

Диспетчеризация выходных данных QoS для коммутаторов Catalyst серии 6500/6000 с программным обеспечением CatOS

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Потеря данных выходной очереди](#)

[Типы организации очереди, которые вовлечены в планирование вывода на Catalyst 6500/6000](#)

[Отбрасывание остатка](#)

[Случайное раннее обнаружение и взвешенное случайное раннее обнаружение](#)

[Весовая круговая система](#)

[Очередь с жестким приоритетом](#)

[Возможность организации очереди вывода других линейных карт на Catalyst 6000 возможности команды show port](#)

[Поймите возможность организации очереди порта](#)

[Создайте QoS на Catalyst 6500/6000](#)

[Механизм планирования вывода на Catalyst 6500/6000](#)

[Настройка, мониторинг планирование выходных данных на Catalyst 6500/6000](#)

[Конфигурация по умолчанию для QoS на Catalyst 6500/6000](#)

[!--- конфигурацию](#)

[Планирование вывода монитора и проверяет конфигурацию](#)

[Использование диспетчеризации исходящих потоков для уменьшения задержек и фазового дрожания](#)

[Уменьшение задержек](#)

[Уменьшение фазового дрожания](#)

[Дополнительные сведения](#)

Введение

Планирование вывода гарантирует, что важный трафик не отброшен в случае значительного превышения объема подписки. Этот документ обсуждает все способы и алгоритмы, которые вовлечены в планирование вывода на Cisco Catalyst 6500/6000 коммутаторы серии, которые выполняют системное программное обеспечение операционной системы Catalyst (CatOS). Этот документ также предоставляет краткий обзор

возможности организации очереди коммутаторов Catalyst 6500/6000 и как настроить другие параметры планирования вывода.

Примечание: При выполнении программного обеспечения Cisco IOS на Catalyst 6500/6000 обратитесь к [Планированию вывода QoS на коммутаторах Catalyst 6500/6000 Series Рабочее системное ПО Cisco IOS](#) для получения дополнительной информации.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Примеры в этом документе были созданы от Catalyst 6000 с Supervisor Engine 1A и Policy Feature Card (PFC). Но примеры также допустимы для Supervisor Engine 2 с PFC2 или для модуля управления Supervisor Engine 720 с PFC3.

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 Условные обозначения технических терминов Cisco.](#)

Общие сведения

Потеря данных выходной очереди

Отбрасывания данных в исходящем потоке вызываются перегрузками интерфейсного устройства. Типичная причина этого могла бы быть трафиком от соединения с высокой пропускной способностью, которое коммутируется к более низкой пропускной способности соединения или трафику от множественных входящих ссылок, которые коммутируются к каналу одиночного экспорта.

Например, если большой объем пульсирующего трафика поступает в гигабитный интерфейс и переключается на интерфейс 100 Мбит/с, это может вызвать увеличение отбрасывания исходящего трафика на интерфейсе 100 Мбит/с. Это происходит потому, что очередь выхода на указанном интерфейсе подавляется избыточным трафиком вследствие несовпадений скорости входящей и исходящей полосы пропускания. Скорость трафика на выходном интерфейсном устройстве не позволяет принять все пакеты, которые необходимо отправить.

Чтобы полностью решить эту проблему, необходимо увеличить скорость передачи по линии. Однако существуют способы предотвратить, уменьшить или проконтролировать

отбрасывание данных на выходе без увеличения скорости линии. Предотвратить отбрасывание исходящих данных возможно лишь в том случае, если это является результатом передачи коротких блоков данных. Если отбрасывания данных на выходе происходят вследствие постоянной высокой скорости потока, их предотвратить нельзя. Однако можно управлять ими.

Типы организации очереди, которые вовлечены в планирование вывода на Catalyst 6500/6000

Отбрасывание остатка

Отбрасывание остатка является основным механизмом предотвращения перегрузок. Когда очереди начинают заполняться во время периодов перегрузки, отбрасывание остатка рассматривает весь трафик одинаково и не дифференцируется между классами обслуживания (CoSs). Когда очередь вывода полна, и отбрасывание остатка в действительности, пакеты отброшены, пока перегрузка не устранена, и очередь больше не полна. Отбрасывание остатка является наиболее базовым типом предотвращения перегрузки и не принимает во внимание параметра QoS.

Catalyst 6000 внедрил расширенную версию предотвращения перегрузки отбрасывания остатка, которое отбрасывает все пакеты с определенным CoS, когда достигнут определенный процент от заполнения буфера. Со взвешенным отбрасыванием остатка можно определить ряд порогов и привязать CoS к каждому порогу. В примере в этом разделе существует четыре возможных порога. Определения каждого порога:

- Когда 50 процентов буфера заполнены, порог 1 достигнут. CoS 0 и 1 назначен на этот порог.
- Когда 60 процентов буфера заполнены, порог 2 достигнут. CoS 2 и 3 назначен на этот порог.
- Когда 80 процентов буфера заполнены, порог 3 достигнут. CoS 4 и 5 назначен на этот порог.
- Когда 100 процентов буфера заполнены, порог 4 достигнут. CoS 6 и 7 назначен на этот порог.

Если буфер составляет заполненных 50 процентов, в схеме на [рисунке 1](#) отброшены все пакеты с CoS 0 или 1. Если буферы составляют заполненных 60 процентов, все пакеты с CoS 0, 1, 2, или 3 отброшены. Пакеты с CoS 6 или 7 отбрасываются при полном заполнении буферов.

