереди FrFab поддерживают конфигурацию Характеристик QoS через MQC. Поскольку очереди ToFab являются виртуальными очередями вывода (VOQ), и не истинными входными очередями, они не поддерживаются MQC. Они должны быть настроены с устаревшими командами CoS.

[Таблица 5](#) перечисляет поддержку MQC на тип Механизма L3.

Таблица 5 – поддерживает для MQC для типов механизма L3

Тип модуля L3	Ядро 0	Engine 1	Engine 2	Процессор 3	Модуль 4	Механизм 4 +
Поддержка MQC	Да	Нет	Да	Да	Да	Да
Версия IOS	12.0 (15) S	-	12.0 (15) S ¹	12.0 (21) S	12.0 (22) S	12.0 (22) S

¹Remember эти исключения с MQC поддерживают на Механизме 0 и 2 линейных платы (LC) s:

- 2xSNC3/STM1 – введено в версии 12.0(17)S.
- 1xOC48 DPT – введено в 12.0(18)S.
- 8xOC3 ATM – Запланирован для 12.0(22)S.

MQC использует эти три шага для создания политики QoS:

1. Определение одного или нескольких классов трафика с помощью команды **class-map**.
2. Создайте политику QoS с помощью команды **policy-map** и назначьте для действий QoS, таких как пропускная способность или приоритет, именованный класс трафика.
3. Используйте команду **service-policy** для присоединения **policy-map** очереди FrFab исходящего интерфейса.

Используйте команду **show policy-map interface** для мониторинга политики.

См. [Обзор Модульного интерфейса командной строки для обеспечения качества обслуживания](#) для получения дополнительной информации.

Шаг 1 - Определяет карты классов

Команда **class-map** используется для определения классов трафика. Внутренне, на Серии Cisco 12000, команда **class-map** назначает класс на определенную очередь CoS на линейной карте (см. [Шаг 4](#) для подробных данных).

Команда **class-map** поддерживает "match-any", который размещает пакеты, которые совпадают с любым из сообщений о совпадении в класс и "match-all", который размещает пакеты в этот класс только, когда все операторы истинны. Эти команды создают класс по имени "Prcs_5" и классифицируют все пакеты с приоритетом IP-трафика 5 к этому классу:

```
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
```

[Таблица 6](#) перечисляет поддерживаемые условия соответствия для каждого типа Механизма L3.

Таблица 6 – поддерживаемые условия соответствия для механизмов L3

	Механизм 0, 2	Процессор 3	Модуль 4	Механизм 4 +
приоритет IP	Да	Да	Да	Да ¹

access-group	Нет	Да	Нет	Нет
exp mpls	Нет	Да	Нет	Да (12.0.26S)
ip dscp	Нет	Да	Нет	Да (12.0.26S)
группа qos	Нет	Да	Нет	Нет
match input-interface POS x/y	Нет	Да (только как политика получения)	Нет	Нет

¹ вход/выход с тех пор 12.0.26S

Шаг 2 - Создает policy-map

Команда policy-map используется для присвоения политики обработки пакетов или действий к одному или более определенным классам. Например, когда вы назначаете резервирование полосы пропускания или применяете случайный профиль отбрасывания.

Серия Cisco 12000 поддерживает подмножество функций MQC, на основе высокоскоростной архитектуры Механизмов L3. [Таблица 7](#) перечисляет команды, которые поддерживаются:

Таблица 7 – поддерживаемые команды

Команда	Описание
bandwidth	Предоставляет минимальную пропускную способность во время периодов перегрузки. Это задано как процент от скорости связи или как абсолютное значение. Если класс не использует или нуждается в пропускной способности, равной зарезервированному кбит/с, доступная пропускная способность может использоваться другими классами полосы пропускания.
police, shape	Ограничивает объем трафика, который может передать класс. Эти команды немного отличаются в функции. Команда политики определяет трафик, который превышает настраиваемую пропускную способность, и отбрасывает или отмечает его. Буферы команд формы любой дополнительный трафик и списки это для передачи в постоянной скорости передачи, но не понижается или замечает.

