

Cisco IOS 시스템 소프트웨어를 실행하는 Catalyst 6500/6000 Series 스위치의 QoS 출력 스케줄링

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[배경 정보](#)

[삭제되는 출력 대기열](#)

[Catalyst 6500/6000의 서로 다른 라인 카드의 출력 대기열 기능](#)

[포트의 대기열 기능 이해](#)

[Catalyst 6500/6000의 구성, 모니터 및 출력 스케줄링 예](#)

[구성](#)

[출력 예약 모니터링 및 구성 확인](#)

[출력 예약 예](#)

[출력 예약을 사용하여 지연 및 지터 감소](#)

[지연 감소](#)

[지터 감소](#)

[결론](#)

[관련 정보](#)

소개

출력 스케줄링을 사용하면 초과 서브스크립션이 많을 경우 중요한 트래픽이 삭제되지 않습니다. 이 문서에서는 Catalyst 6500/6000 스위치의 출력 스케줄링과 관련된 모든 기술과 알고리즘에 대해 설명합니다. 이 문서에서는 Cisco IOS® Software를 실행하는 Catalyst 6500/6000에서 출력 예약의 작업을 구성하고 확인하는 방법에 대해서도 설명합니다.

WRED(Weighted Random Early Detection), WRR(Weighted Round Robin) 및 tail 드롭에 대한 자세한 내용은 [CatOS 시스템 소프트웨어를 실행하는 Catalyst 6500/600 Series 스위치](#)의 QoS 출력 예약을 참조하십시오.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팁 표기 규칙을 참고하십시오.](#)

배경 정보

삭제되는 출력 대기열

출력 삭제는 혼잡한 인터페이스로 인해 발생합니다. 이러한 문제의 일반적인 원인은 낮은 대역폭 링크로 전환되는 높은 대역폭 링크의 트래픽 또는 단일 아웃바운드 링크로 전환되는 여러 인바운드 링크의 트래픽일 수 있습니다.

예를 들어, 많은 양의 버스트 트래픽이 기가비트 인터페이스에서 들어오고 100Mbps 인터페이스로 전환되면 100Mbps 인터페이스에서 출력이 감소할 수 있습니다. 수신 및 발신 대역폭 간의 속도 불일치로 인해 해당 인터페이스의 출력 대기열이 초과 트래픽에 의해 압도되기 때문입니다. 발신 인터페이스의 트래픽 속도는 전송해야 하는 모든 패킷을 수락할 수 없습니다.

이 문제를 해결하려면 회선 속도를 높이는 것이 가장 좋습니다. 그러나 회선 속도를 높이지 않으려는 경우 출력 삭제를 방지, 감소 또는 제어할 수 있는 방법이 있습니다. 출력 삭제가 짧은 데이터 버스트의 결과인 경우에만 이 문제를 방지할 수 있습니다. 높은 비율의 지속적인 플로우로 인해 출력 삭제가 발생하는 경우에는 이러한 삭제를 방지할 수 없습니다. 그러나 이 문제를 제어할 수는 있습니다.

Catalyst 6500/6000의 서로 다른 라인 카드의 출력 대기열 기능

포트의 큐잉 기능에 대해 잘 모르겠으면 `show queuing interface {gigabitethernet | ftthnet} mod/port` 명령다음은 `show queueing interface` 명령의 첫 번째 출력 줄입니다. 포트는 Supervisor Engine 1A 라인 카드에 있습니다.

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
```

```
QoS is disabled globally
Trust state: trust DSCP
Default COS is 0
Transmit group-buffers feature is enabled
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          WRR low           2
      2          WRR high          2
      3          Priority           1
```

!--- Output suppressed.

출력은 이 포트에 lp2q2t라는 출력 대기열 유형이 있음을 보여줍니다.

특정 포트에서 사용 가능한 대기열 유형을 확인하는 또 다른 방법은 `show interface capabilities` 명령을 실행하는 것입니다.

