

Отбрасывания выходных данных из-за QoS на технических примечаниях по поиску и устранению проблем уровней доступа коммутатора

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Общие сведения](#)

[Информация о функциональной возможности](#)

[Методика устранения проблем](#)

[Типичные неполадки](#)

[Вопросы и ответы](#)

[Дополнительные сведения](#)

Введение

Этот документ описывает, как устранить неполадки отбрасываний выходных данных из-за качества обслуживания (QoS) на Серии коммутаторов Cisco Catalyst 2960, 3750, 3750G, 3750X, 3560.

Предварительные условия

Требования

Cisco рекомендует иметь базовые знания о QoS.

Используемые компоненты

Сведения в этом документе основываются на этих платформах: Серия коммутаторов Cisco Catalyst 2960, 3750, 3750G, 3750X, 3560.

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.

Общие сведения

QoS используется для расположения по приоритетам более важных данных во времена перегрузки. В результате после того, как QoS включено, менее важные объемные данные могли бы испытать отбрасывания.

Уровни доступа коммутатора Cisco внедряют Характеристики QoS в аппаратных средствах. Этот документ помогает вам определять, вызваны ли отбрасывания QoS, и описывает различную организацию очереди и опции настройки буфера для смягчения их.

Информация о функциональной возможности

Входная очередь по умолчанию

Выходная очередь по умолчанию

Методика устранения проблем

1. Определите интерфейсы, которые несут исходящие данные для приложения, на которое влияют, или того опыта отбрасывания выходных данных тот инкремент. Сравните скорость выходной очереди удержания интерфейса и интерфейсную скорость и гарантируйте, что отбрасывания не происходят из-за по использованию

ССЫЛКИ. Switch#**show int gil1/0/1**

!-- Some output omitted.

GigabitEthernet0/1 is up, line protocol is up (connected)

MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

Full-duplex, **1000Mb/s**, media type is 10/100/1000BaseTX

input flow-control is off, output flow-control is unsupported

Input queue: 0/75/0/0 (size/max/drops/flushes); **Total output drops: 1089**

Queueing strategy: fifo

Output queue: 0/40 (size/max)

5 minute input rate 4000 bits/sec, 6 packets/sec

5 minute output rate 3009880 bits/sec, 963 packets/sec

2. Гарантируйте, что QoS включено на коммутаторе. Если это не включено, отбрасывания выходных данных не отнесены к QoS, и следовательно дальнейшие шаги, упомянутые

здесь, не важны. Switch#**show mls qos**

QoS is enabled

QoS ip packet dscp rewrite is enabled

3. Определите маркировку исходящего потока данных, который отброшен на интерфейсе.

Switch#**show mls qos int gil1/0/1 statistics**

GigabitEthernet1/0/1 (All statistics are in packets)

dscp: incoming

```
-----  
0 - 4 : 0 0 0 0 0  
5 - 9 : 0 0 0 0 0  
10 - 14 : 0 0 0 0 0  
15 - 19 : 0 0 0 0 0  
20 - 24 : 0 0 0 0 0  
25 - 29 : 0 0 0 0 0  
30 - 34 : 0 0 0 0 0  
35 - 39 : 0 0 0 0 0  
40 - 44 : 0 0 0 0 0  
45 - 49 : 0 198910 0 0 0  
50 - 54 : 0 0 0 0 0  
55 - 59 : 0 0 0 0 0  
60 - 64 : 0 0 0 0
```

dscp: outgoing

```
-----  
0 - 4 : 0 0 0 0 0  
5 - 9 : 0 0 0 0 0  
10 - 14 : 0 0 0 0 0  
15 - 19 : 0 0 0 0 0  
20 - 24 : 0 0 0 0 0  
25 - 29 : 0 0 0 0 0  
30 - 34 : 0 0 0 0 0  
35 - 39 : 0 0 0 0 0  
40 - 44 : 0 0 0 0 0  
45 - 49 : 0 248484 0 0 0  
50 - 54 : 0 0 0 0 0  
55 - 59 : 0 0 0 0 0  
60 - 64 : 0 0 0 0
```

cos: incoming

```
-----  
0 - 4 : 2 0 0 0 0  
5 - 7 : 0 0 0
```

cos: outgoing

```
-----  
0 - 4 : 0 0 0 0 0  
5 - 7 : 0 0 0
```

output queues enqueued:
queue: threshold1 threshold2 threshold3

```
-----  
queue 0: 248484 0 0  
queue 1: 0 0 0  
queue 2: 0 0 0  
queue 3: 0 0 0
```

output queues dropped:
queue: **threshold1** threshold2 threshold3

```
-----  
queue 0: 1089 0 0  
queue 1: 0 0 0  
queue 2: 0 0 0  
queue 3: 0 0 0
```

Policer: Inprofile: 0 OutofProfile: 0 **Примечание:** Данный пример показывает отброшенные пакеты на очереди 0/threshold1 понижающиеся пакеты. В других

примерах в документе нумерация очереди равняется 1 - 4; поэтому, это значение будет очередью 1.

