Confronta criteri traffico e forma traffico per limitare la larghezza di banda

Sommario

Introduzione

Questo documento descrive le differenze funzionali tra il traffic shaping e il traffic policing, che limitano entrambi la velocità di output.

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni

[nei suggerimenti tecnici.]

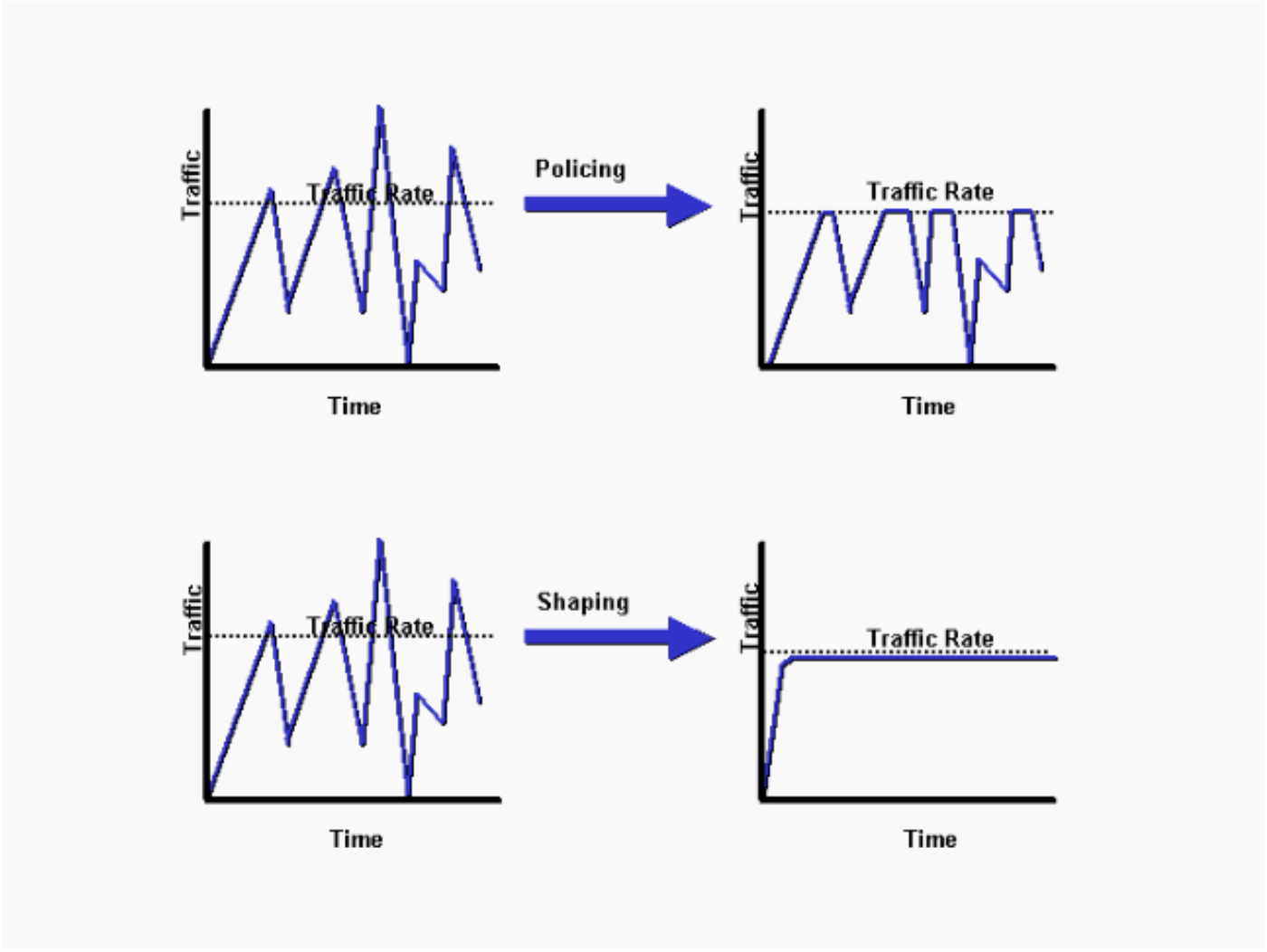
Premesse

Questo documento chiarisce le differenze funzionali tra traffic shaping e policing. Entrambi limitano la velocità di output del traffico. Entrambi i meccanismi utilizzano un bucket di token come misuratore del traffico per misurare la velocità del pacchetto. Per ulteriori informazioni sui bucket di token, vedere Che cos'è un bucket di token

Differenze tra traffic policing e traffic shaping

Il traffic shaping propaga i burst. Quando la velocità del traffico raggiunge la velocità massima configurata, il traffico in eccesso viene eliminato (o contrassegnato diversamente). Il risultato è una velocità di trasmissione che appare come un'onda a dente di sega con picchi ad andamento positivo e negativo. A differenza del traffic policing, il traffic shaping mantiene in una coda i pacchetti in eccesso e poi pianifica una trasmissione successiva del traffico in eccesso nell'arco di incrementi temporali. Il risultato del traffic shaping è una velocità di trasmissione dei pacchetti fluida e uniforme.