```

la-orion#show interface gigabitethernet 6/2 capabilities
GigabitEthernet6/2
  Model: WS-SUP720-BASE
  Type: No GBIC
  Speed: 1000
  Duplex: full
  Trunk encap. type: 802.1Q,ISL
  Trunk mode: on,off,desirable,nonegotiate
  Channel: yes
  Broadcast suppression: percentage(0-100)
  Flowcontrol: rx-(off,on,desired),tx-(off,on,desired)
  Membership: static
  Fast Start: yes
  QoS scheduling: rx-(1p1q4t), tx-(1p2q2t)
  CoS rewrite: yes
  ToS rewrite: yes
  Inline power: no
  SPAN: source/destination
  UDLD: yes
  Link Debounce: yes
  Link Debounce Time: yes
  Ports on ASIC: 1-2

```

포트의 대기열 기능 이해

Catalyst 6500/6000 스위치에는 여러 유형의 큐가 있습니다.다음 표에서는 포트 QoS 아키텍처의 표기법에 대해 설명합니다.

전송(Tx)/수신(Rx)	대기열 표기법	큐 수	우선 순위 대기열	WR R 큐 수	WRR 대기열에 대한 임계값 및 유형
Tx	2q2t	2	—	2	구성 가능한 tail 삭제 2개
Tx	1p2q2t	3	1	2	구성 가능한 WRED 2개
Tx	1p3q1t	4	1	3	구성 가능한 WRED 1개
Tx	1p2q1t	3	1	2	구성 가능한 WRED 1개
Rx	1q4t	1	—	—	구성 가능한 tail 드롭 4개
Rx	1p1q4t	2	1	1	구성 가능한 tail 드롭 4개
Rx	1p1q0t	2	1	1	구성할 수 없음
Rx	1p1q8t	2	1	1	구성 가능한 WRED 8개
Tx	1p3q8t	4	1	3	8개의 구성 가능한 WRED 또는 tail 삭제
Tx	1p7q8t	8	1	7	8개의 구성 가능한 WRED 또는 tail 삭제

Rx	1q2t	1	—	—	구성 가능한 tail drop 1개 = 구성 불가 1개
Rx	1/8분기	1	—	—	8개의 구성 가능한 tail 삭제
Rx	2q8t	2	—	2	8개의 구성 가능한 tail 삭제

다음 표에는 인터페이스 또는 포트의 Rx 및 Tx 측에 있는 일부 모듈과 대기열 유형이 나열되어 있습니다. 모듈이 여기에 나열되지 않은 경우 **show interface capabilities** 명령을 사용하여 사용 가능한 대기열 기능을 확인합니다. **show interface capabilities** 명령은 [Catalyst 6500/6000](#) 섹션의 [Output Queuing Capability of Different Line Card](#)에 설명되어 있습니다.

모듈	Rx 큐	Tx 큐
WS-X6K-S2-PFC2	1p1q4t	1p2q2t
WS-X6K-SUP1A-2GE	1p1q4t	1p2q2t
WS-X6K-SUP1-2GE	1q4t	2q2t
WS-X6501-10GEX4	1p1q8t	1p2q1t
WS-X6502-10GE	1p1q8t	1p2q1t
WS-X6516-GBIC	1p1q4t	1p2q2t
WS-X6516-GE-TX	1p1q4t	1p2q2t
WS-X6416-GBIC	1p1q4t	1p2q2t
WS-X6416-GE-MT	1p1q4t	1p2q2t
WS-X6316-GE-TX	1p1q4t	1p2q2t
WS-X6408A-GBIC	1p1q4t	1p2q2t
WS-X6408-GBIC	1q4t	2q2t
WS-X6524-100FX-MM	1p1q0t	1p3q1t
WS-X6324-100FX-SM	1q4t	2q2t
WS-X6324-100FX-MM	1q4t	2q2t
WS-X6224-100FX-MT	1q4t	2q2t
WS-X6548-RJ-21	1p1q0t	1p3q1t
WS-X6548-RJ-45	1p1q0t	1p3q1t
WS-X6348-RJ-21	1q4t	2q2t
WS-X6348-RJ21V	1q4t	2q2t
WS-X6348-RJ-45	1q4t	2q2t
WS-X6348-RJ-45V	1q4t	2q2t
WS-X6148-RJ-45V	1q4t	2q2t
WS-X6148-RJ21V	1q4t	2q2t
WS-X6248-RJ-45	1q4t	2q2t
WS-X6248A-TEL	1q4t	2q2t
WS-X6248-TEL	1q4t	2q2t
WS-X6024-10FL-MT	1q4t	2q2t

Catalyst 6500/6000의 구성, 모니터 및 출력 스케줄링 예

구성

이 섹션에서는 Cisco IOS 소프트웨어를 실행하는 Catalyst 6500/6000에서 출력 예약을 구성하는데 필요한 모든 단계를 설명합니다. Catalyst 6500/6000 기본 컨피그레이션은 [케이스 1](#)을 참조하십시오. [QoS가 활성화되고 이 문서의 기본 매개변수](#) 섹션이 사용됩니다.

Catalyst 6500/6000의 구성은 다음 5단계를 의미합니다.

1. [QoS 활성화](#)
2. [가능한 각 CoS\(Class of Service\) 값을 큐 및 임계값에 매핑\(선택 사항\)](#)
3. [WRR 가중치 구성\(선택 사항\)](#)
4. [각 대기열에 할당된 버퍼 구성\(선택 사항\)](#)
5. [각 대기열에 대한 임계값 수준 구성\(선택 사항\)](#)

주: 단계 1을 제외하고 이러한 각 단계는 선택 사항입니다. 하나 이상의 매개변수에 대한 기본값을 그대로 둘 수 있습니다.

1단계:QoS 활성화

먼저 QoS를 활성화합니다. QoS는 기본적으로 비활성화되어 있습니다. QoS가 비활성화되면 구성된 CoS 매핑이 결과에 영향을 주지 않습니다. 첫 번째 수신, 첫 번째 발신(FIFO) 방식으로 제공되는 하나의 대기열이 있으며 모든 패킷이 그곳에서 삭제됩니다.

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#mls qos
```

```
QoS is enabled globally
Microflow policing is enabled globally
```

```
QoS global counters:
Total packets: 552638
IP shortcut packets: 0
Packets dropped by policing: 0
IP packets with TOS changed by policing: 0
IP packets with COS changed by policing: 0
Non-IP packets with CoS changed by policing: 0
```