Queue-limit	Назначает фиксированную длину очереди данному классу трафика. Можно задать это в количестве пакетов, которые могут быть проведены в очереди.
приоритет	<p>Определяет очередь как очередь с низкой задержкой. Поддержка строго режима MQC только для PQ. Альтернативный режим не поддерживается через MQC. Используйте приоритетную команду без значения процента для включения режима строгого соблюдения приоритета.</p> <p>Примечание: Реализация приоритетной команды на Серии Cisco 12000 отличается от реализации на других маршрутизаторах, которые выполняют программное обеспечение Cisco IOS. На этой платформе приоритетный трафик не ограничен настроенным значением кбит/с во время периодов перегрузки. Таким образом необходимо также настроить команду политики для ограничения, сколько пропускной способности класс приоритета может использовать и гарантировать достаточной пропускной способности для других классов. В это время команда политики только поддерживается на линейных картах Механизма 3. На других линейных картах механизма только class-default позволен при настройке класса приоритета.</p>
random-detect	Назначает профиль WRED. Используйте команду random-detect precedence для настройки значений WRED на по умолчанию на значение приоритета IP-трафика.

На LC Механизма 3 необходимо настроить очереди FrFab с Командной строкой Modular QoS CLI (MQC); устаревший Интерфейс командной строки (CLI) не поддерживается.

При настройке **команды bandwidth** обратите внимание, что Механизм 0 и 2 LC поддерживает шесть классов полосы пропускания только. Седьмой класс может использоваться для сервиса низкой задержки и восьмого класса, который является class-default, используется для всего неаналогичного трафика. Поэтому всего получается восемь очередей. Класс по умолчанию не используется как класс приоритетов.

На LC Механизма 3 команда **процента полосы пропускания** преобразована в значение кбит/с, которое меняется в зависимости от основной скорости соединения, и затем настроенный непосредственно на очереди. Точность этой минимальной гарантированной пропускной способности составляет 64 кбит/с.

Несмотря на то, что никакое преобразование в количественное значение не сделано с **командой bandwidth**, у всех очередей есть квант. На LC Механизма 3 количественное

значение установлено внутренне на основе максимального размера передаваемого блока данных (MTU) интерфейса и установлено одинаково для всех очередей. Механизм интерфейса командной строки MQC для изменения этой квантовой величины непосредственно или косвенно отсутствует. Количественное значение должно быть больше, чем или равным максимальному размеру блока данных (MTU) интерфейса. Внутренне, количественное значение находится в модулях 512 байтов. Таким образом, с MTU 4470 байтов, минимальное количественное значение MTU должно быть 9.

[MDRR на LC механизма 3](#)

Этот раздел предоставляет примечания к конфигурации для реализации WRED и MDRR на LC Механизма 3.

- Пропускная способность MDRR, настроенная в CLI, преобразована в сумму, соответствующую L2 (например, издержки L1 удалены). Это количество затем округляется до следующих 64 кбит/сек и заносится в оборудование в виде программы.
- Три различных профиля WRED поддерживаются для одного класса.
- WRED (максимальный порог - минимальный порог) приближен к самому близкому питанию 2. В то время как максимальный порог сохранен неизменным, минимальный порог тогда отрегулирован автоматически.
- Поддерживается метка значения вероятности 1.
- Конфигурация экспоненциальной весовой константы не поддерживается.
- Приоритет IP-трафика, биты EXP MPLS и DSCP-значения поддерживаются.

Примечание: Каждый порт или канал на Tetra (4GE-SFP-LC =) или CHOC12/DS1-IR-SC = Линейные карты frostbite имеют четыре очереди, выделенные по умолчанию. Эти четыре очереди состоят из придерживающегося:

- Одна очередь с приоритетами (LLQ) класс
- Один класс очереди по умолчанию
- Два обычных некласс приоритета

При применении стратегии обслуживания, содержащей больше, чем эти четыре класса (1 HPQ, 2 LPQs и class-default) к интерфейсу, сообщат об ошибке слежения:

Маршрутизатор (config-if) #service-policy выходные данные mdr-policy

% Недостаточно ресурсов организации очереди, доступных для удовлетворения запроса.

С 12.0 (26) S, команда была добавлена для 4GE-SFP-LC = линейная плата Tetra, которая позволяет конфигурацию восьми очередей/VLAN вместо четыре. Эти восемь очередей состоят из придерживающегося:

- Один LLQ
- Одна очередь class-default
- Шесть обычных очередей

Использование этой команды потребует повторной загрузки микрокода линейной платы и приведет к способности настроить только 508 VLAN вместо 1022. Синтаксис команды приведен ниже:

[никакой] hw-module slot <slot#> очереди интерфейса qos 8

Пример:

Маршрутизатор (config) #hw-module очереди интерфейса qos слота 2 8

% Warning: Микроперезагрузка линейная плата для этой команды для вступления в силу

Маршрутизатор (config) #microcode перезагружается 2

Эта команда будет доступна для CНОС12/DS1-IR-SC = Линейная карта frostbite в 12.0 (32) S

Пример #1 - Команда процента полосы пропускания

В данном примере 20 процентов полосы пропускания выделяется для класса трафика Prec_4 и 30 процентов для класса трафика Prec_3. Это оставляет оставшиеся 50 процентов классу класс по умолчанию.

