트래픽 정책 및 트래픽 형태를 비교하여 대역폭 제한

목차

소개

이 문서에서는 출력 속도를 제한하는 트래픽 셰이핑과 트래픽 폴리싱의 기능적 차이에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참고하십시오.

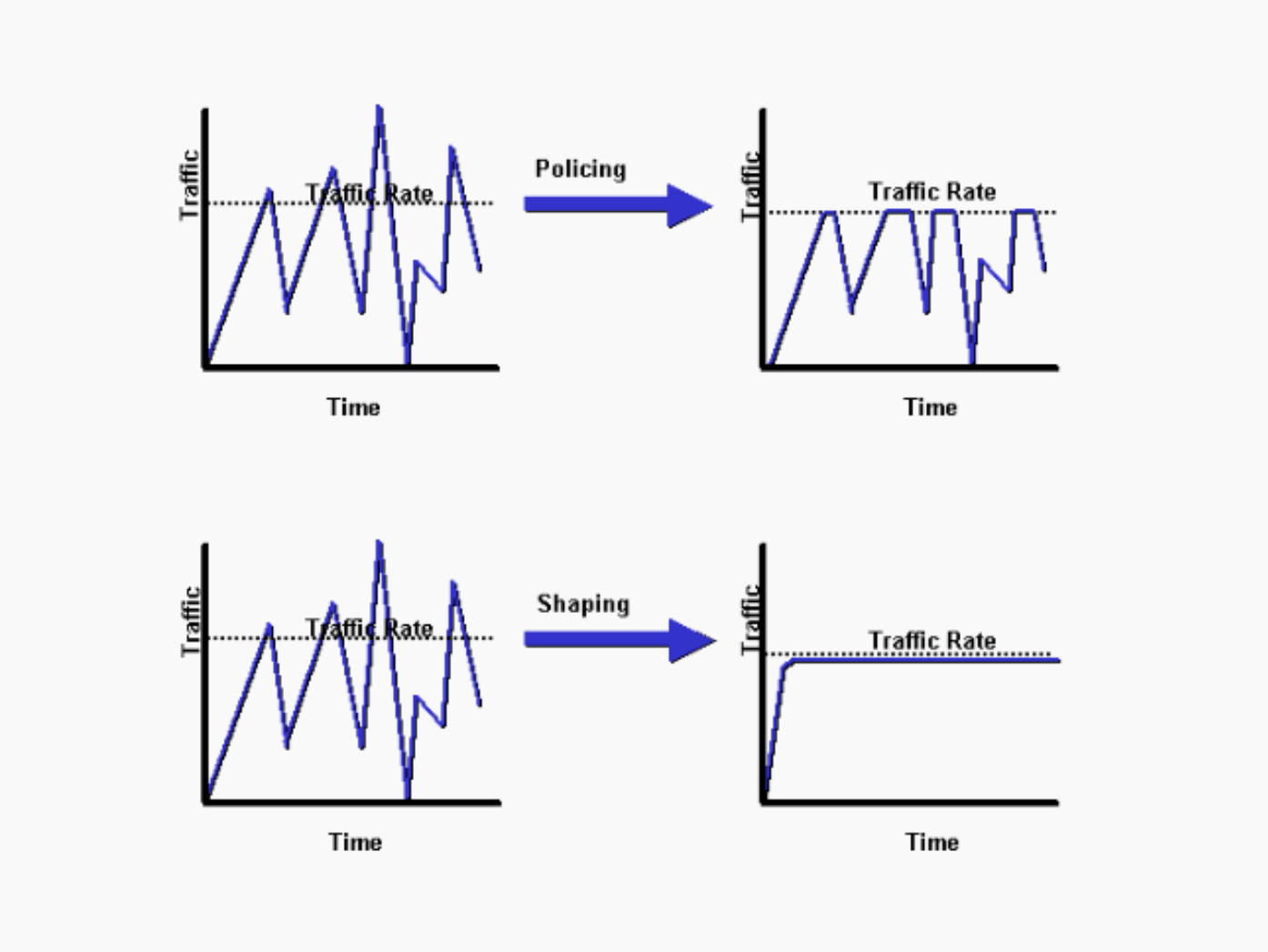
배경 정보

이 문서에서는 트래픽 셰이핑과 폴리싱의 기능적 차이를 설명합니다. 두 기능 모두 트래픽 출력 속도를 제한합니다. 두 메커니즘 모두 토큰 버킷을 트래픽 측정기로 사용하여 패킷 속도를 측정합니다. 토큰 버킷에 대한 자세한 내용은 토큰 버킷이란 무엇인지를 참조하십시오