2단계:가능한 각 CoS 값을 큐 및 임계값에 매핑

모든 대기열 유형에 대해 대기열 및 임계값에 CoS를 할당합니다. 2q2t 포트 유형에 대해 정의된 매핑은 1p2q2t 포트에 적용되지 않습니다. 또한 2q2t에 대한 매핑은 2q2t 대기열 처리 메커니즘을 가진 모든 포트에 적용됩니다. 인터페이스 아래에서 **cos-map** 명령을 실행합니다.

```
wrr-queue cos-map Q_number_(1-2) threshold_number_(1-2) cos_value_1 cos_value_2
priority-queue cos-map Q_number_(always 1) cos_value_1 cos_value_2
```

참고: 이러한 각 명령은 한 줄에 있어야 합니다.

WRR 큐를 별도로 구성할 수 있습니다. 우선순위 큐가 있는 경우 **priority-queue** 명령을 사용하여 구성할 수 있습니다.

참고: 대기열은 항상 가능한 가장 낮은 우선순위 대기열부터 사용 가능한 엄격한 우선순위 대기열로 끝납니다. 예를 들면 다음과 같습니다.

- 대기열 1은 낮은 우선 순위 WRR 대기열입니다.
- 대기열 2는 우선 순위가 높은 WRR 대기열입니다.
- 대기열 3은 엄격한 우선 순위 대기열입니다.

모든 유형의 대기열에 대해 이 작업을 반복하거나, 기본 CoS 할당이 남아 있습니다. 다음은 1p2q2t 구성의 예입니다.

```
cosmos#configure terminal
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#priority-queue cos-map 1 5
!--- Assign a CoS of 5 to priority queue. cos-map configured on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 1 1 0 1
!--- Assign CoS 0 and 1 to the first threshold of low-priority WRR queue. cos-map configured on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 1 2 2 3
!--- Assign CoS 2 and 3 to the second threshold of low-priority WRR queue. cos-map configured on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 2 1 4 6
!--- Assign CoS 4 and 6 to the first threshold of high-priority WRR queue. cos-map configured on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 2 2 7
!--- Assign CoS 7 to the first threshold of high-priority WRR queue. cos-map configured on: Gi1/1 Gi1/2
```

구성을 확인합니다.

```
cosmos#show queueing interface gigabitethernet 1/1
!--- Output suppressed. queue thresh cos-map ----- 1 1 0 1 1 2
2 3 2 1 4 6 2 2 7 3 1 5 !--- Output suppressed.
```

3단계:WRR 가중치 구성

두 WRR 대기열에 대한 WRR 가중치를 구성합니다. 이 interface 명령을 실행합니다.

```
wrr-queue bandwidth weight_for_Q1 weight_for_Q2
```

가중치 1은 우선순위 낮은 WRR 대기열이어야 하는 대기열 1과 관련이 있습니다. 항상 이 가중치를 가중치 2보다 한 레벨 낮게 유지합니다. 가중치는 1에서 255 사이의 값을 사용할 수 있습니다. 다음 공식을 사용하여 퍼센트를 지정합니다.

- 대기열에서 1— $[가중치\ 1 / (가중치\ 1 + 가중치\ 2)]$
- 대기열에서 2— $[가중치\ 2 / (가중치\ 1 + 가중치\ 2)]$

모든 유형의 대기열에 대한 가중치를 정의해야 합니다. 이러한 가중치 유형은 같을 필요가 없습니다. 이것은 2q2t의 예입니다. 여기서 대기열 1은 시간의 20%, 대기열 2는 시간의 80%를 제공합니다.

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue bandwidth ?
<1-255> enter bandwidth weight between 1 and 255
```

```
cosmos(config-if)#wrr-queue bandwidth 20 80
!--- Queue 1 is served 20% of the time, and queue 2 is served !--- 80% of the time.
```

```
cosmos(config-if)#
구성을 확인합니다.
```

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
     1          WRR low           2
     2          WRR high           2
     3          Priority           1

  WRR bandwidth ratios:  20[queue 1]  80[queue 2]
  queue-limit ratios:    90[queue 1]  5[queue 2]
```

!--- Output suppressed.

참고: CatOS 소프트웨어를 사용할 수 없는 경우 각 인터페이스에 대해 서로 다른 WRR 가중치를 구성할 수 있습니다.