4. Проверьте маркировку для вывода карты на коммутаторе для определения, который пара порога очереди сопоставляет с маркировкой, которая отброшена. В этом сценарии queue1/threshold1 сопоставлен с dscp 46, который отброшен на интерфейсе. Это означает, что трафик dscp 46 передается queue1 и отброшен, потому что у той очереди есть недостаточные буферные или меньшие циклы ЦПУ. Switch#show mls qos maps dscp-output-q

```
Dscp-outputq-threshold map:
d1 :d2 0 1 2 3 4 5 6 7 8 9
-----
0 : 02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01
1 : 02-01 02-01 02-01 02-01 02-01 02-01 03-01 03-01 03-01 03-01
2 : 03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01
3 : 03-01 03-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01
4 : 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 04-01
5 : 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01
6 : 04-01 04-01 04-01 04-01
```

5. Существует два метода для решения этих отбрасываний. Первый метод должен изменить буфер и пороговые значения для очереди, которая отбрасывает пакеты. Второй метод должен настроить планировщика так, чтобы очередь, которая отбрасывает пакеты, была обслужена чаще, чем остаток очередей.

Это шагает, показывает, как вы можете изменить буфер и порог для очередей, на которые влияют, и проверяете буфер и пороговые значения, привязанные к очереди, определенной в шаге 4. **Примечание:** Каждая очередь установила, имеет опцию для настройки размера буфера и порогового значения для этих четырех выходных очередей. Затем можно примениться, любая из очереди устанавливает в любой из портов. По умолчанию все интерфейсы используют queue-set 1 для очередей вывода, пока явно не настроено для использования queue-set 2. В этом сценарии у очереди 1 в queue-set 1 есть 25% пространства полного буфера, и порог 1 установлен к

100% Switch#show mls qos queue-set

```
Queueset: 1
Queue : 1 2 3 4
```

```
-----
buffers : 25 25 25 25
threshold1: 100 200 100 100
threshold2: 100 200 100 100
reserved : 50 50 50 50
maximum : 400 400 400 400
Queueset: 2
Queue : 1 2 3 4
```

```
-----
buffers : 25 25 25 25
threshold1: 100 200 100 100
threshold2: 100 200 100 100
reserved : 50 50 50 50
maximum : 400 400 400 400
```

6. Если вы хотите изменить буфер и пороговые значения для интерфейса, на который влияют, только, изменить queue-set 2 и настроить интерфейс, на который влияют, для использования queue-set 2. **Примечание:** Можно изменить queue-set 1 также; однако, как все интерфейсы queue-set 1 использования по умолчанию, изменение отражено ко всем интерфейсам. В данном примере изменен queue-set 2 так, чтобы очередь 1 получила 70% полного буфера. Switch(config)#mls qos queue-set output 2 buffers 70 10 10

10 В данном примере queue-set 2 и очереди изменен 1 порог. И порог 1 и порог 2 сопоставлены с 3100 так, чтобы они могли вытянуть буфер от зарезервированного пула при необходимости. Switch(config)#mls qos queue-set output 2 threshold 1 3100 3100 100 3200

7. Проверьте, что изменения отражают под корректной очередью и queue-set. Switch#show

```
mls qos queue-set
```

```
Queueset: 1
```

```
Queue : 1 2 3 4
```

```
-----  
buffers : 25 25 25 25
```

```
threshold1: 100 200 100 100
```

```
threshold2: 100 200 100 100
```

```
reserved : 50 50 50 50
```

```
maximum : 400 400 400 400
```

```
Queueset: 2
```

```
Queue : 1 2 3 4
```

```
-----  
buffers : 70 10 10 10
```

```
threshold1: 3100 100 100 100
```

```
threshold2: 3100 100 100 100
```

```
reserved : 100 50 50 50
```

```
maximum : 3200 400 400 400
```

8. Сделайте интерфейсный queue-set 2 использования, на который влияют так, чтобы изменения вошли в силу на этом интерфейсе. Switch(config)#int gi1/0/1

```
Switch(config-if)#queue-set 2
```

```
Switch(config-if)#end Проверьте, что интерфейс сопоставлен с queue-set 2. Switch#show run int gi1/0/1
```

```
interface GigabitEthernet1/0/1
```

```
switchport mode access
```

```
mls qos trust dscp
```

```
queue-set 2
```

```
end Проверьте, продолжает ли интерфейс отбрасывать пакеты.
```

