

Сравнение надзора за трафиком и формирования трафика для ограничения пропускной способности

Содержание

[Введение](#)

[Перед началом работы](#)

[Условные обозначения](#)

[Предварительные условия](#)

[Используемые компоненты](#)

[Применение политик и формирование](#)

[Критерий выбора](#)

[Частота обновления Token](#)

[Формирование трафика](#)

[Применение к трафику политик](#)

[Сравнение минимального и максимального контроля полосы пропускания](#)

[Дополнительные сведения](#)

Введение

В данном документе объясняются функциональные различия между формированием трафика и соблюдением политики трафика, оба из которых ограничивают скорость вывода. Хотя оба метода используют алгоритм Token Bucket в качестве измерителя трафика для оценки скорости пакета, между ними имеются важные функциональные различия. ([Алгоритм Token Bucket описан в разделе "Что такое алгоритм Token Bucket?"](#)).

Перед началом работы

Условные обозначения

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

Предварительные условия

Для данного документа отсутствуют предварительные условия.

Используемые компоненты

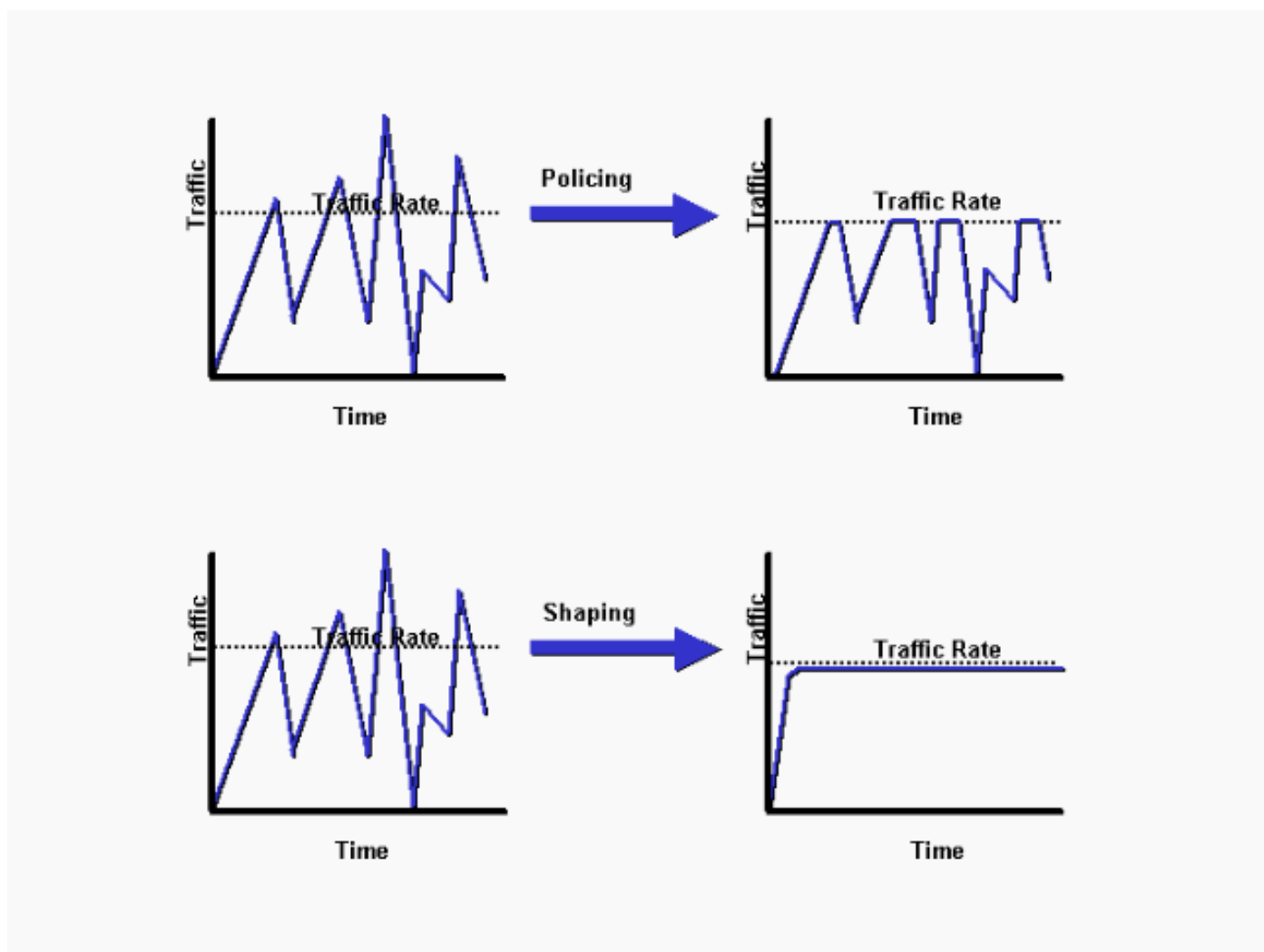
Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям

программного обеспечения и оборудования.

Сведения, содержащиеся в данном документе, были получены с устройств в специальной лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. При работе с реальной сетью необходимо полностью осознавать возможные результаты использования всех команд.

Применение политик и формирование

На следующей диаграмме показано главное различие. Ограничение скорости трафика приводит к распространению пакетной передачи. Когда скорость трафика достигает заданной максимальной скорости, лишний трафик отбрасывается (или помечается). Результат скорости вывода отображается в виде пилообразной линии с гребнями и впадинами. В противоположность применению политик, в процессе формирования трафика избыточные пакеты помещаются в очередь и планируются для последующей передачи. Результатом процесса формирования трафика является более ровная выходная скорость передачи пакетов.



Формирование подразумевает наличие очереди и достаточно большой области памяти для буферизации задержанных пакетов, тогда как для упорядочения трафика этого не требуется. Queueing - это понятие из области исходящей передачи; пакеты идя из интерфейса, получаютя queued и могут быть оформены. Ко входящему трафику на интерфейсе может быть применено только соблюдение политики. Убедитесь в наличии достаточного объема памяти при включении функции формирования пакетов. Кроме того,

формирование требует функции планирования для последующей передачи любого задержанного пакета. Эта функция планирования позволяет разбить очередь формирования трафика на несколько различных очередей. Примерами функций планирования являются очередь с весами, основанными на классах (CBWFQ), и очередь с малым временем ожидания (LLQ).

Критерий выбора

В нижеприведенной таблице указаны различия между функцией формирования трафика и функцией соблюдения политики трафика, что поможет выбрать наиболее подходящее решение.

	Формирование	Применение политик
Цель	Буферизация и постановка в очередь избыточных пакетов сверх оговоренной скорости.	Сброс (или отметка) избыточных пакетов выше фиксированной скорости. Не обеспечивает буферизацию.* *
Частота обновления Token	Увеличивается в начале временного интервала. (Требуется минимальное количество интервалов.)	Постоянно на основе следующей формулы: $1 /$
Значения маркеров	Выражается в битах в секунду.	Выражается в байтах.
Параметры настройки	<ul style="list-style-type: none"> • команда <code>shape</code> в модульном интерфейсе командной строки качества обслуживания (MQC) для выполнения формирования на основе класса. • команда <code>frame-relay traffic-shape</code> для реализации формирования трафика с ретрансляцией кадров (FRTS). • команда <code>traffic-</code> 	<ul style="list-style-type: none"> • команда <code>police</code> в MQC для реализации классовой политики. • команда <code>rate-limit</code> для реализации функции согласованной скорости доступа (CAR).

	shape для реализации общего формирования трафика (GTS).	
Применимо ко входящему трафику	Нет	Да
Применимо на исходящем	Да	Да
Пакеты	Контролирует пакеты, усредняя скорость вывода по крайней мере по восьми временным интервалам. Использует "дырявое ведро" для задержки трафика, чем достигается эффект сглаживания.	Распространяет пакеты. Не выполняется сглаживание.
Преимущества	Меньше вероятности, что излишние пакеты будут отброшены, так как излишние пакеты находятся в буфере. (Буферы распределяют пакеты по длине очереди. Может произойти отбрасывание, если излишний трафик передается на большой скорости). Как правило, избегает повторной передачи ввиду отброшенных пакетов.	Управляет скоростью передачи выходного трафика путем отбрасывания пакетов. Избегает задержек благодаря организации очереди.
Недостатки	Может вызвать задержку из-за очереди, особенно при очередях с большой глубиной.	Отбрасывает излишние пакеты (при соответствующих настройках), сокращая размеры окна TCP и общую скорость передачи выходных данных