Рисунок 1

Примечание: Как только заполнение буфера опускается ниже определенного порога, пакеты с связанным CoS больше не отбрасываются.

Случайное раннее обнаружение и взвешенное случайное раннее обнаружение

Взвешенное произвольное раннее обнаружение (WRED) является механизмом предотвращения перегрузок, который случайным образом отбрасывает пакеты с определенным приоритетом IP-трафика, когда буферы достигают определенного порога заполнения. WRED является комбинацией этих двух функций:

- Отбрасывание остатка
- Произвольное раннее обнаружение (RED)

КРАСНЫЙ не осведомлен о приоритетах или осведомлен о CoS. Когда пороговое значение для буфера заполняется, КРАСНЫЙ использует один из отдельных порогов. КРАСНЫЙ начинает случайным образом отбрасывать пакеты (но не все пакеты, как в отбрасывании остатка), пока не достигнут максимальный (Max.) порог. После того, как максимальный порог достигнут, все пакеты отброшены. Вероятность, что пакет отброшен увеличивается линейно с увеличением заполнения буфера выше порога. Схема на [рисунке 2](#) показывает вероятность отклонения пакета:

Рисунок 2 – вероятность сброса пакетов

Примечание: Вероятность метки в этой схеме является настраиваемой в КРАСНОМ, что означает, что наклон линейной вероятности обрыва является корректируемым.

КРАСНЫЙ и WRED являются очень полезными механизмами предотвращения перегрузок для на основе TCP трафика. Для других типов трафика RED не очень эффективен. Это вызвано тем, что КРАСНЫЙ использует преимущества механизма управления окнами что использование TCP для управления перегрузкой. КРАСНЫЙ избегает типичной перегрузки, которая происходит на маршрутизаторе, когда несколько TCP - сеансов проходят порт того жя маршрутизатор. Механизм называют синхронизацией глобальной сети. Схема на [рисунке 3](#) показывает, как КРАСНЫЙ имеет эффект сглаживания на загрузку:

Рисунок 3 – КРАСНЫЙ для предотвращения перегрузки

Для получения дополнительной информации о том, как КРАСНЫЙ может уменьшить перегрузку и пригладить трафик через маршрутизатор, обратитесь к, [Как маршрутизатор Взаимодействует с](#) разделом [TCP Обзора Предотвращения перегрузки](#) документа.

WRED подобен КРАСНОМУ в тот и определить некоторый минимум (min) пороги и, когда те минимальные пороги достигнуты, пакеты случайным образом отброшены. WRED также определяет определенные максимальные пороги и, когда те максимальные пороги достигнуты, все пакеты отброшены. WRED также осведомлен о CoS, что означает, что одно или более значений CoS добавлены к каждому минимальному порогу / пара максимального порога. Когда min-threshold превышен, пакеты случайным образом отброшены с CoS, который назначен. Рассмотрите данный пример с двумя порогами в очереди:

- CoS 0 и 1 назначен на минимальный порог 1 и на максимальный порог 1. Минимальный порог 1 установлен в 50 процентов заполнения буфера, и максимальный порог 1 установлен в 80 процентов.
- CoS 2 и 3 назначен на минимальный порог 2 и на максимальный порог 2. Минимальный порог 2 установлен в 70 процентов заполнения буфера, и максимальный порог 2 установлен в 100 процентов.

Как только буфер превышает минимальный порог 1 (50 процентов), пакеты с CoS 0 и 1 начинают случайным образом отбрасываться. Больше пакетов отброшено, когда растет использование буфера. Если минимальный порог 2 (70 процентов) достигнут, пакеты с CoS 2 и 3 начинают случайным образом отбрасываться.

Примечание: На данном этапе вероятность сброса для пакетов с CoS 0 и 1 намного выше, чем вероятность сброса для пакетов с CoS 2 или CoS 3.

Каждый раз, когда максимальный порог 2 достигнут, пакеты с CoS 0 и 1 все отброшены, в то время как пакеты с CoS 2 и 3 продолжают быть случайным образом отброшенными.

Наконец, когда 100 процентов достигнуты (максимальный порог 2), все пакеты с CoS 2 и 3 отброшены.

Схемы на [рисунке 4](#) и [рисунке 5](#) иллюстрируют пример этих порогов:

Рисунок 4 – WRED с двумя наборами минимальных порогов и максимальных порогов (два сервиса) Рисунок 5 – WRED с двумя наборами сервисов, но обоими минимальными порогами равный 0

В то время как минимальный порог было трудно закодировать к 0 процентам, ранняя реализация CatOS WRED только установила максимальный порог. Нижняя часть схемы на [рисунке 5](#) выделяет последующее поведение.

Примечание: Вероятность сброса для пакета является всегда непустой, потому что эта вероятность всегда выше минимального порога. Это поведение было исправлено в версии программного обеспечения 6.2 и позже.