Кроме того, он настраивает WRED в качестве механизма сброса для всех классов данных.

Пример #1 - процент полосы пропускания

```
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
```

Пример #2 - пропускная способность {кбит/с} Команда

Данный пример иллюстрирует, как применить команду bandwidth как значение абсолютных значений kbr вместо процента.

Пример #2 - пропускная способность {кбит/с}

```
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
```

Пример №3 – команда priority

Данный пример разработан для поставщиков услуг, которые используют маршрутизатор Cisco серии 12000 в качестве маршрутизатора границы провайдера (PE) MPLS и должны настроить политику обслуживания QoS на ссылке между Периферийным маршрутизатором и маршрутизатором порта заказчика Customer Edge (CE). Это размещает приоритет IP-трафика 5 пакетов в очередь с приоритетами и ограничивает выходные данные той очереди к 64 Мбит/с. Это тогда назначает часть остатка полосы пропускания к классам полосы пропускания.

Все очереди неклассы приоритета настроены с командой **random-detect** для включения WRED как политики отбрасывания. Всему классу полосы пропускания и class-default нужно было настроить WRED явно.

Пример #3 - приоритет

```
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
```

[Шаг 3 - Назначает policy-map на очередь исходящего интерфейса](#)

Как упомянуто выше, MQC работает только с очередями FrFab на исходящем интерфейсе. Для применения определенного policy-map используйте команду **service-policy output**, как показано здесь:

```
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
```

[Шаг 4 - Монитор и проверяет политику обслуживания](#)

Используйте команду **show policy-map interface** для просмотра приложения политики. Команда **show policy-map interface** отображает придерживающееся:

- Настраиваемая пропускная способность и классы приоритета и критерии соответствия.
- Любые профили WRED.
- Форма и параметры политики.
- Учет трафика и скорости.
- Внутренняя очередь CoS, с которой сопоставлен отдельный класс. На эти очереди ссылается тот же индекс, который используется в выходных данных команды **show controller frfab queue**.

Вот пример завершенной конфигурации и команд показа для мониторинга политики:

```

Законченная конфигурация
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
policyle 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 policyle !--- Step 3-
Attach policy-map to the interface.

```

Используйте команду **show policy-map interface** для просмотра политики, настроенной на интерфейсе, наряду со всеми настроенными классами. Вот выходные данные команды:

```

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

```

Команды для мониторинга управления перегрузками сети и предотвращения

Этот раздел перечисляет команды, которые можно использовать для мониторинга управления перегрузками сети и политики избежания.

[Таблица 8](#) перечисляет подходящие команды для Входных линейных плат и Выходных линейных плат.

Таблица 8 – команды для линейных карт

Входная линейная плата	Выходная линейная карта
<ul style="list-style-type: none"> • show interfaces • очередь tofab sh controller слота ехес <x> • ехес slot <x> show controller tofab queue <слот> <порт> • ехес slot <x> show controller tofab qm stat 	<ul style="list-style-type: none"> • show interfaces • show interfaces <y> случайный • очередь frfab покажите контроллер слота ехес <y> • ехес slot <y> show controller frfab queue <порт> • ехес slot <y> show controller frfab QM stat

Эти команды объяснены в этом разделе.

Команда "show interfaces"

Перед использованием этой команды подтвердите корректную "Стратегию организации очереди". Если выходные данные отображают First In, First Out (FIFO), гарантируйте, что команда **service-policy** появляется в рабочей конфигурации (если MQC использовался для настройки MDRR).

Определите число отбрасываний выхода, которое представляет собой общее число отбрасываний WRED FrFab, произошедших для исходящего трафика на данном интерфейсе. Количество отбрасываний выходных данных в выходных данных команды **show interfaces** должно быть равно или выше, чем количество отбрасываний выходных данных в выходных данных команды **show interfaces <number> random**.