4단계:각 대기열에 할당된 버퍼 구성

전송 대기열 비율을 정의해야 합니다.이렇게 하면 버퍼가 서로 다른 큐 간에 분할되는 방법이 결정됩니다.

```
wrr-queue queue-limit percentage_WRR_Q1 percentage_WRR_Q2
cosmos(config)#interface gigabitethernet 1/2
cosmos(config-if)#wrr-queue queue-limit 70 15
!--- Queue 1 has 70% of the buffers. !--- Queues 2 and 3 both have 15% of the buffers. queue-
limit configured on: Gi1/1 Gi1/2
```

참고: 기가비트 포트의 큐잉 기능이 1p1q2t인 경우 엄격한 우선 순위 대기열과 높은 우선 순위 WRR 대기열에 동일한 레벨을 사용해야 합니다.이러한 레벨은 하드웨어 이유로 다를 수 없습니다.두 WRR 대기열의 대역폭만 구성됩니다.우선순위가 높은 WRR 대기열과 엄격한 우선순위 대기열이 있는 경우 동일한 값을 자동으로 사용합니다.

일부 큐 유형에 조정 가능한 큐 크기가 없습니다.예를 들어 WS-X6548RJ45에서 사용할 수 있는 1p3q1t가 있습니다. 이러한 큐 유형은 고정되어 있으며 수정할 수 없습니다.

구성을 확인합니다.

```
cosmos#show queueing interface gigabitethernet 1/2
Interface GigabitEthernet1/2 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
     1          WRR low           2
     2          WRR high           2
     3          Priority           1
```

```
WRR bandwidth ratios:      5[queue 1] 255[queue 2]
queue-limit ratios:       70[queue 1] 15[queue 2]
```

참고: 우선 순위가 낮은 WRR 대기열의 버퍼에서 가장 큰 부분을 유지하는 것이 좋습니다. 추가 버퍼링을 활성화해야 하는 대기열입니다. 다른 대기열은 우선 순위가 더 높습니다.

5단계: 각 대기열에 대한 임계값 레벨 구성

마지막 단계로 WRED 대기열 또는 테일 드롭 대기열에 대한 임계값 레벨을 구성합니다. 이 목록은 다음 명령을 제공합니다.

- WRED를 임계값에 대한 삭제 메커니즘으로 사용하는 대기열의 경우 다음 명령을 실행합니다.

```
wrr-queue random-detect min-threshold Q_number threshold_1_value threshold_2_value
wrr-queue random-detect max-threshold Q_number threshold_1_value threshold_2_value
```

참고: 이러한 각 명령은 한 줄에 있어야 합니다.

- 삭제 메커니즘으로 tail drop을 사용하는 대기열의 경우 다음 명령을 실행합니다.

```
wrr-queue threshold Q_number threshold_1_value threshold_2_value
```

참고: 이 명령은 한 줄에 있어야 합니다.

WRED 대기열에 대한 구성:

```
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue random-detect min-threshold 1 20 50
!--- This sets the threshold of queue 1 to 20 and 50% minimum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect min-threshold 2 20 50
!--- This sets the threshold of queue 2 to 20 and 50% minimum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 1 50 80
!--- This sets the threshold of queue 1 to 50 and 80% maximum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 2 40 60
!--- This sets the threshold of queue 2 to 49 and 60% maximum threshold !--- configured on Gi1/1
Gi1/2.
```

비상 삭제 대기열에 대한 컨피그레이션:

```
cosmos(config)#interface fastethernet 3/1
cosmos(config-if)#wrr-queue threshold ?
<1-2> enter threshold queue id (1-2)
cosmos(config-if)#wrr-queue threshold 1 ?
<1-100> enter percent of queue size between 1 and 100
cosmos(config-if)#wrr-queue threshold 1 50 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 1 (low-priority)
to 50 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)# cosmos(config-if)# cosmos(config-if)#wrr-
queue threshold 2 40 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 2 (high-priority)
to 40 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)#
```

구성을 확인합니다.

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
```

```

Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = lp2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low     2
2             WRR high    2
3             Priority    1

WRR bandwidth ratios:  20[queue 1]  80[queue 2]
queue-limit ratios:    70[queue 1]  15[queue 2]

```

```
queue random-detect-min-thresholds
```

```
-----
1    20[1] 50[2]
2    20[1] 50[2]
```

```
queue random-detect-max-thresholds
```

```
-----
1    50[1] 80[2]
2    40[1] 60[2]
```

```
cosmos#show queuing interface fastethernet 3/1
```

```
Interface FastEthernet3/1 queuing strategy:  Weighted Round-Robin
```

```

Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low     2
2             WRR high    2

WRR bandwidth ratios:  100[queue 1] 255[queue 2]
queue-limit ratios:    90[queue 1]  10[queue 2]

```