Весовая круговая система

Взвешенный алгоритм кругового обслуживания (WRR) является другим механизмом для планирования вывода на Catalyst 6000. Механизм WRR работает с двумя и более очередями. Очереди для WRR освобождены в порядке круговой очереди, и можно настроить вес для каждой очереди. По умолчанию порты имеют две очереди WRR на Catalyst 6000. По умолчанию:

- Служить высокоприоритетной очереди WRR 70 процентов времени
- Служить низкоприоритетной очереди WRR 30 процентов времени

Схема на [рисунке 6](#) показывает WRR, который имеет три очереди, которые подаются в WRR форма. Очередь с высоким приоритетом (красные пакеты) передает больше пакетов, чем две других очереди:

Рисунок 6 – планирование вывода: WRR

Примечание: Большинство этих 6500 линейных карт действительно внедряет WRR на пропускную способность. Эта реализация WRR на пропускную способность означает, что каждый раз планировщик позволяет очереди передавать пакеты, определенное число байтов позволены быть переданным. Это количество байтов может представлять несколько пакетов. Например, при передаче 5120 байтов в одном повороте можно передать три 1518 пакетов в 1 байт для в общей сложности 4554 байтов. Байты в избытке потеряны (5120 – 4554 = 566 байтов). Поэтому с некоторым экстремальным весом (как 1 процент для очереди 1 и 99 процентов для очереди 2), точный настроенный вес не может быть достигнут. Этот сбой для достижения к точному весу часто имеет место для больших пакетов.

Некоторые линейные карты следующего поколения, как 6548 RJ-45, преодолевают это ограничение посредством реализации взвешенного алгоритма кругового обслуживания дефицита (DWRR). DWRR передает от очередей, но не исчерпал ресурсы очередь с низким приоритетом. DWRR отслеживает очередь с низким приоритетом, которая находится под передачей и компенсирует в следующем кругу.

Очередь с жестким приоритетом

Другой тип очереди в Catalyst 6000, очереди строго по приоритету, всегда освобождается сначала. Как только существует пакет в очереди строго по приоритету, пакет передан.

WRR или Очереди WRED проверены только после того, как очередь строго по приоритету освобождена. После того, как каждый пакет передан или от очереди WRR или от Очереди WRED, очередь строго по приоритету проверена и освобождена, при необходимости.

Примечание: Все линейные карты с организацией очереди вводят подобный 1p2q1 т, 1p3q8 т, и DWRR использования на 1p7q8 т. Другие линейные карты используют стандартный WRR.

[Возможность организации очереди вывода других линейных карт на Catalyst 6000](#)

[возможности команды show port](#)

Если вы не уверены в возможности организации очереди порта, можно выполнить команду **show port capabilities**. Это - выходные данные от команды на линейной карте WS-X6408-GBIC:

```
Model                WS-X6408-GBIC
Port                 4/1
Type                 No GBIC
Speed                1000
Duplex                full
Trunk encap type     802.1Q,ISL
Trunk mode            on,off,desirable,auto,nonegotiate
Channel              yes
Broadcast suppression percentage(0-100)
Flow control          receive-(off,on,desired),send-(off,on,desired)
Security              yes
MembershIP            static,dynamic
Fast start            yes
QoS scheduling        rx-(1q4t),tx-(2q2t)
CoS rewrite           yes
ToS rewrite           DSCP
UDLD                  yes
SPAN                  source,destination
COPS port group      none
```

Этому порту выводили тип организации очереди, который называют 2q2 т.

[Поймите возможность организации очереди порта](#)

Существует несколько типов очередей, которые доступны на коммутаторах Catalyst 6500/6000. Таблицы в этом разделе могут стать неполными, поскольку освобождены карты новой линии. Карты новой линии могут представить новые комбинации организации очереди. Для текущего описания всей организации очереди, которая доступна для модулей коммутатора Catalyst 6500/6000, обратитесь к разделу *QoS Настройки* для вашей Версии CatOS [Документации Программного обеспечения серии Catalyst 6500](#).

Примечание: Модуль сред связи (СММ) Cisco не поддерживает все Характеристики QoS. Проверьте Комментарии к выпуску для своего определенного выпуска ПО для определения функций, которые поддерживаются.

В таблице приводится система обозначений для архитектуры порта QoS:

Язы Tx1/ Rx2	Обозначение очереди	Нет. из очередей	Приоритетная очередь	Нет. из очередей WRR	Нет. и тип порога для очередей WRR
Tx	2q2 т	2	—	2	2, настраиваемое отбрасывание остатка
Tx	1p2q2 т	3	1	2	2, настраиваемый WRED
Tx	1p3q1 т	4	1	3	1, настраиваемый WRED
Tx	1p2q1 т	3	1	2	1, настраиваемый WRED
Rx	1q4 т	1	—	1	4, настраиваемое отбрасывание остатка
Rx	1p1q4 т	2	1	1	4, настраиваемое отбрасывание остатка
Rx	1p1q0 т	2	1	1	Не настраивается
Rx	1p1q8 т	2	1	1	8, настраиваемый WRED

¹ Tx = передача.