정책 대 형성

트래픽 폴리싱은 버스트를 전파합니다. 트래픽 속도가 설정된 최대 속도에 도달하면 초과 트래픽은 삭제(또는 다시 마킹)됩니다. 그 결과, 출력 속도는 톱니 모양으로 표시됩니다. 폴리싱과 달리 트래픽 셰이핑은 초과 패킷을 대기열에 유지한 다음, 시간 증가에 따라 나중에 전송할 수 있도록 초과 패킷을 예약합니다. 트래픽 쉐이핑의 결과는 평활화된 패킷 출력 속도입니다.

다음 다이어그램은 두 트래픽 옵션의 주요 차이점을 보여줍니다.



정책 VS 형성

셰이핑은 큐가 있고 지연된 패킷을 버퍼링할 충분한 메모리가 있음을 의미합니다. 그러나 폴리싱은 그렇지 않습니다. 큐는 아웃바운드 개념입니다. 인터페이스를 떠나는 패킷은 큐에 추가되며 셰이핑될 수 있습니다. 인터페이스의 인바운드 트래픽에는 폴리싱만 적용할 수 있습니다. 셰이핑을 활성화할 때 충분한 메모리가 있는지 확인합니다. 또한 셰이핑은 지연된 패킷의 나중에 전송을 예약하

는 기능을 필요로 합니다. 이 예약 기능을 사용하면 셰이핑 대기열을 다른 대기열로 구성할 수 있습니다. 이 기능의 예로는 CBWFQ(Class Based Weighted Fair) Queuing 및 LLQ(Low Latency) Queuing 가 있습니다.

선택 기준

다음 표에는 적절한 트래픽 솔루션을 선택하는 데 도움이 되는 쉐이핑과 폴리싱의 차이점이 나열되어 있습니다.

	셰이핑	치안
목표	커밋된 속도 이상의 초과 패킷을 버퍼하고 대기시킵니다.	커밋된 속도를 초과하는 초과 패킷을 삭제(또는 설명)합니다. 버퍼링하지 않습니다.*
토큰 새로 고침 빈도	시간 간격의 시작 시 증가합니다. 최소 간격 수가 필요합니다.	공식 기준 연속:**1/커밋된 정보 전송률**
토큰 값	초당 비트 수로 구성됩니다.	바이트 단위로 구성됩니다.
컨피그레이션 옵션	• 클래스 기반 셰이핑을 구현하기 위한 모듈식 MQC(Quality of Service Command-Line Interface)의 셰이프 명령입니다. • FRTS(Frame Relay Traffic Shaping)를 구현하기 위한 Frame-relay traffic-shape 명령입니다. • GTS(Generic Traffic Shaping)를 구현하기 위한 traffic-shape 명령입니다.	• MQC의 경찰 지휘부가 클래스 기반 치안을 구현합니다. • CAR(Committed Access Rate)을 구현하기 위한 rate-limit 명령입니다.
인바운드에 적용	아니요	예
아웃바운드에 적용 가능	예	예
버스트	최소 8개 시간 간격으로 버스트 및 출력 속도를 제어합니다. 트래픽을 지연시키는 데 leaky 버킷을 사용하여 매끄러움 효과를 얻습니다.	버스트 전파. 매끄럽게 하지 않습니다.
장점	초과 패킷이 버퍼링되므로 초과 패킷을 삭제할 가능성이 적습니다. (큐 길이까지 패킷을 버퍼링합니다. 초과 트래픽이 높은 속도로 지속되는 경우 드롭이 발생할 수 있습니다.) 일반적으로 삭제된 패킷으로 인한 재전송을 방지합니다.	패킷 삭제를 통해 출력 속도를 제어합니다. 이로 인한 지연 queuing방지

단점	특히 대기열이 깊기 때문에 queuing지연이 발생할 수 있습니다.	과도한 패킷(구성된 경우)을 삭제하고, TCP 창 크기를 줄이며, 영향을 받는 트래픽 스트림의 전체 출력률을 줄입니다. 버스트 크기가 지나치게 크면 과도한 패킷이 드롭되고 특히 TCP 기반 플로우의 경우 전반적인 출력 속도가 저하될 수 있습니다.
선택적 패킷 리마킹	아니요	예(레거시 CAR 기능 포함).