```
queue tail-drop-thresholds
```

```
-----
1    50[1] 100[2]
2    40[1] 100[2]
```

임계값을 구성하고 포트당 대기열에 CoS를 할당할 수 없습니다. 모든 변경 사항은 연속된 포트 집합에 적용됩니다.

- 기가비트 라인 카드용 포트 4개 - 포트 1~4가 함께 있고 포트 5~8이 함께 있습니다.
- 10/100 포트용 포트 12개 또는 1q4t/2q2t 대기열 처리를 기반으로 하는 100 파이버 포트 12개 (1~12, 13~24, 25~36, 36~48).
- 동일한 ASIC에 속하는 정확한 포트를 확인하려면 **show interface capabilities** 명령을 사용합니다.

출력 예약 모니터링 및 구성 확인

출력 예약과 관련하여 포트의 현재 런타임 구성을 확인하기 위해 실행하는 가장 쉬운 명령은 **show queuing interface {gigabitethernet | fthethernet} slot/port** 명령입니다. 명령은 포트의 대기열 유형, 다른 대기열 및 임계값에 CoS 매핑, 버퍼 공유 및 WRR 가중치를 표시합니다. 여기서는 대기열 1의 20% WRR과 대기열 2의 80% WRR입니다. 이 명령은 출력 예약에 대해 구성된 모든 정보와 각 임계값에 대해 각 대기열에서 삭제된 패킷 수를 표시합니다.

cosmos#show queueing interface gigabitethernet 1/1

Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin

Port QoS is enabled

Port is untrusted

Default COS is 0

Transmit queues [type = lp2q2t]:

Queue Id	Scheduling	Num of thresholds
----------	------------	-------------------

1	WRR low	2
2	WRR high	2
3	Priority	1

WRR bandwidth ratios: 20[queue 1] 80[queue 2]

queue-limit ratios: 70[queue 1] 15[queue 2]

queue random-detect-max-thresholds

1	50[1]	80[2]
2	40[1]	60[2]

queue thresh cos-map

1	1	0	1
1	2	2	3
2	1	4	6
2	2	7	
3	1	5	

Receive queues [type = lp1q4t]:

Queue Id	Scheduling	Num of thresholds
----------	------------	-------------------

1	Standard	4
2	Priority	1

queue tail-drop-thresholds

1	100[1]	100[2]	100[3]	100[4]
---	--------	--------	--------	--------

queue thresh cos-map

1	1	0	1
1	2	2	3
1	3	4	
1	4	6	7
2	1	5	

Packets dropped on Transmit:

BPDU packets: 0

queue thresh	dropped	[cos-map]
--------------	---------	-----------

1	1	0 [0 1]
1	2	0 [2 3]
2	1	0 [4 6]
2	2	0 [7]
3	1	0 [5]

Packets dropped on Receive:

BPDU packets: 0

queue thresh	dropped	[cos-map]
--------------	---------	-----------

```

-----
1      1          0 [0 1 ]
1      2          0 [2 3 ]
1      3          0 [4 ]
1      4          0 [6 7 ]
2      1          0 [5 ]

```

출력 예

이 트래픽은 Catalyst 6500/6000에 삽입됩니다.

- 포트 기가비트 1/2:1기가비트 트래픽의 우선 순위가 0입니다.
- 포트 기가비트 5/2:133MB의 트래픽(우선 순위 7개)133MB의 트래픽(우선 순위 6개)133MB의 트래픽(우선 순위 5개)133MB의 트래픽(우선 순위 4개)133MB의 트래픽(우선 순위 3개)133MB의 트래픽(우선 순위 2개)133MB의 트래픽, 우선 순위 1개

모든 유니캐스트 트래픽은 매우 초과 가입된 포트 기가비트 1/1당 스위치를 종료합니다.

사례 1:QoS가 활성화되고 기본 매개변수가 사용됩니다.

show queuing interface gigabitethernet 1/1 명령은 이 예에서 모든 출력을 구성합니다.이 명령은 입력 예약에 대한 추가 정보를 제공합니다.그러나 이 문서에서는 출력 예약만 다루므로 해당 출력이 억제됩니다.

QoS가 전역적으로 활성화되고 모든 기본 매개변수가 사용 중인 경우 이 출력은 몇 분 후에 발생합니다.

```

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          WRR low      2
      2          WRR high     2
      3          Priority     1

WRR bandwidth ratios:  100[queue 1] 255[queue 2]
queue-limit ratios:    90[queue 1]  5[queue 2]

queue random-detect-max-thresholds
-----
      1      40[1] 100[2]
      2      40[1] 100[2]

queue thresh cos-map
-----
      1      1      0 1
      1      2      2 3
      2      1      4
      2      2      6 7
      3      1      5