2 Rx = принятый.

Эта таблица приводит все модули и типы очереди в Rx и Стороне TX интерфейса или порта:

Модуль	Входные	Выходные
--------	---------	----------

	очереди (Rx)	очереди (Tx)
WS-X6K-S2-PFC2	1p1q4 т	1p2q2 т
WS-X6K-SUP1A-2GE	1p1q4 т	1p2q2 т
WS-X6K-SUP1-2GE	1q4 т	2q2 т
WS-X6501-10GEX4	1p1q8 т	1p2q1 т
WS-X6502-10GE	1p1q8 т	1p2q1 т
WS-X6516-GBIC	1p1q4 т	1p2q2 т
WS-X6516-GE-TX	1p1q4 т	1p2q2 т
WS-X6416-GBIC	1p1q4 т	1p2q2 т
WS-X6416-GE-MT	1p1q4 т	1p2q2 т
WS-X6316-GE-TX	1p1q4 т	1p2q2 т
WS-X6408A-GBIC	1p1q4 т	1p2q2 т
WS-X6408-GBIC	1q4 т	2q2 т
WS-X6524-100FX-MM	1p1q0 т	1p3q1 т
WS-X6324-100FX-SM	1q4 т	2q2 т
WS-X6324-100FX-MM	1q4 т	2q2 т
WS-X6224-100FX-MT	1q4 т	2q2 т
WS-X6548-RJ-21	1p1q0 т	1p3q1 т
WS-X6548-RJ-45	1p1q0 т	1p3q1 т
WS-X6348-RJ-21	1q4 т	2q2 т
WS-X6348-RJ21V	1q4 т	2q2 т
WS-X6348-RJ-45	1q4 т	2q2 т
WS-X6348-RJ21V	1q4 т	2q2 т
WS-X6148-RJ-45V	1q4 т	2q2 т
WS-X6148-RJ21V	1q4 т	2q2 т
WS-X6248-RJ-45	1q4 т	2q2 т
WS-X6248A-TEL	1q4 т	2q2 т
WS-X6248-TEL	1q4 т	2q2 т
WS-X6024-10FL-MT	1q4 т	2q2 т

[Создайте QoS на Catalyst 6500/6000](#)

Три поля на Catalyst 6500/6000 используются для создания QoS:

- Приоритет IP-трафика — первые три бита Поля типа обслуживания (TOS) в IP - заголовке
- Кодовая точка дифференцированных сервисов (DSCP) — первые шесть битов поля ToS в IP - заголовке
- CoS — три бита использовали на Уровне 2 (L2) уровень Эти три бита являются или частью заголовка Протокола ISL или являются в IEEE 802.1Q (dot1q) меткой. Нет никакого CoS в без меток Пакет Ethernet.

[Механизм планирования вывода на Catalyst 6500/6000](#)

Когда кадр передан от шины данных, которая будет передана, CoS пакета является единственным параметром, который рассматривают. Пакет тогда проходит планировщика, который выбирает очередь, в которую помещен пакет. Поэтому помните, что планирование вывода и все механизмы, которые обсуждает этот документ, только осведомлены о CoS.

Catalyst 6500/6000 с Функциональной Картой Многоуровневого Коммутатора (MSFC) использует внутренний DSCP для классификации пакета. Catalyst 6500/6000, который настроен с включенным QoS, назначает DSCP-значение, когда решение по перенаправлению сделано на уровне PFC. Этот DSCP назначен на любой пакет, который включает пакеты не-IP и сопоставлен с CoS для включения планирования вывода. Можно настроить сопоставление от DSCP до значений CoS на Catalyst 6500/6000. При отъезде значения по умолчанию можно получить CoS из DSCP. Используется следующая формула:

$DSCP_value / 8$

Кроме того, DSCP-значение сопоставлено в CoS исходящего пакета, если пакет является пакетом IP, который является ISL или dot1q теговый (несобственный VLAN). DSCP-значение также записано в поле ToS IP - заголовка.

Схема на [рисунке 7](#) показывает очередь на 1p2q2 т. Очереди WRR освобождены с использованием планировщика WRR. Существует также арбитр, который проверяет между каждым пакетом от очередей WRR, чтобы определить, существует ли что-то в очереди строго по приоритету.

Рисунок 7

1. Поле ToS переписано в IP - заголовке и 802.1p/ISL поле CoS.
2. Очередь планирования и порог выбраны на основе CoS через конфигурируемую карту.
3. У каждой очереди есть настраиваемый размер и пороги, и у некоторых очередей есть WRED.
4. Исключение из очереди WRR использования между двумя очередями.
5. Исходящая инкапсуляция может быть dot1q, ISL или ни одним.

[Настройка, мониторинг планирование выходных данных на Catalyst 6500/6000](#)

[Конфигурация по умолчанию для QoS на Catalyst 6500/6000](#)

Этот раздел предоставляет пример выходных данных от конфигурации QoS по умолчанию на Catalyst 6500/6000, в дополнение к информации о том, что эти значения среднее значение и как можно настроить значения.

QoS отключено по умолчанию при выдаче этой команды:

```
set qos disable
```

Команды в этом списке показывают назначение по умолчанию для каждого CoS в порту на 2q2 т. Очередь 1 имеет CoS 0 и 1 назначенный в его первый порог и назначила CoS 2 и 3 в его второй порог. Очереди 2 назначили CoS 4 и 5 в его первый порог и назначила CoS 6 и 7 в его второй порог:

```
set qos map 2q2t tx 1 1 cos 0
set qos map 2q2t tx 1 1 cos 1
set qos map 2q2t tx 1 2 cos 2
set qos map 2q2t tx 1 2 cos 3
set qos map 2q2t tx 2 1 cos 4
set qos map 2q2t tx 2 1 cos 5
set qos map 2q2t tx 2 2 cos 6
set qos map 2q2t tx 2 2 cos 7
```