Примечание: На маршрутизаторе Cisco серии 12000 выходные сбросы интерфейса обновляются после обновления сбросов WRED. Существует маленький шанс, что при использовании программного средства для запроса обоих счетчиков сбросов, еще не обновлены интерфейсные отбрасывания.

```
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
```

Show interfaces {номер} случайная Команда

При использовании этой команды вы должны:

- Проверьте, что корректный шаблон **cos-queue-group** применен к этому интерфейсу.
- Проверьте веса MDRR. Для каждой очереди MDRR можно проверить взвешенное среднее для длины очереди и самого высокого достигнутого значения (в пакетах). Значения вычислены как взвешенное среднее и не должны отражать фактический Maximum Queue Depth, никогда достигнутый.
- Проверьте минимальные и максимальные значения порогов WRED.
- Проверьте количество случайных и пороговых отбрасываний для каждого маркера RED (отбрасывания To Fabric указывают общее количество отбрасываний для этого маркера на всех линейных картах).
- "Счетчик" отбрасываний TX-queue-limit используется только на Механизме 1 LC, которые не поддерживают WRED. Механизм 1 карта позволяет вам установить предел очередей MDRR с командой **TX-queue-limit interface**. Там, где есть поддержка WRED, пороговые значения WRED определяют глубину очередей MDRR.

```
Router#show interfaces POS 4/0 random POS4/0 cos-queue-group: oc12 RED Drop Counts TX Link To
Fabric RED Label Random Threshold Random Threshold 0 29065142 73492 9614385 0 1 0 0 0 0 2 0 0 0
0 3 0 0 0 0 4 0 0 0 0 5 0 0 0 0 6 0 0 0 0 TX-queue-limit drops: 0 Queue Lengths TX Queue (DRR
configured) oc12 Queue Average High Water Mark Weight 0 0.000 2278.843 1 1 0.000 0.000 73 2
0.000 0.000 10 3 0.000 0.000 10 4 0.000 0.000 10 5 0.000 0.000 10 6 0.000 0.000 10 Low latency
0.000 0.000 10 TX RED config Precedence 0: 375 min threshold, 2423 max threshold, 1/1 mark
weight Precedence 1: not configured for drop Precedence 2: not configured for drop Precedence 3:
```


В этом разделе описываются команды, используемые для мониторинга входящего управления перегрузками сети.

Команда "show interfaces"

Перед выдачей этой команды проверьте, является ли значение в числе пропущенных ошибок на увеличении. Вы будете видеть пропущенные пакеты, если у вас закончится память на стороне ToFab или если линейная карта не принимает пакеты достаточно быстро. [Дополнительные сведения см. в документе "Устранение отбрасываний входа на Интернет-маршрутизаторе Cisco серии 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
```

Команда show controller tofab queue слота exec (x)

Когда не было никакой перегрузки на выходной линейной плате в слоте 3, этот пример выходных данных команды `exec slot <x> show controller tofab queue` был перехвачен.

```
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
```

Когда была перегрузка на слоте 3, следующий результат был перехвачен:

```
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
```

Очередь tofab покажите контроллер слота exec (x) (слот) (порт) Команда

Используйте эту команду для определения, сколько памяти используется на стороне ToFab. В частности обратите внимание на номер в "#Qelem" столбец. Заметьте что:

- Когда никакая память не используется, значения в их самом высоком.
- Значение "#Qelem" уменьшений столбца как пакеты буферизовано.
- Когда столбец "#Qelem" станет равным 0, будут использоваться все выделенные буферы. На LC модуля 2 небольшие пакеты могут заимствовать пространство буфера у более крупных пакетов.

Можно также использовать эту команду для определения количества пакетов в очереди на виртуальной очереди вывода. Пример здесь показывает, как проверить слот 14 для мгновенного числа пакетов на этих очередях для слота 4, порта 1 (POS 4/1). Мы видим 830 пакетов, помещенных в очередь на очереди MDRR 1.


```
Router# execute-on slot 14 show controllers tofab queue 4 1 ===== Line Card (Slot 14)
===== ToFab Queue Slot 4 Int 1 DRR# Head Tail Length Average Weight Deficit 0 0 0 0 0.000 4608
0 1 203005 234676 830 781.093 41472 37248 2 0 0 0 0.000 9216 0 3 0 0 0 0.000 9216 0 4 0 0 0
0.000 9216 0 5 0 0 0 0.000 9216 0 6 0 0 0 0.000 9216 0 7 0 0 0 0.000 9216 0
```

Слот ехес (х) показывает Команду статистики QM tofab контроллера

Воспользуйтесь этой командой, чтобы определить количество отбрасываний ToFab для линейной платы. Также не проверьте для "никакого счетчика" отбрасывания памяти, который инкрементно увеличивается. Значение данного счетчика увеличивается, если на стороне ToFab не настроены функции CoS.