9. Можно также настроить планировщика для увеличения скорости, на которой очередь 1 обслуживается с опциями формы и ресурсом общего доступа. В данном примере очередь 1 один получает 50% общих циклов ЦПУ, и другие три очереди коллективно получают 50% циклов ЦПУ. Switch(config-if)#srr-queue bandwidth share 1 75 25 5

```
Switch(config-if)#srr-queue bandwidth shape 2 0 0 0 Проверьте, продолжает ли интерфейс отбрасывать пакеты.
```

10. Включите очередь с приоритетами на этом интерфейсе. Это действие гарантирует, что весь трафик в очереди с приоритетами обработан перед любой другой очередью.

Примечание: Очередь с приоритетами обслуживается, до пустой, прежде чем другие очереди обслуживаются. По умолчанию на коммутаторах 2960/3560/3750, очередь 1 является очередью с приоритетами. Switch(config)#int gi1/0/1

```
Switch(config-if)#priority-queue out
```

```
Switch(config-if)#end Маркирование пакета, который отброшен на интерфейсе, может быть сопоставлено так, чтобы это перешло к очереди 1 (очередь с приоритетами).
```

Это действие гарантирует, что трафик с этой маркировкой всегда обрабатывается

```
перед чем-либо еще. Switch(config)#mls qos srr-queue output dscp-map queue 1 threshold 1
```

Типичные неполадки

Вот некоторые типичные проблемы:

- Отбрасывания выходных данных на интерфейсах после QoS включены.
- Вызовы прерывистого голоса.
- Добавленная задержка вызывает субоптимальный видеотрафик.
- Сброс соединения.

Вопросы и ответы

Вопрос. : Когда я использую совместное использование/формирование, когда я изменяю queue-set и?

О: Решение зависит от природы отбрасываний. Если отбрасывания инкрементно увеличиваются периодически, эта проблема происходит, скорее всего, из-за пульсирующего трафика. Наоборот, если отбрасывания инкрементно увеличиваются постоянно в постоянной скорости передачи, очередь, которая отбрасывает пакеты, скорее всего, получает больше данных, чем это может отослать.

Для неустойчивых отбрасываний у очереди должен быть большой буфер, который может принять случайные пакеты. Для реализации этого решения необходимо изменить queue-set и выделить больше буфера очереди, на которую влияют, и увеличить пороговые значения также.

Для непрерывных отбрасываний необходимо настроить планировщика, чтобы обслужить очередь, на которую влияют, чаще и вынуть больше пакетов от очереди на цикл ЦПУ. Для реализации этого решения вы должны configure совместно использование/формирование на выходных очередях.

Вопрос. : Каково различие между совместно используемым режимом и сформированным режимом?

О: В режиме профилированного трафика для исходящих очередей гарантируется процент от пропускной способности канала, и очереди ограничены этим значением. Даже если ссылка является простаивающей, имеющий форму трафик не использует больше, чем выделенная полоса. Имеющий форму режим предоставляет более ровный поток трафика в течение долгого времени и уменьшает пики и долины пульсирующего трафика. С формированием абсолютное значение каждого веса используется для вычислений пропускной способности, доступной для очередей.

srr-queue bandwidth shape weight1 weight2 weight3 weight4

Отношение обратных величин (*1/взвешивать*) средства управления пропускная способность формирования для этой очереди. Другими словами, queue1 зарезервирован $1/\text{weight1}$ процент общей пропускной способности и так далее. Если вы настраиваете вес 0, соответствующая очередь действует в совместно используемом режиме. Вес, заданный с командой **srr-queue bandwidth shape**, проигнорирован, и веса, заданные с командой **настройки интерфейса srr-queue bandwidth share** для очереди, входят в силу.

В совместно используемом режиме очереди совместно используют пропускную способность среди них на основе настроенных весов. Полоса пропускания на этом уровне гарантирована, однако им не ограничена. Например, если очередь пуста и больше не требует ресурса общего доступа ссылки, остающиеся очереди могут расшириться в неиспользуемую пропускную способность и совместно использовать ее среди них.

srr-queue bandwidth share weight1 weight2 weight3 weight4

queue1 гарантируют минимум $\text{weight1} / (\text{weight1} + \text{weight2} + \text{weight3} + \text{weight4})$ процент пропускной способности, но может также съесть в пропускную способность других несформированных очередей при необходимости.

Дополнительные сведения

- [Cisco Systems – техническая поддержка и документация](#)