Эти команды отображают пороговый уровень по умолчанию на порту на 2q2 т для каждой очереди:

```
set qos drop-threshold 2q2t tx queue 1 80 100
set qos drop-threshold 2q2t tx queue 2 80 100
```

Можно назначить вес по умолчанию на каждую из очередей WRR. Выполните эту команду для присвоения веса по умолчанию для очереди 1 и очереди 2:

Примечание: Очередь с низким приоритетом подается 5/260 процента времени, и очередь с высоким приоритетом подается 255/260 процента времени.

```
set qos wrr 2q2t 5 255
```

Доступность полного буфера разделена среди этих двух очередей. Очередь с низким приоритетом правильно назначена на 80 процентов буферов, которые доступны, потому что это - очередь, которой, скорее всего, буферизуют пакеты и находящийся в течение некоторого времени. Выполните эту команду для определения доступности:

```
set qos txq-ratio 2q2t 80 20
```

Можно просмотреть подобные параметры настройки для порта на 1p2q2 т в этой конфигурации:

```
set qos map 1p2q2t tx 1 1 cos 0
set qos map 1p2q2t tx 1 1 cos 1
set qos map 1p2q2t tx 1 2 cos 2
set qos map 1p2q2t tx 1 2 cos 3
set qos map 1p2q2t tx 2 1 cos 4
set qos map 1p2q2t tx 3 1 cos 5
set qos map 1p2q2t tx 2 1 cos 6
set qos map 1p2q2t tx 2 2 cos 7
set qos wrr 1p2q2t 5 255
set qos txq-ratio 1p2q2t 70 15 15
```

```
set qos wred 1p2q2t tx queue 1 80 100
```

```
set qos wred 1p2q2t tx queue 2 80 100
```

Примечание: По умолчанию (голосовой трафик) CoS 5 назначен на очередь строго по приоритету.

!--- конфигурацию

Первое действие настройки должно включить QoS. Помните, что по умолчанию служба QoS отключена. Когда QoS отключено, сопоставление CoS не важно. Существует одна очередь, которая подается в качестве FIFO, и все пакеты отброшены там.

```
bratan> (enable) set qos enable
```

```
QoS is enabled
```

```
bratan> (enable) show qos status
```

```
QoS is enabled on this switch
```

Значение CoS должно быть назначено на очередь или порог для всех типов очереди. Сопоставление, которое определено для типа на 2q2 т порта, не применено ни к какому порту на 1p2q2 т. Кроме того, сопоставление, которое сделано для 2q2 т, применено ко всем портам, которые имеют механизм организации очередей на 2q2 т. Введите следующую команду:

```
set qos map queue_type tx Q_number threshold_number cos value
```

Примечание: Очереди всегда перечисляются, чтобы запуститься с самой низкой очереди с приоритетами и закончиться очередью строго по приоритету, которая доступна. Например:

- Очередь 1 – это очередь WRR с меньшим приоритетом
- Очередь 2 – это очередь WRR с большим приоритетом
- Очередь 3 имеет строгий приоритет

Необходимо повторить эту операцию для всех типов очередей. В противном случае вы поддерживаете присвоение CoS по умолчанию. Вот пример для 1p2q2 т:

!--- конфигурацию

```
set qos map 1p2q2t tx 1 1 cos 0
```

```
!--- This is the low-priority WRR queue threshold 1, CoS 0 and 1. set qos map 1p2q2t tx 1 1 cos 1 and 1
```

```
set qos map 1p2q2t tx 1 2 cos 2
```

```
!--- This is the low-priority WRR queue threshold 2, CoS 2 and 3. set qos map 1p2q2t tx 1 2 cos 3 and 3
```

```
set qos map 1p2q2t tx 2 1 cos 4
```

```
!--- This is the high-priority WRR queue threshold 1, CoS 4. set qos map 1p2q2t tx 3 1 cos 5
```

```
!--- This is the strict priority queue, CoS 5. set qos map 1p2q2t tx 2 1 cos 6
```

```
!--- This is the high-priority WRR queue threshold 2, CoS 6. set qos map 1p2q2t tx 2 2 cos 7 and 7
```

Вывод консоли

```
tamer (enable) set qos map 1p2q2t tx 1 1 cos 0
```

QoS tx priority queue and threshold mapped to cos successfully

Необходимо настроить вес WRR для двух очередей WRR. Введите следующую команду:

```
set qos wrr Q_type weight_1 weight_2
```

Вес 1 относится к очереди 1, которая должна быть очередью WRR с меньшим приоритетом. Weight_1 должен всегда быть ниже, чем weight_2. Вес может принять любое значение между 1 и 255. Можно назначить процент с этими формулами:

- Очередь 1:

```
set qos wrr Q_type weight_1 weight_2
```
- Очередь2:

```
set qos wrr Q_type weight_1 weight_2
```

Необходимо также определить вес для различных типов очередей. Вес не должен быть тем же. Например, для 2q2 т, где очередь 1 подается, 30 процентов времени и очереди 2 подаются 70 процентов времени, можно выполнить эту команду для определения веса:

```
set qos wrr 2q2t 30 70
```

!--- This ensures that the high-priority WRR queue is served 70 percent of the time !--- and that the low-priority WRR queue is served 30 percent of the time.