		соответствующего потока трафика. Использование слишком большого размера пакетов может привести к повышенной потере пакетов и значительному снижению суммарной выходной скорости передачи данных, особенно для потоков данных на базе протокола TCP.
Необязательная повторная маркировка пакетов	Нет	Да (с существующей функцией CAR).

* Хотя ограничение трафика не включает буферизацию, настроенный механизм формирования очередей применяется к соответствующим пакетам, которые необходимо поставить в очередь для сериализации на физическом интерфейсе.

Частота обновления Token

Ключевым отличием формирования трафика от применения политики является скорость пополнения маркеров. Это отличие рассматривается в данном разделе.

Как функция формирования трафика, так и функция соблюдения политики используют метафору контейнера маркеров. Сам по себе алгоритм token bucket не имеет политик отбрасывания или приоритизации. Страница технической поддержки концентраторов Cisco VPN серии 5000:

- Маркеры помещаются в сегмент с определенной частотой.
- Каждый маркер является разрешением для источника отправлять определенное количество битов в сеть.
- Чтобы отправить пакет, регулятор трафика должен быть способен удалить из участка памяти число маркеров, равное размеру пакета.
- Если на участке памяти недостаточно маркеров для отправки пакета, пакет либо ждет, пока на участке не накопится достаточно маркеров (в случае формирователя), либо пакет отбрасывается или понижается (в случае ограничителя скорости).
- "Ведро" имеет определенную вместимость. После того как буфер заполнится, поступающие в него новые маркеры будут отбрасываться и окажутся недоступны для будущих пакетов. Таким образом, в любой момент, самый большой пакет сигналов,

который источник может послать в сеть, приблизительно пропорционален размеру бакета. Буфер маркеров предполагает блочную отправку, при этом ограничивая ее темпы.

С учетом метафоры "маркерное ведро" давайте рассмотрим, каким образом формирование и определение политик добавляет маркеры в ведро.

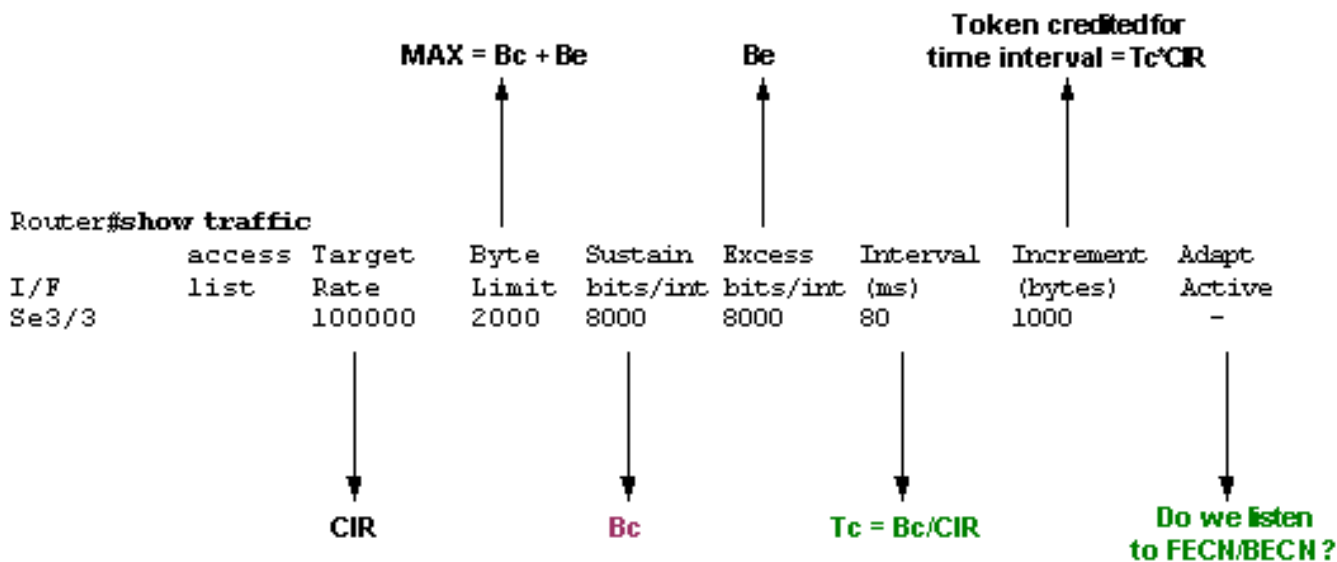
Формирование увеличивает буфер маркеров через определенные интервалы с помощью значения бит/с. Средство формирования использует следующую формулу:

$$T_c = B_c / CIR \text{ (in seconds)}$$

В данном уравнении B_c обозначает согласованный размер пакета, а CIR обозначает согласованную скорость передачи данных. [\(См. раздел Настройка формирования трафика с ретрансляцией кадров для получения дополнительных сведений\)](#). Значение T_c определяет временной интервал, в течение которого необходимо посылать биты B_c , чтобы поддержать среднюю скорость согласованной передачи данных в секундах.

Диапазон для T_c находится между 10 мс и 125 мс. С функцией формирования распределенного трафика (DTS) на устройствах Cisco серии 7500 минимальное значение T_c равно 4 мс. Маршрутизатор внутренне рассчитывает значение на основе значений CIR и B_c . Если B_c / CIR меньше 125 мс, используется значение T_c , рассчитанное с помощью этого уравнение. Если значение B_c / CIR больше или равно 125 мс, и Cisco IOS® определяет, что поток трафика будет более устойчивым при меньшем значении интервала, для него используется внутреннее значение T_c .

помощью команды `show traffic-shape` определите, использует ли ваш маршрутизатор внутреннее значение для T_c или значение, настроенное в командной строке. Приведенный ниже пример выходных данных команды `show traffic-shape` поясняется в разделе команды `show` для управления трафиком с ретрансляцией кадров.



Если избыточный пакет (B_e) настроен на другое значение, кроме 0, средство формирования позволяет хранение маркеров в контейнере, до $B_c + B_e$. Наибольшее значение, которого когда-либо может достичь маркированный блок - это $B_c + B_e$, и избыточные маркеры вырезаются. Единственный способ получить больше, чем маркеры B_c в участке памяти - не использовать все маркеры B_c во время одного или нескольких T_c . Так как область памяти маркеров пополняется всеми маркерами T_c с B_c , можно накапливать неиспользуемые маркеры до значений $B_c + B_e$ для последующего использования.

Напротив, политика на основе классов и ограничение скорости постоянно добавляют

маркеры в контейнер. Конкретно, скорость поступления маркеров рассчитывается следующим образом:

$$(time\ between\ packets < which\ is\ equal\ to\ t-t_1 > * policer\ rate) / 8\ bits\ per\ byte$$

Другими словами, если предыдущее поступление пакета произошло в t_1 , а текущее время t , контейнер обновляется значением байт $t-t_1$ на основе скорости поступления маркеров. Обратите внимание, что ограничитель трафика использует значения пакетов, указанные в байтах, а вышеприведенная формула преобразует биты в байты.

Рассмотрим пример с использованием согласованной скорости передачи CIR (или скорости ограничителя), равной 8000 бит/с и с обычным пакетом, равным 1000 байт.

```
Router(config)# policy-map police-setting Router(config-pmap)# class access-match Router(config-pmap-c)# police 8000 1000 conform-action transmit exceed-action drop
```

"Token buckets" начинает заполняться при 1000 байтах. Если поступает пакет в 450 байт, то он принимается, так как в Token Bucket доступно достаточно байт. Соответствующее действие (передача) выполняется для пакета и 450 байт удаляются из token bucket (маркерного ведра) (остается 550 байт). Если следующий пакет прибывает на 0,25 секунды позднее, 250 байт добавляются в контейнер маркеров согласно следующей формуле:

$$(0.25 * 8000) / 8$$

Согласно расчетам в контейнере маркеров остается 700 байт. Если размер следующего пакета составляет 800 байт, пакет превышает значение, и выполняется действие при превышении значения (отбрасывание). Из "маркерного ведра" байты не берутся.