Packets dropped on Transmit:
BPDU packets:  0

queue thresh      dropped  [cos-map]

```

```
-----
1      1      149606424  [0 1 ]
1      2              0 [2 3 ]
2      1      16551394  [4 ]
2      2      4254446   [6 7 ]
3      1              0 [5 ]
```

이 출력에서 기본값은 다음과 같습니다.

- 대기열 1 - $100 / (100 + 255) = 28\%$
- 대기열 2 - $255 / (255 + 100) = 72\%$
- 버퍼 공유:—대기열 1, 대기열 2 5%, 엄격한 우선 순위 대기열 5%

우선 순위가 낮은 WRR 대기열의 대부분의 패킷은 삭제되지만, 두 임계값 모두에 대해 우선순위가 높은 WRR 대기열에서 일부 패킷은 여전히 삭제됩니다. 총 170,412,264개(149,606,424 + 16,551,394 + 4,254,446)가 있습니다. 이러한 삭제는 다음과 같이 분할됩니다.

- $149,606,424 / 170,412,264 =$ 대기열 1에 있는 삭제 중 88%(CoS 0 및 1이 있는 첫 번째 임계값 패킷)
- $16,551,394 / 170,412,264 =$ 대기열 2에서 10% 삭제(CoS 4의 첫 번째 임계값 패킷)
- $4,254,446 / 170,412,264 =$ 대기열 2에서 삭제(CoS가 6 또는 7인 두 번째 임계값 패킷) 2%

참고: strict-priority 대기열에는 어떤 드롭도 표시되지 않습니다.

사례 2:WRR 가중치 수정

[사례 1](#)에 나와 있는 대로:[QoS가 활성화되고 기본 매개변수가 사용됨](#) 섹션, 대기열 2의 패킷은 여전히 삭제됩니다.대기열 2에 더 많은 대역폭을 제공하도록 WRR 가중치를 수정합니다. 이제 대기열 1은 시간의 4%를 비우고 대기열 2는 시간의 96%를 비웁니다.

```
show run interface gigabitethernet 1/1
```

```
interface GigabitEthernet1/1
no ip address
wrr-queue bandwidth 10 255
mls qos trust dscp
switchport
switchport mode access
end
```

```
nelix#show queueing interface gigabitethernet 1/1
```

```
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = 1p2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
1             WRR low    2
2             WRR high   2
3             Priority   1

WRR bandwidth ratios:  10[queue 1] 255[queue 2]
queue-limit ratios:   90[queue 1]  5[queue 2]

queue random-detect-max-thresholds
-----
1      40[1] 100[2]
2      40[1] 100[2]
```

```

queue thresh cos-map
-----
1      1      0 1
1      2      2 3
2      1      4
2      2      6 7
3      1      5

```

Packets dropped on Transmit:
 BPDU packets: 0

```

queue thresh      dropped [cos-map]
-----
1      1          2786205 [0 1 ]
1      2              0 [2 3 ]
2      1          11363 [4 ]
2      2              69 [6 7 ]
3      1              0 [5 ]

```

이 출력에서 볼 수 있듯이 대기열 2의 삭제 비율은 이제 훨씬 낮습니다. 총 2,797,637개의 삭제는 다음과 같이 분할됩니다.

- $2,786,205 / 2,797,637 =$ 대기열 1에서 99.591% 삭제(CoS 0 및 1 패킷 포함)
- $11,363 / 2,797,637 =$ 대기열 2에서 0.408% 삭제(패킷 CoS 4의 첫 번째 임계값)
- $69 / 2,797,637 =$ 대기열 2에서 0.001% 삭제(CoS 6 및 7의 패킷에 대한 두 번째 임계값)

다양한 WRR 가중치를 사용하는 경우 대기열 2에서 더 많은 QoS를 보장합니다.

사례 3: 추가 WRR 가중치 수정

WRR 무게로 더욱 공격적일 수 있습니다. 이 샘플 출력에서는 가중치의 0.39%만 대기열 1에 주어집니다.

```

show run interface gigabitethernet 1/1
interface GigabitEthernet1/1
no ip address
wrr-queue bandwidth 1 255
mls qos trust dscp
switchport
switchport mode access
end

```

```

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = lp2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          WRR low           2
      2          WRR high           2
      3          Priority           1

WRR bandwidth ratios:      1[queue 1] 255[queue 2]
queue-limit ratios:      90[queue 1] 5[queue 2]

queue random-detect-max-thresholds

```

```

-----
 1    40[1] 100[2]
 2    40[1] 100[2]

queue thresh cos-map
-----
 1     1     0 1
 1     2     2 3
 2     1     4
 2     2     6 7
 3     1     5

```

Packets dropped on Transmit:
BPDU packets: 0

```

queue thresh    dropped  [cos-map]
-----
 1     1          2535315  [0 1 ]
 1     2              0  [2 3 ]
 2     1             705  [4 ]
 2     2             73  [6 7 ]
 3     1              0  [5 ]

```

공격적인 WRR 가중치에서도 패킷은 여전히 대기열 2에서 삭제됩니다. 그러나 이에 비해 패킷은 많지 않습니다. 이제 큐 2에는 0.03% 패킷 삭제만 있습니다.

사례 4: 큐 제한 버퍼 할당 수정

[사례 2](#)에서 볼 수 있듯이 WRR 무게 및 [케이스 3 수정: 추가 WRR Weight Modification\(WRR 가중치 수정\)](#) 섹션은 WRR 백분율로 인해 드롭이 최소로 간주되지만 패킷은 여전히 대기열 2에 드롭됩니다. 그러나 대기열 2에서 두 번째 임계값(100%로 설정)에 도달하면 일부 패킷은 계속 삭제됩니다.

이를 개선하기 위해 큐 제한(각 대기열에 할당된 버퍼의 크기)을 변경합니다. 이 예에서는 대기열 제한이 대기열 1의 경우 70%, 대기열 2의 경우 15%, 엄격한 우선순위 대기열의 경우 15%로 설정됩니다.