Вывод консоли

```
tamer (enable) set qos wrr 2q2t 30 70
```

QoS wrr ratio is set successfully

Также необходимо определить коэффициент очереди передачи, который обращается к способу, которым буферы разделены среди других очередей. Введите следующую команду:

```
set qos txq-ratio port_type queue1_val queue2_val ... queueN_val
```

Примечание: Если у вас есть три очереди (1p2q2 т), необходимо установить высокоприоритетную очередь WRR и очередь строго по приоритету на том же уровне для причин связанная с аппаратным обеспечением.

!--- конфигурацию

```
set qos txq-ratio 1p2q2t 70 15 15
```

!--- This gives 70 percent of the buffer of all 1p2q2t ports to the low-priority WRR !--- queue and gives 15 percent to each of the other two queues. set qos txq-ratio 2q2t 80 20 !--- This gives 80 percent of the buffer to the low-priority queue, !--- and gives 20 percent of the buffer to the high-priority queue.

Вывод консоли

```
tamer (enable) set qos txq-ratio 1p2q2t 70 15 20
```

Queue ratio values must be in range of 1-99 and add up to 100

Example: set qos txq-ratio 2q2t 20 80

```
tamer (enable) set qos txq-ratio 1p2q2t 70 30 30
```

Queue ratio values must be in range of 1-99 and add up to 100

Example: set qos txq-ratio 2q2t 20 80

```
tamer (enable) set qos txq-ratio 1p2q2t 80 10 10
```

QoS txq-ratio is set successfully

Поскольку эти выходные данные консоли иллюстрируют, сумма значений очереди должна быть 100. Оставьте самую большую часть буферов для низкоприоритетной очереди WRR, потому что этой очереди нужна большая часть буферизации. Другие очереди обслуживаются с более высоким приоритетом.

Последний шаг – это конфигурирование порогового уровня для очереди WRED или для очереди отбрасывания остатка. Введите следующие команды:

```
set qos wred port_type [tx] queue q_num thr1 thr2 ... thrn
```

```
set qos drop-threshold port_type tx queue q_num thr1 ... thr2
```

!--- конфигурацию

```
set qos drop-threshold 2q2t tx queue 1 50 80
```

!--- For low-priority queues in the 2q2t port, the first threshold is defined at 50 !--- percent and the second threshold is defined at 80 percent of buffer filling. set qos drop-threshold 2q2t tx queue 2 40 80

!--- For high-priority queues in the 2q2t port, the first threshold is defined at 40 !--- percent and the second threshold is defined at 80 percent of buffer filling. set qos wred 1p2q2t tx queue 1 50 90

!--- The commands for the 1p2q2t port are identical. set qos wred 1p2q2t tx queue 2 40 80

Вывод консоли

```
tamer (enable) set qos drop-threshold 2q2t tx queue 1 50 80
```

Transmit drop thresholds for queue 1 set at 50% 80%

```
tamer (enable) set qos drop-threshold 2q2t tx queue 2 40 80
```

Transmit drop thresholds for queue 2 set at 40% 80%

```
tamer (enable) set qos wred 1p2q2t tx queue 1 50 90
```

WRED thresholds for queue 1 set to 50 and 90 on all WRED-capable 1p2q2t ports

```
tamer (enable) set qos wred 1p2q2t tx queue 2 40 80
```

WRED thresholds for queue 2 set to 40 and 80 on all WRED-capable 1p2q2t ports

Команда set qos wred 1p2q2t tx queue 2 40 80 работает в сочетании с CoS для сопоставления порогов. Например, при выдаче команд в списке ниже вы гарантируете, что — на порту на 1p2q2 т в направлении передачи — пакеты с CoS 0,1, 2, и 3 переданы в первой очереди (низкая очередь WRR). Когда буферы в той очереди составляют заполненных 50 процентов, WRED начинает отбрасывать пакеты с CoS 0 и 1. Пакеты с CoS 2 и 3 отброшены только, когда буферы на очереди составляют заполненных 90 процентов.

```
set qos map 1p2q2t tx 1 1 cos 0
```

```
set qos map 1p2q2t tx 1 1 cos 1
```

```
set qos map 1p2q2t tx 1 2 cos 2
```

```
set qos map lp2q2t tx 1 2 cos 3
```

```
set qos wred lp2q2t tx queue 1 50 90
```

Планирование вывода монитора и проверяет конфигурацию

Простая команда для использования для проверки текущей динамической настройки для планирования вывода порта является *mod/port* времени выполнения `show qos info`. Команда отображает эту информацию:

- Тип организации очереди на порту
- Сопоставление CoS другим очередям и порогам
- Совместное использование буфера
- Вес WRR