* 폴리싱은 버퍼를 적용하지 않지만 구성된 메커니즘은 물리적 인터페이스에서 직렬화를 기다리는 동안 대기해야 하는 정합성이 보장된 패킷에 queuing 적용됩니다.

토큰 새로 고침 빈도

셰이핑과 폴리싱의 주요 차이점은 토큰이 보충되는 속도입니다. 쉐이핑과 폴리싱 모두 토큰 버킷 은유를 사용합니다. 토큰 버킷 자체에는 폐기 또는 우선순위 정책이 없습니다.

토큰 버킷 기능:

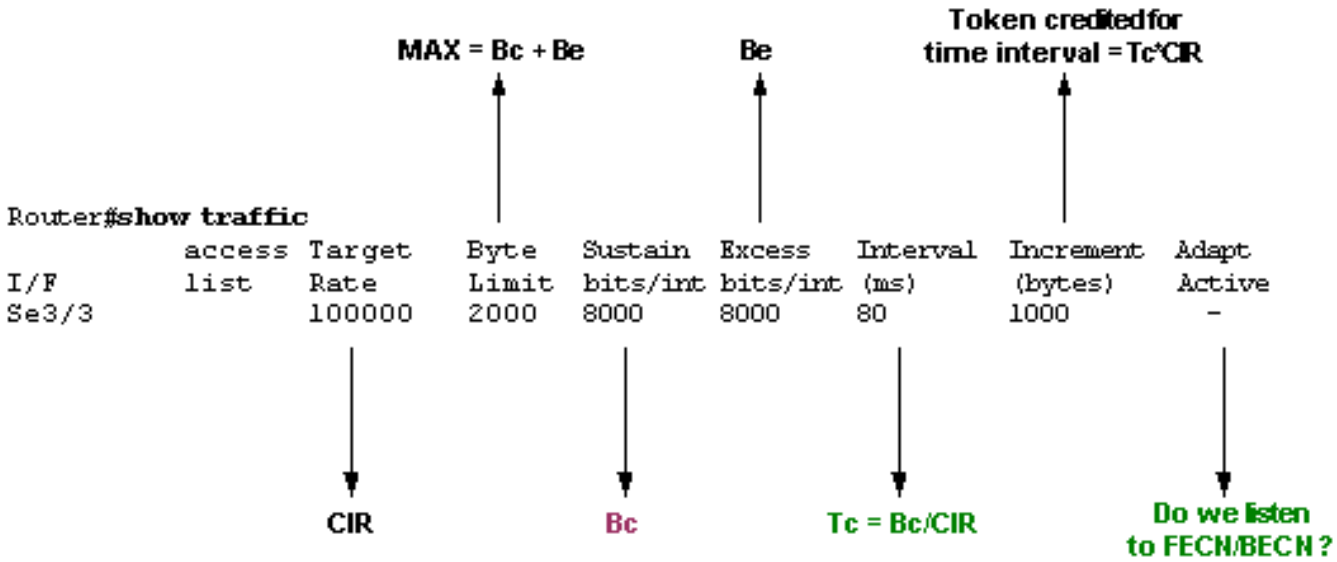
- 토큰이 일정한 속도로 버킷에 담깁니다.
- 각 토큰은 소스에서 네트워크로 특정 비트 수를 보낼 수 있는 권한을 부여합니다.
- 패킷 하나를 전송하려면 트래픽 조절기는 해당 패킷 크기에 대해 표시된 것과 동등한 수의 토큰을 버킷에서 제거할 수 있어야 합니다.
- 버킷에 패킷을 전송할 수 있는 충분한 토큰이 없는 경우, 패킷은 버킷에 충분한 토큰이 있을 때까지 대기하거나(쉐이퍼의 경우), 패킷이 폐기되거나 아래로 표시됩니다(폴리서의 경우).
- 버킷 자체에는 지정된 용량이 있습니다. 버킷의 용량이 가득 차면 도착하는 새 토큰은 폐기되며 이후 패킷에서는 사용할 수 없습니다. 따라서 언제든지 소스가 네트워크로 전송할 수 있는 최대 버스트는 버킷의 크기와 거의 비례합니다. 토큰 버킷은 재물을 허용하지만 그것을 제한한다.