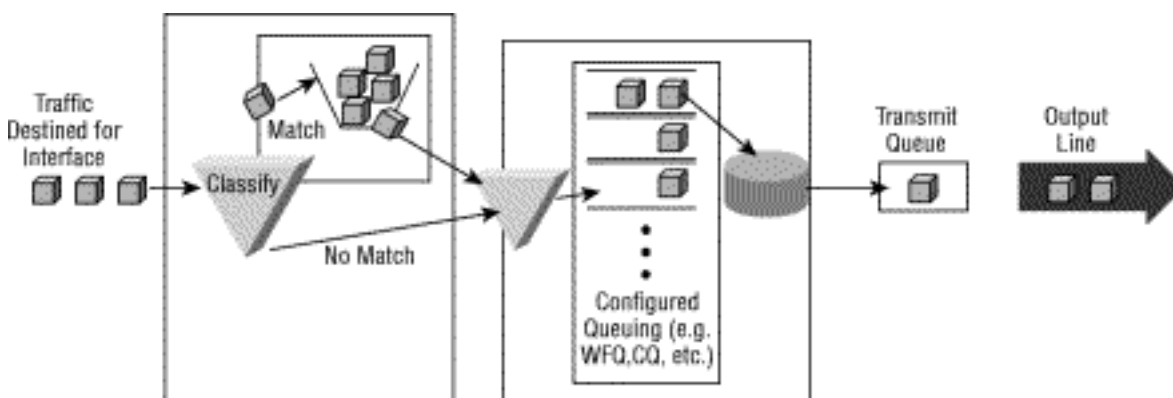
Формирование трафика

Cisco IOS поддерживает следующие методы формирования трафика:

- [Формирование общего трафика](#)
- [Frame-relay traffic-shaping](#)
- [Формирование на основе классов и Распределенное формирование на основе классов](#)

Все методы формирования трафика реализуются аналогичным образом, хотя интерфейсы командной строки (CLI) имеют небольшие различия, и в них используются разные типы очередей для содержания и формирования задерживаемого трафика. Cisco рекомендует формирование на основе классов и распределенное формирование, которые настраиваются при помощи модульного QoS CLI.

На следующей схеме показано, как политика QoS сортирует трафик по классам и ставит в очередь пакеты, переходящие границы настроенных скоростей формирования.



Применение к трафику политик

Cisco IOS поддерживает следующие методы регулировки трафика:

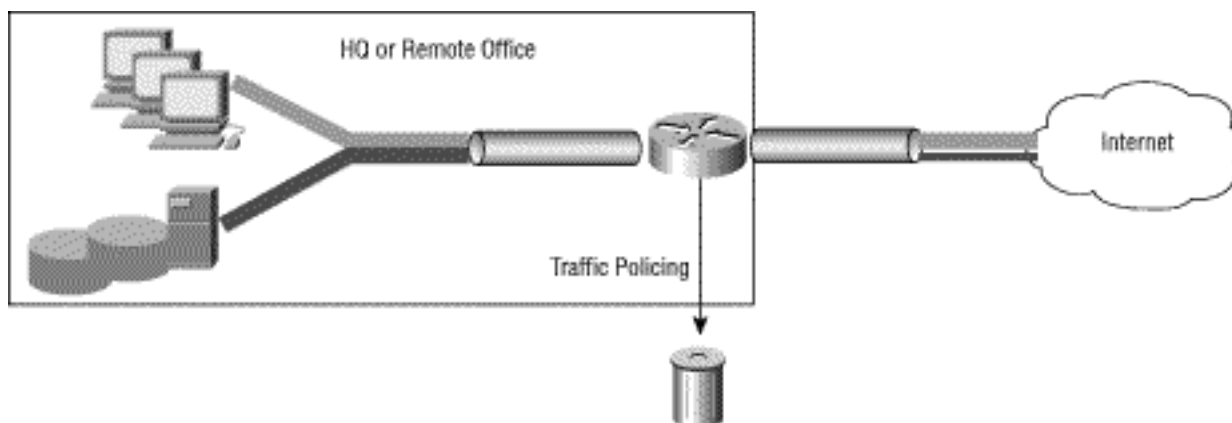
- [Согласованная скорость доступа](#)
- [Применение политики на основе классов](#)