В данном примере значения в 20-процентном WRR для очереди 1 и 80-процентном WRR для очереди 2:

```
tamer (enable) show qos info runtime 1/1
```

```
Run time setting of QoS:
```

```
QoS is enabled
```

```
Policy Source of port 1/1: Local
```

```
Tx port type of port 1/1 : lp2q2t
```

```
Rx port type of port 1/1 : lp1q4t
```

```
Interface type: port-based
```

```
ACL attached:
```

```
The qos trust type is set to untrusted
```

```
Default CoS = 0
```

```
Queue and Threshold Mapping for lp2q2t (tx):
```

```
Queue   Threshold   CoS
```

```
-----
```

```
1         1         0 1
```

```
1         2         2 3
```

```
2         1         4 6
```

```
2         2         7
```

```
3         1         5
```

```
Queue and Threshold Mapping for lp1q4t (rx):
```

```
All packets are mapped to a single queue
```

```
Rx drop thresholds:
```

```
Rx drop thresholds are disabled
```

```
Tx drop thresholds:
```

```
Tx drop-thresholds feature is not supported for this port type
```

```
Tx WRED thresholds:
```

```
Queue #           Thresholds - percentage (* abs values)
```

```
-----
```

```
1           80% (249088 bytes) 100% (311168 bytes)
```

```
2           80% (52480 bytes) 100% (61440 bytes)
```

```
Queue Sizes:
```

```
Queue #           Sizes - percentage (* abs values)
```

```
-----
```

```
1           70% (311296 bytes)
```

```
2           15% (65536 bytes)
```

```
3           15% (65536 bytes)
```

```
WRR Configuration of ports with speed 1000Mbps:
```

```
Queue #           Ratios (* abs values)
```

```
-----
```

```
1           20 (5120 bytes)
```

```
2           80 (20480 bytes)
```

(*) Runtime information may differ from user configured setting due to hardware granularity.

```
tamer (enable)
```

В следующем примере обратите внимание, что веса WRR не являются значением по умолчанию 1. Веса были установлены в значения 20 для очереди 1 и 80 для очереди 2. Данный пример использует генератор трафика для передачи 2 Гбит трафика к Catalyst 6000. Эти 2 Гбита трафика должны выйти через порт 1/1. Поскольку порт 1/1 превышен, много пакетов отброшены на (1 Гбит/с). Команда **show mac** показывает, что существует много отбрасывания выходных данных:

```
tamer (enable) show mac 1/1
```

Port	Rcv-Unicast	Rcv-Multicast	Rcv-Broadcast
1/1	0	1239	0

Port	Xmit-Unicast	Xmit-Multicast	Xmit-Broadcast
1/1	73193601	421	0

Port	Rcv-Octet	Xmit-Octet
1/1	761993	100650803690

MAC	Dely-Exced	MTU-Exced	In-Discard	Out-Discard
1/1	0	-	0	120065264

```
Last-Time-Cleared
```

```
-----  
Fri Jan 12 2001, 17:37:43
```

Проанализируйте отброшенные пакеты. Это - то, как разделен предложенный вид трафика:

- 1 Гбит трафика с приоритетом IP-трафика 0
- 250 Мбит трафика с приоритетом IP-трафика 4
- 250 Мбит трафика с приоритетом IP-трафика 5
- 250 Мбит трафика с приоритетом IP-трафика 6
- 250 Мбит трафика с приоритетом IP-трафика 7

Согласно сопоставлению CoS, передается этот трафик:

- 1 Гбит трафика для организации очереди 1 порога 1
- 0 Мбит трафика для организации очереди 1 порога 2
- 500 Мбит трафика для организации очереди 2 порогов 1
- 250 Мбит трафика для организации очереди 2 порогов 2
- 250 Мбит трафика для организации очереди 3 (очередь строго по приоритету)

Коммутатор должен доверять принимаемому трафику так, чтобы приоритеты входящего IP были сохранены в коммутаторе и использовались для сопоставления со значением CoS для планирования вывода.

Примечание: Приоритеты IP по-умолчанию к сопоставлению CoS являются приоритетом IP-трафика, равняется CoS.

Выполните команду **show qos stat 1/1** для наблюдения пакетов, которые были отброшены и приблизительный процент:

- В настоящий момент в очереди 3 (CoS 5) не отброшено ни одного пакета.
- 91.85 процента отброшенных пакетов являются CoS 0 пакетов в очереди 1.
- 8 процентов отброшенных пакетов являются CoS 4 и 6 в очереди 2, порог 1.

- 0.15 процента отброшенных пакетов являются CoS 7 в очереди 2, порог 2.

Эти выходные данные иллюстрируют использование команды:

```
tamer (enable) show qos stat 1/1
```

```
Tx port type of port 1/1 : lp2q2t
```

```
Q3T1 statistics are covered by Q2T2.
```

```
Q #      Threshold #:Packets dropped
-----
1        1:110249298 pkts, 2:0 pkts
2        1:9752805 pkts, 2:297134 pkts
3        1:0 pkts
```

```
Rx port type of port 1/1 : lp1q4t
```

```
Rx drop threshold counters are disabled for untrusted ports
```

```
Q #      Threshold #:Packets dropped
-----
1        1:0 pkts, 2:0 pkts, 3:0 pkts, 4:0 pkts
2        1:0 pkts
```

При возврате веса WRR к значению по умолчанию после того, как счетчики были очищены, только 1 процент отброшенных пакетов происходит в очереди 2 вместо 8 процентов, которые появились ранее:

Примечание: Значение по умолчанию 5 для очереди 1 и 255 для очереди 2.

```
tamer (enable) show qos stat 1/1
```

```
TX port type of port 1/1 : lp2q2t
```

```
Q3T1 statistics are covered by Q2T2
```

```
Q #      Threshold #:Packets dropped
-----
1        1:2733942 pkts, 2:0 pkts
2        1:28890 pkts, 2:6503 pkts
3        1:0 pkts
```

```
Rx port type of port 1/1 : lp1q4t
```

```
Rx drop threshold counters are disabled for untrusted ports
```

```
Q #      Threshold #:Packets dropped
-----
1        1:0 pkts, 2:0 pkts, 3:0 pkts, 4:0 pkts
2        1:0 pkts
```

[Использование диспетчеризации исходящих потоков для уменьшения задержек и фазового дрожания](#)

Пример в [Планировании вывода Монитора](#) раздела [и Проверяет, что Конфигурация](#) демонстрирует преимущество реализации планирования вывода, которая избегает отбрасывания VoIP или критически - важного трафика в случае превышения подписки порта вывода. Превышение подписки нечасто происходит в стандартной сети, особенно на Гигабитной ссылке. Обычно, превышение подписки только происходит в течение времен пиковой нагрузки или во время пакетов трафика в очень коротком периоде времени.