셰이핑은 bps(bits per second) 값을 사용하는 시간 간격에 따라 토큰 버킷을 증가시킵니다. 쉐이퍼는 다음 공식을 사용합니다.

```
Tc = Bc/CIR (in seconds)
```

이 식에서 Bc는 커밋된 버스트이고, CIR은 커밋된 정보율을 의미한다. 자세한 내용은 프레임 릴레이 트래픽 셰이핑 구성을 참조하십시오. Tc 값은 CIR의 평균 속도(초)를 유지하기 위해 Bc 비트를 전송하는 시간 간격을 정의합니다.

Tc의 범위는 10ms에서 125ms 사이입니다. Cisco 7500 Series의 DTS(Distributed Traffic Shaping)에서는 최소 Tc가 4ms입니다. 라우터는 내부적으로 CIR 및 Bc 값을 기반으로 이 값을 계산합니다. Bc/CIR이 125 ms 미만일 경우 해당 식에서 계산된 Tc를 사용한다. Bc/CIR이 125ms 이상인 경우 Cisco IOS®에서 트래픽 흐름이 더 작은 간격으로 더 안정적일 수 있다고 판단할 경우 내부 Tc 값을 사용합니다. 라우터에서 **Tc에** 대한 내부 값을 사용할지 또는 명령줄에서 구성한 값을 사용할지를 확인하려면 show traffic-shape 명령을 사용합니다. show traffic-shape 명령의 다음 샘플 **출력**에 대해서는 Show Commands for Frame Relay Traffic Shaping에서 설명합니다.