[У двух механизмов имеются важные функциональные различия, как это указано в разделе "Сравнение политик на основе классов и согласованной скорости доступа"](#). Cisco рекомендует соблюдение политики на основе классов и другие функции модульного QoS CLI во время применения политик QoS.

Используйте команду `police`, чтобы указать максимальное значение скорости, задаваемое для определенного класса трафика, а также незамедлительное действие, выполняемое в случае превышения скорости. Иными словами, при использовании команды `police` пакеты не помещаются в буфер с последующей отправкой, как это делается в случае команды `shape`.

В дополнение к политикам, контейнер маркеров определяет, удовлетворяет ли пакет установленной скорости или превышает ее. В любом случае ограничение реализует настраиваемое действие, включающее задание приоритета IP или DSCP.

На следующем графике показано обычное применение политики трафика в точке перегрузки, где обычно применяются возможности QoS.



Сравнение минимального и максимального контроля полосы пропускания

Как команда `shape`, так и команда `police` ограничивают скорость передачи выходных данных до максимального значения кбит/с. Но главное, ни один из этих механизмов не гарантирует сохранение минимальной полосы пропускания в периоды перегрузок. **Чтобы обеспечить данные условия, используйте команду `bandwidth` или `priority`.**

Иерархическая политика использует две политики обслуживания – родительскую политику для применения механизма QoS к суммарному трафику и дочернюю политику для применения механизма QoS к потоку или к подмножеству суммарного трафика. Логические интерфейсы, например, подинтерфейсы и туннельные интерфейсы, требуют функции ограничения трафика на родительском уровне и функции создания очереди на более низких уровнях. Функция ограничения трафика снижает скорость передачи выходных данных и (предположительно) создает перегрузку, о чем свидетельствует формирование очередей излишних пакетов.

Следующая конфигурация является не совсем оптимальной и показана для иллюстрации различия между командами `police` и `shape` при ограничении суммарного трафика. В данном случае значение класса по умолчанию соответствует максимальному значению скорости. В данной конфигурации команда `police` отправляет пакеты из дочерних классов на основе размера пакета и количества байт, которые остаются в непереполненном и в переполненном контейнерах маркеров. ([См. раздел Соблюдение политики трафика.](#)) В результате невозможно гарантировать соблюдение значений скорости, установленных для классов передачи голоса через IP (VoIP) и Интернет-протокола (IP), так как функция `police` отменяет гарантии, созданные функцией `priority`.

Однако если используется команда `shape`, то результатом будет иерархическая система организации очереди, и будут выполнены все гарантии. Другими словами, когда предлагаемая нагрузка превышает нормированную скорость, классы VoIP и IP выравнивают их скорость, и все потери несет трафик класса по умолчанию (на дочернем уровне).

Внимание. : Эта конфигурация не рекомендуется и приведена лишь для того, чтобы показать разницу между командами `police` и `shape` при ограничении суммарного трафика.

```
class-map match-all IP
  match ip precedence 3
class-map match-all VoIP
  match ip precedence 5

policy-map child
  class VoIP
    priority 128
  class IP
    priority 1000

policy-map parent
  class class-default
    police 3300000 103000 103000 conform-action transmit exceed-action drop
    service-policy child
```

Чтобы обеспечить эффективность вышеприведенной конфигурации, необходимо заменить соблюдение политики функцией формирования трафика. Пример:

```
policy-map parent
  class class-default
    shape average 3300000 103000 0
    service-policy child
```

Для узнавания больше о родителе и дочерней политике см. [Дочернюю Политику обслуживания QoS для Класса приоритета.](#)

Дополнительные сведения

- [Поддержка технологии QoS](#)
- [Техническая поддержка - Cisco Systems](#)