Даже когда перегрузки не возникают, диспетчеризация исходящих потоков может принести большую пользу в сети со сквозной реализацией службы QoS. Планирование вывода помогает уменьшать задержку и дрожание. В этом разделе приводятся примеры, показывающие, как диспетчеризация исходящих потоков может способствовать уменьшению задержек и фазового дрожания.

[Уменьшение задержек](#)

Задержка пакета увеличена, к тому времени, когда “потеряно” в буфере каждого коммутатора во время ожидания передачи. Например, небольшой голосовой пакет с классом обслуживания 5 посылается из порта во время передачи большого файла или резервной копии. Если у вас нет QoS для порта вывода, и если вы предполагаете, что небольшой речевой пакет помещен в очередь после 10 больших 1500 пакетов в 1 байт можно легко вычислить Гигабитное время скорости, которое необходимо для передачи этих 10 больших пакетов:

```
tamer (enable) show qos stat 1/1
```

```
TX port type of port 1/1 : lp2q2t
Q3T1 statistics are covered by Q2T2
Q #           Threshold #:Packets dropped
-----
1             1:2733942 pkts, 2:0 pkts
2             1:28890 pkts, 2:6503 pkts
3             1:0 pkts
Rx port type of port 1/1 : lp1q4t
Rx drop threshold counters are disabled for untrusted ports
Q #           Threshold #:Packets dropped
-----
1             1:0 pkts, 2:0 pkts, 3:0 pkts, 4:0 pkts
2             1:0 pkts
```

Если этот пакет должен пересечь восемь или девять коммутаторов, в то время как он проходит через сеть, задержка приблизительно 1 мс может закончиться. Эта сумма считает только задержки очереди вывода коммутатора, который скрещен в сети.

Примечание: Если необходимо поместить те же 10 больших пакетов в очередь на интерфейсе на 10 Мбит/с (например, с IP-телефоном и со связанным ПК), задержка, которая представлена:

```
tamer (enable) show qos stat 1/1
```

```
TX port type of port 1/1 : lp2q2t
Q3T1 statistics are covered by Q2T2
Q #           Threshold #:Packets dropped
-----
1             1:2733942 pkts, 2:0 pkts
2             1:28890 pkts, 2:6503 pkts
3             1:0 pkts
Rx port type of port 1/1 : lp1q4t
Rx drop threshold counters are disabled for untrusted ports
Q #           Threshold #:Packets dropped
-----
1             1:0 pkts, 2:0 pkts, 3:0 pkts, 4:0 pkts
2             1:0 pkts
```

Реализация планирования вывода гарантирует, что голосовые пакеты с CoS 5 помещены в очередь строго по приоритету. Это размещения гарантируют, что эти пакеты переданы перед любыми пакетами с CoS меньше чем 5, который уменьшает задержки.

Уменьшение фазового дрожания

Другое важное преимущество реализации планирования вывода - то, что она уменьшает дрожание. Дрожание является изменением в задержке, которая наблюдается для пакетов в том же потоке. Схема на [рисунке 8](#) показывает пример сценария того, как планирование вывода может уменьшить дрожание:

Рис. 8

В этом сценарии существует два потока, которые должен передать одиночный порт вывода:

- Один поток голосовых данных, который поступает на порте Ethernet 10 Мбит/с
- Один поток FTP, который поступает на Восходящем канале связи Ethernet на 1 Гбит/с

Оба потока покидают коммутатор через один и тот же выходной порт. Данный пример показывает то, что может произойти без использования планирования вывода. Все большие пакеты данных могут быть чередованы между двумя голосовыми пакетами, который создает дрожание в приеме голосового пакета от того же поток. Существует большая задержка между приемом пакета n и пакетом $n+1$, поскольку коммутатор передает большой пакет данных. Однако, задержка между пакетами $n + 1$ и $n + 2$ пренебрежимо мала. Это приводит к фазовому дрожанию в потоке голосового трафика. Этой проблемы можно легко избежать, если использовать очередь со строгим приоритетом. Гарантируйте, что значение CoS голосовых пакетов сопоставлено с очередью строго по приоритету.

Дополнительные сведения

- [Диспетчеризация исходящих потоков с помощью службы QoS на коммутаторах Catalyst серий 6500/6000 под управлением системы Cisco IOS](#)
- [Общие сведения о качестве обслуживания \(QoS\) для коммутаторов серии Catalyst 6000](#)
- [Страницы поддержки продуктов LAN](#)
- [Страница поддержки коммутационных решений для локальной сети](#)
- [Cisco Systems – техническая поддержка и документация](#)