```

show run gigabitethernet 1/1
interface GigabitEthernet1/1
no ip address
wrr-queue bandwidth 1 255
wrr-queue queue-limit 70 15
mls qos trust dscp
switchport
switchport mode access
end

```

```

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0
Transmit queues [type = 1p2q2t]:
Queue Id    Scheduling  Num of thresholds
-----
 1          WRR low     2
 2          WRR high   2
 3          Priority   1

WRR bandwidth ratios:    1[queue 1] 255[queue 2]

```

```
queue-limit ratios:      70[queue 1] 15[queue 2]
```

```
queue random-detect-max-thresholds
```

```
-----  
1    40[1] 100[2]  
2    40[1] 100[2]
```

```
queue thresh cos-map
```

```
-----  
1    1    0 1  
1    2    2 3  
2    1    4  
2    2    6 7  
3    1    5
```

```
Receive queues [type = 1plq4t]:
```

```
Queue Id    Scheduling  Num of thresholds  
-----  
1           Standard    4  
2           Priority    1
```

```
queue tail-drop-thresholds
```

```
-----  
1    100[1] 100[2] 100[3] 100[4]
```

```
queue thresh cos-map
```

```
-----  
1    1    0 1  
1    2    2 3  
1    3    4  
1    4    6 7  
2    1    5
```