```
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
Q status errors 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Примеры практического применения

Этот пример практического применения показывает, как настроить обычную политику для ядра сети среды поставщика услуг. Это применяет команды очереди и позволяет вам использовать MDRR/WRED для активного управления очередью. Политики QoS в краевых маршрутизаторах обычно используют маркировку трафика, создание условий, и так далее, чтобы позволить маршрутизаторам в ядре сортировать трафик в классы на основе приоритета IP-трафика или DiffServ Code Point (DSCP) значения. Этот пример практического применения использует Характеристики QoS программного обеспечения Cisco IOS для совещания трудных Соглашений об уровне обслуживания (SLA) и другие уровни сервиса для голоса, видео и сервисов передачи данных на той же магистрали IP.

В подходе поставщик услуг внедрил три класса трафика. Наиболее важным является класс LLQ, или Low Latency Queueing (очередь с низкой латентностью). Это класс для речи и видео. Этот класс должен испытать минимальную задержку и дрожать и никогда не должен испытывать потерю пакета или переупорядоченные пакеты, пока пропускная способность этого класса не превышает пропускную способность соединения. В архитектуре DiffServ данный класс называется трафиком срочной пересылки для каждого узла (EF PHB). Интернет-провайдер (ISP) разработал сеть в способе, которым этот класс не превышает 30% на нормальной нагрузке пропускной способности соединения. Другие два класса: бизнес-класс и класс наилучшего сервиса.

В дизайне мы настроили маршрутизаторы таким способом, которым бизнес-класс всегда получает 90% остатка полосы пропускания, и класс максимальной эффективности получает 10%. Эти два класса имеют меньше срочного трафика и могут испытать потерю трафика, более высокую задержку и дрожание. В дизайне фокус находится на линейных картах Engine 2: 1xOC48 об В, 4xOC12 об В, и 8xOC3 линейные карты.

Линейные карты версии В подходят лучше всего для переноса Трафика VoIP из-за пересмотренного ASIC и архитектуры аппаратного обеспечения, которая представляет очень мало задержки. С пересмотренным ASIC очередь FIFO передачи изменена драйвером линейной карты примерно к два раза самому большому MTU на карте. Ищите "-В", добавленный к номеру изделия, такой как OC48E/POS-SR-SC-B =.

Примечание: Не следует путать очередь передачи FIFO с очередями FrFab, которые могут быть настроены на модуле Engine 0 линейных плат с помощью команды tx-queue-limit interface.

[Таблица 9](#) перечисляет критерии соответствия для каждого класса.

Таблица 9 – критерии соответствия для каждого класса

Имя класса	Критерии сопоставления
Очередь по приоритету – голосовой трафик	Приоритет 5
Бизнес-очередь	Приоритет 4
Очередь оптимального уровня	Приоритеты 0

Линейные платы OC48 могут ставить в очереди ToFab большое количество пакетов. Таким образом, важно настроить MDRR/WRED в очередях ToFab, особенно когда выходной интерфейс является высокоскоростным, таким как OC48. Матрица может только коммутировать трафик к получающей линейной плате при теоретической максимальной скорости 3 Гбит/с (пакеты по 1500 байт). Если передаваемый общий объем трафика будет больше, чем, что коммутационная матрица может нести в ее карту получения, то много пакетов будут помещены в очередь на очередях ToFab.

Interface POS3/0

```

description OC48 egress interface ip address 10.10.105.53 255.255.255.252 no ip directed-
broadcast ip router Isis encapsulation ppp mpls traffic-eng tunnels tag-switching ip no peer
neighbor-route crc 32 clock source internal POS framing sdh POS scramble-atm POS threshold sf-
ber 4 POS flag sls0 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 sls0 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 sls0 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

```

Ожидается что, чем больше Трафика VoIP вы имеете, тем больший деловой трафик должен ждать, прежде чем это будет подано. Однако это не проблема, потому что трудный SLA не требует никакого отбрасывания пакета, и очень низкая задержка и дрожание для очереди с приоритетами.

Дополнительные сведения

- [Чтение выходных данных команд "show controller frfab" и "tofab queue" на Internet-маршрутизаторе Cisco серии 12000](#)
- [Устранение неполадок, связанных с игнорируемыми пакетами и нехваткой памяти на IP-маршрутизаторе серии Cisco 12000](#)
- [Устранение неисправностей при просмотре входной информации на Интернет-маршрутизаторе Cisco серии 12000](#)
- [Механизм взвешенного случайного раннего обнаружения на Cisco 12000 Series Router](#)
- [Обзор модульного интерфейса командной строки для обеспечения качества обслуживания](#)
- [Страница технической поддержки IP - маршрутизаторов "серии 12000"](#)
- [Техническая поддержка - Cisco Systems](#)