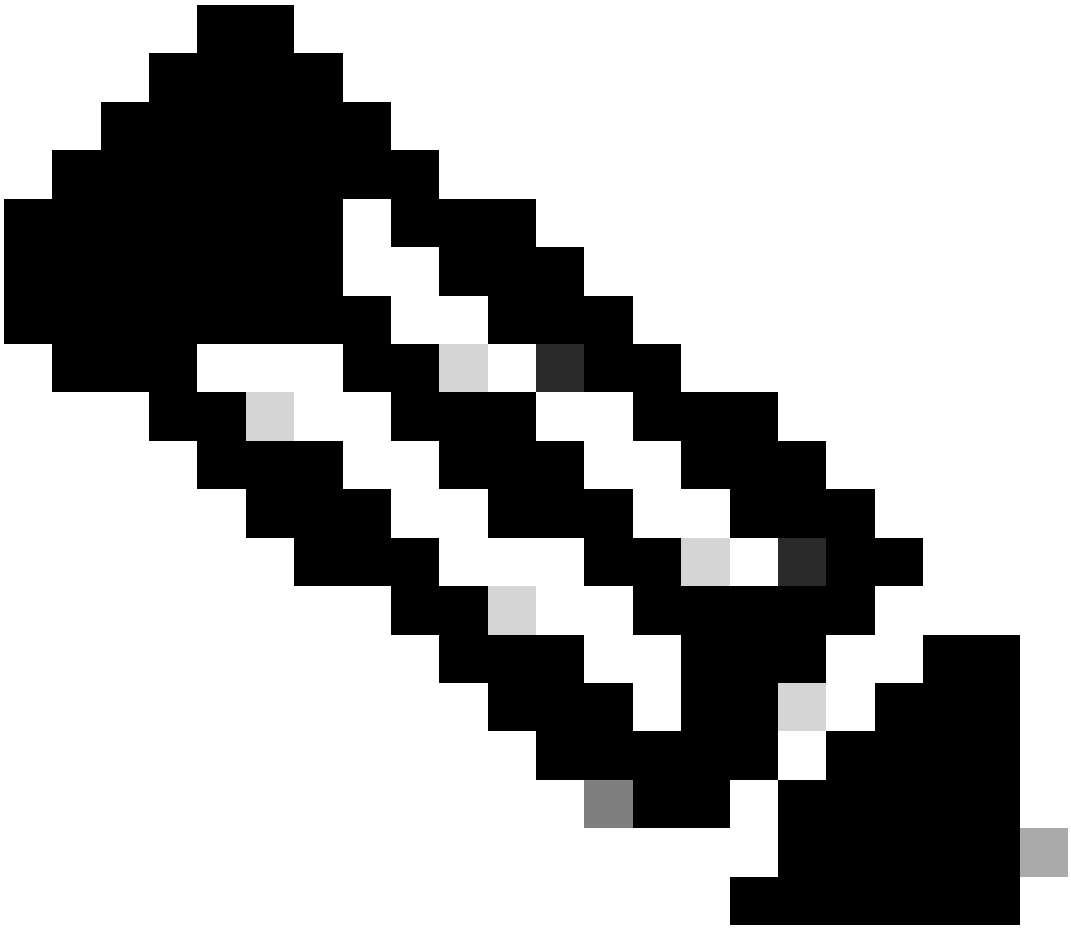
트래픽 출력 표시

초과 버스트(Be)가 0과 다른 값으로 구성된 경우 쉐이퍼는 최대 Bc + Be까지 토큰이 버킷에 저장되도록 허용합니다. 토큰 버킷이 도달할 수 있는 최대 값은 Bc+Be이며 오버플로 토큰은 삭제됩니다. 버킷에 Bc 토큰 이상을 갖는 유일한 방법은 하나 이상의 Tc 동안 모든 Bc 토큰을 사용하지 않는 것이다. 토큰 버킷은 모든 Tc에 Bc 토큰으로 보충되기 때문에 나중에 Bc+Be까지 사용할 수 있도록 사용하지 않은 토큰을 누적할 수 있다.

이와 달리, 클래스 기반 정책 및 속도 기반 정책은 limiting 버킷에 지속적으로 토큰을 추가합니다. 특히 토큰 도착율은 다음과 같이 계산됩니다.

```
(time between packets<which is equal to t-t1>* policer rate)/8 bits per byte
```

즉, 패킷의 이전 도착이 t1이고 현재 시간이 t이면, 토큰 도착률을 기준으로 t-t1 바이트의 값으로 버킷을 업데이트한다.



참고: 트래픽 폴리서는 바이트로 지정된 버스트 값을 사용하며, 이전 공식은 비트에서 바이트로 변환합니다.

다음은 8000bps의 CIR(또는 폴리서 레이트)와 1000바이트의 일반 버스트를 사용하는 예입니다.

<#root>

```
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

토큰 버킷은 1000바이트에서 가득 차기 시작합니다. 450바이트 패킷이 도착하면 토큰 버킷에서 사용할 수 있는 바이트가 충분하기 때문에 패킷이 일치합니다. Conform 작업(전송)은 패킷에 의해 수행되며 토큰 버킷에서 450바이트가 제거됩니다(그리고 550바이트는 남겨둡니다). 다음 패킷이 0.25초 후에 도착하면 다음 공식에 따라 250바이트가 토큰 버킷에 추가됩니다.