```
Packets dropped on Transmit:
```

```
BPDU packets: 0
```

```
queue thresh    dropped  [cos-map]
```

```
-----  
1    1    154253046  [0 1 ]  
1    2           0  [2 3 ]  
2    1           0  [4 ]  
2    2           0  [6 7 ]  
3    1           0  [5 ]
```

이제 삭제는 대기열 1에서만 발생합니다.

출력 예약을 사용하여 지연 및 지터 감소

Example of Output Scheduling 섹션의 사례는 출력 포트의 오버서브스크립션 시 VoIP 또는 미션 크리티컬 트래픽이 손실되지 않도록 출력 스케줄링을 구현하여 얻을 수 있는 이점을 보여줍니다. 오버서브스크립션은 일반 네트워크(특히 기가비트 링크에서)에서 자주 발생하지 않습니다. 초과 서브스크립션은 트래픽의 피크 시간 또는 매우 짧은 시간 내에 발생하는 트래픽 버스트 동안에만 발생해야 합니다.

초과 서브스크립션이 없더라도 QoS가 엔드 투 엔드 구현되는 네트워크에서 출력 스케줄링이 큰 도움이 될 수 있습니다. 이 섹션에서는 출력 스케줄링을 통해 지연 및 지터를 줄이는 방법에 대한 예를 제공합니다.

지연 감소

패킷이 전송되기를 기다리는 동안 각 스위치의 버퍼에서 "lost" 시간이 발생하여 패킷의 지연이 증가합니다. 예를 들어 CoS가 5인 작은 음성 패킷은 대규모 백업 또는 파일 전송 중에 포트에서 전송됩니다. 출력 포트에 대한 QoS가 없으며 100바이트의 큰 패킷 1500바이트 후 소규모 음성 패킷이 대기됩니다. 이 경우 10개의 큰 패킷을 전송하는 데 필요한 기가비트 속도 시간을 쉽게 계산할 수 있습니다.

- $(10 \times 1500 \times 8) = 120,000$ 비트가 120마이크로초 단위로 전송됨

네트워크를 통과하는 동안 이 패킷이 8 또는 9개의 스위치를 통과해야 하는 경우 약 1밀리초 지연이 발생할 수 있습니다. 여기에는 네트워크에 연결된 스위치의 출력 대기열에서 지연만 포함됩니다.

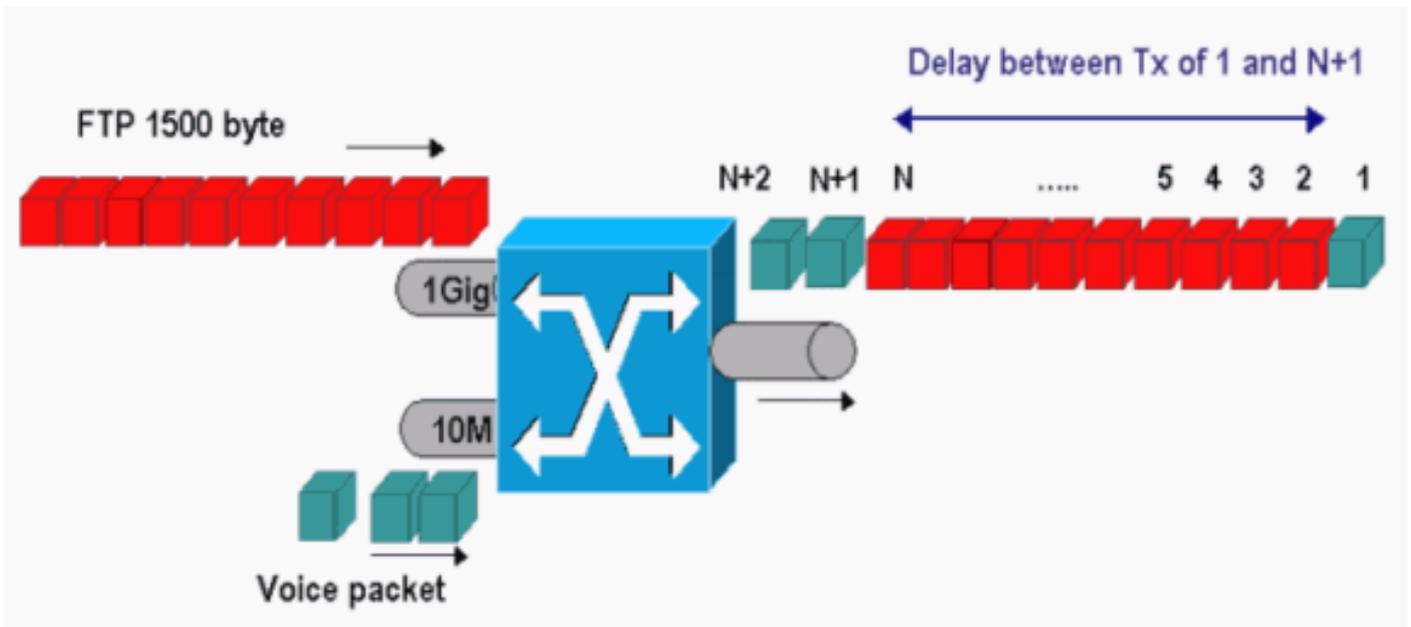
참고: 10MB 인터페이스에서 동일한 10개의 대용량 패킷을 대기해야 하는 경우(예: IP 전화 및 PC에 연결됨) 지연이 발생합니다.

- $(10 \times 1500 \times 8) = 120,000$ 비트 전송(12밀리초)

출력 스케줄링을 구현하면 CoS가 5인 음성 패킷이 엄격한 우선순위 대기열에 배치되고 CoS가 5보다 작은 패킷보다 먼저 전송되도록 할 수 있습니다. 이렇게 하면 지연이 줄어듭니다.

지터 감소

출력 스케줄링의 또 다른 중요한 이점은 지터 감소입니다. 지터는 동일한 흐름 내의 패킷에 대한 지연 변수입니다. 이 예제 시나리오에서는 출력 스케줄링이 지터를 줄이는 방법을 보여줍니다.



이 시나리오에서는 동일한 출력 포트가 두 개의 스트림을 전송해야 합니다.

- 10MB 이더넷 포트에 들어오는 음성 스트림 1개
- 1기가비트 이더넷 업링크 포트에서 수신 FTP 스트림 1개

두 스트림 모두 동일한 출력 포트를 통해 스위치를 종료합니다. 이 예에서는 출력 예약을 사용하지 않고 발생할 수 있는 상황을 보여 줍니다. 모든 대용량 데이터 패킷은 두 음성 패킷 간에 인터리빙될 수 있습니다. 이렇게 하면 동일한 스트림에서 음성 패킷의 수신에 지터가 생성됩니다. 스위치가 대용량 데이터 패킷을 전송하므로 패킷 1과 패킷 $n+1$ 의 수신 사이에 더 큰 지연이 있습니다. 그러나 $n+1$ 과 $n+2$ 사이의 지연은 무시해도 됩니다. 이로 인해 음성 트래픽 스트림에서 잡음이 발생합니다. 엄격한 우선 순위 큐를 사용하면 이 문제를 쉽게 방지할 수 있습니다. 음성 패킷의 CoS 값을 엄격한 우선 순위 대기열에 매핑해야 합니다.

결론

이 문서에서는 Cisco IOS Software를 실행하는 Catalyst 6500/6000에서 출력 대기열 예약을 구성하고 문제를 해결하는 방법에 대한 사례 연구를 살펴보았습니다. 음성 트래픽이 있는 대부분의 네트워크에서 출력 스케줄링의 장점도 살펴보았습니다.

- 출력 포트의 오버서브스크립션 시 중요한 트래픽의 삭제를 방지합니다.
- 지연을 줄입니다.
- 지터를 줄입니다.

관련 정보

- [CatOS 시스템 소프트웨어를 실행하는 Catalyst 6500/6000 Series 스위치의 QoS 출력 스케줄링](#)
- [Catalyst 6000 제품군 스위치의 서비스 품질 이해](#)
- [LAN 제품 지원 페이지](#)
- [LAN 스위칭 지원 페이지](#)
- [기술 지원 및 문서 - Cisco Systems](#)