```
(0.25 * 8000)/8
```

이 계산은 토큰 버킷에 700바이트를 남깁니다. 다음 패킷이 800바이트이면 패킷이 초과되고 초과 작업(삭제)이 수행됩니다. 토큰 버킷에서 가져온 바이트가 없습니다.

트래픽 셰이핑

Cisco® IOS는 다음 트래픽 셰이핑 방법을 지원합니다.

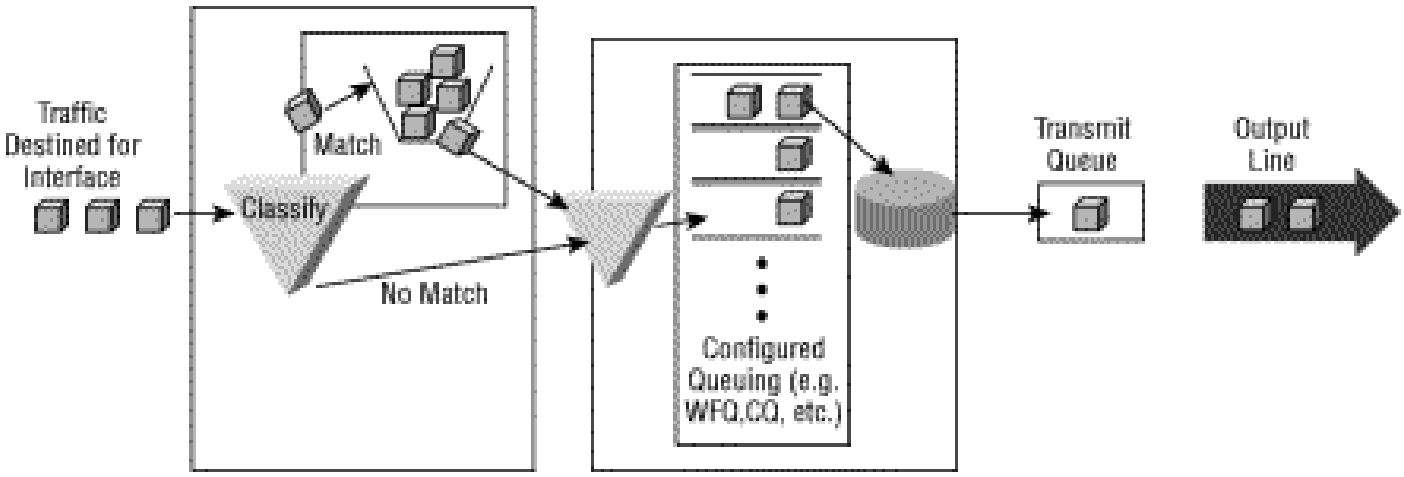
- 일반 트래픽 셰이핑

- 프레임 릴레이 트래픽 셰이핑

- 클래스 기반 셰이핑 및 분산 클래스 기반 셰이핑

모든 트래픽 셰이핑 방법은 구현 면에서 유사하지만, CLI(Command-Line Interface)는 다소 다르며, 서로 다른 유형의 큐를 사용하여 지연된 트래픽을 포함하고 셰이핑합니다. Cisco에서는 모듈형 QoS CLI로 구성되는 클래스 기반 셰이핑 및 분산 셰이핑을 권장합니다 .

다음 다이어그램은 QoS 정책이 어떻게 트래픽을 클래스로 정렬하고 구성된 쉐이핑 속도를 초과하는 패킷을 큐잉하는지 보여줍니다.



트래픽 폴리싱

Cisco IOS는 다음 트래픽 폴리싱 방법을 지원합니다.

- 커밋된 액세스 속도

- 클래스 기반 폴리싱

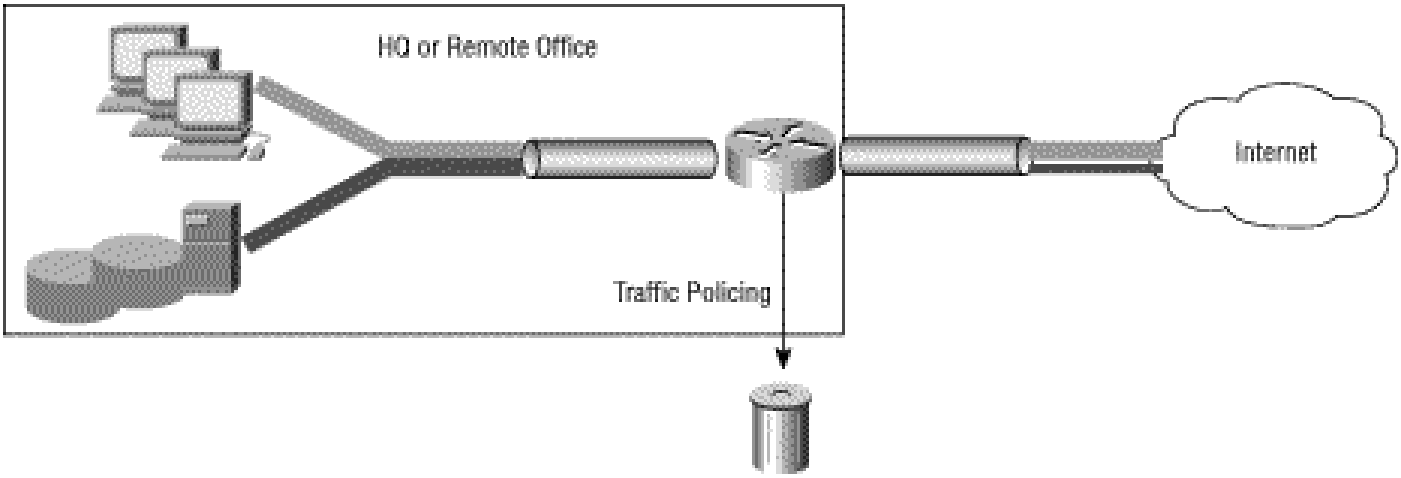
클래스 기반 정책 비교 및 약정된 액세스 속도에 설명된 대로 두 메커니즘은 중요한 기능적 차이를 가집니다. Cisco에서는 QoS 정책이 적용될 때 클래스 기반 정책 및 모듈형 QoS CLI의 기타 기능을 권장합니다.

Police **명령**을 사용하여 트래픽 클래스에 지정된 최대 속도가 있어야 하며, 이 속도를 초과하면 즉시 조치를 취해야 합니다. 즉, **police**

명령을 사용하면 **shape 명령**의 경우처럼 패킷을 버퍼링한 후 나중에 전송하는 옵션이 **아닙니다**.

또한 폴리싱을 통해 토큰 버킷은 패킷이 적용된 속도를 초과하는지 또는 준수하는지 여부를 결정합니다. 두 경우 모두 폴리싱은 구성 가능한 작업을 구현합니다. 여기에는 IP 우선 순위 또는 DSCP(Differentiated Services Code Point)가 포함됩니다.

다음 다이어그램은 일반적으로 QoS 기능이 적용되는 혼잡 지점에서 트래픽 폴리싱의 일반적인 적용을 보여줍니다.



최소 및 최대 대역폭 제어

shape 및 **police 명령**은 모두 출력 속도를 최대 kbps 값으로 제한합니다. 중요한 것은 어느 메커니즘도 혼잡 기간 동안 최소 대역폭 보장을 제공하지 않는다는 것입니다. 이러한 보장을 제공하려면 **bandwidth** 또는 **priority** 명령을 사용합니다.

계층적 정책은 두 가지 서비스 정책, 즉 상위 정책을 사용하여 트래픽 집계에 QoS 메커니즘을 적용하고 하위 정책을 사용하여 트래픽 집계의 플로우 또는 하위 집합에 QoS 메커니즘을 적용합니다. 하위 인터페이스 및 터널 인터페이스와 같은 논리적 인터페이스에는 상위 레벨의 트래픽 기능과 하위 limiting 레벨의 큐잉이 포함된 계층적 정책이 필요합니다. 트래픽 기능은 limiting 초과 패킷에서 볼 수 있는 것처럼 출력 속도를 줄이고 혼잡을 queuing 일으킬 수 있습니다.

다음 컨피그레이션은 최적화되지 않으며 트래픽이 최대 속도로 집계될 때 **경찰**과 **shape** 명령 사이 limiting 의 차이를 설명하기 위해 표시됩니다(이 경우 class-default). 이 컨피그레이션에서는 **police 명령**이 패킷의 크기 및 conform 및 exceed 토큰 버킷에 남아 있는 바이트 수에 따라 하위 클래스에서 패킷을 전송합니다. (트래픽 폴리싱을 참조하십시오.) 그 결과, 경찰 기능이 우선 순위 기능이 보장한 것을 무시하므로 VoIP(Voice over IP) 및 IP(Internet Protocol) 클래스에 부여되는 속도를 보장할 수 없습니다.

그러나 **shape** 명령을 사용하면 계층적 큐잉 시스템이므로 모든 보증이 이루어집니다. 다시 말해, 제공된 로드가 셰이프 속도를 초과하면 VoIP 및 IP 클래스의 속도가 보장되고 클래스 기본 트래픽(하위 수준)에 어떤 삭제도 발생합니다.

⚠ **주의**: 이 컨피그레이션은 권장되지 않으며, 트래픽 집계를 제한할 때 **경찰**과 **shape** 명령 간의 차이를 설명하기 위해 표시됩니다.

```
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
```

이전 컨피그레이션이 타당하기 위해서는 폴리싱을 셰이핑으로 교체해야 합니다.

예를 들면 다음과 같습니다.

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

참고: 상위 및 하위 정책에 대한 자세한 내용은 우선 순위 클래스에 대한 QoS 하위 서비스 정책을 참조하십시오.

관련 정보

- **QoS(Quality of Services) 기술 지원**
- **기술 지원 및 문서 - Cisco Systems**

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.