Nel diagramma successivo vengono illustrate le principali differenze tra le due opzioni di traffico.



Il traffic shaping implica la presenza di una coda e di una memoria sufficiente per salvare nel buffer i pacchetti rimandati per una trasmissione successiva, a differenza del traffic policing. Le code sono un concetto di uscita; i pacchetti che lasciano un'interfaccia vengono messi in coda e possono essere modellati. Solo il traffic policing può essere applicato al traffico in entrata in un'interfaccia. Quando attivate il shaping, assicuratevi di disporre di memoria sufficiente. Inoltre, il shaping richiede una funzione che programma la trasmissione successiva di qualsiasi pacchetto ritardato. Questa funzionalità di pianificazione consente di organizzare la coda di shaping in code diverse. Esempi di questa funzionalità sono Class Based Weighted Fair Queuing (CBWFQ) e Low Latency Queuing (LLQ).

Criteri di selezione

La tabella seguente elenca le differenze tra il shaping e il policing per agevolare la scelta della soluzione di traffico appropriata.

	Traffic shaping	Traffic policing
Obiettivo	Memorizzare nel buffer e mettere in coda i pacchetti in eccesso rispetto alle velocità di commit.	Elimina (o segnala) i pacchetti in eccesso rispetto alle tariffe impegnate. Non viene eseguito il salvataggio nel buffer.*
Frequenza di aggiornamento dei token	Maggiore all'inizio di un intervallo temporale. (È richiesto un numero minimo di intervalli.)	Continuo in base alla formula:`1 / tasso informazioni vincolate`
Valori dei token	Configurati in bit al secondo.	Configurati in byte.
Opzioni di configurazione	• Comando shape nell'interfaccia a riga di comando per la qualità del servizio modulare (MQC) per implementare il Class-Based Shaping. • Comando frame-relay traffic-shape per implementare FRTS (Frame Relay Traffic Shaping). • Comando traffic-shape per implementare GTS (Generic Traffic Shaping).	• Comando police in MQC per implementare il Class-Based Policing. • Comando rate-limit per implementare CAR (Committed Access Rate).
Applicabile in entrata	No	Sì
Applicabile in uscita	Sì	Sì
Burst	Controlla i picchi e smussa la velocità di output su almeno otto intervalli di tempo.	Propaga i burst. Non esegue alcun livellamento.

	Utilizza l'algoritmo leaky bucket per ritardare il traffico, ottenendo un effetto uniforme.	
Vantaggi	Meno probabilità di eliminare i pacchetti in eccesso poiché vengono salvati nel buffer. (Salvataggio nel buffer di un numero di pacchetti pari al massimo alla lunghezza della coda. Le cadute possono verificarsi se il traffico in eccesso viene mantenuto a velocità elevate.) In genere, evita ritrasmissioni a causa di pacchetti eliminati.	Controlla la velocità di trasmissione tramite le perdite di pacchetti. Evita ritardi dovuti a queuingerrori.
Svantaggi	Può introdurre ritardi dovuti a queuingcode particolarmente profonde.	Elimina i pacchetti in eccesso (quando configurati), limita le dimensioni della finestra TCP e riduce la velocità di output complessiva dei flussi di traffico interessati. Dimensioni di burst eccessivamente aggressive possono portare a perdite di pacchetti in eccesso e limitare la velocità di output complessiva, in particolare con i flussi basati su TCP.
Pacchetti contrassegnati diversamente in via opzionale	No	Sì (con funzionalità CAR legacy).

* Sebbene il controllo non applichi un buffer, un queuing meccanismo configurato si applica ai pacchetti conformati che devono essere accodati mentre attendono di essere serializzati sull'interfaccia fisica.

Frequenza di aggiornamento dei token

Una differenza fondamentale tra il traffic shaping e il traffic policing è la velocità con cui vengono ricaricati i token, Sia il shaping che il policing utilizzano la metafora del token bucket. Un token bucket non dispone di policy di eliminazione o priorità.

Con funzionalità token bucket:

- I token vengono inseriti nel bucket a una determinata velocità.
- Ogni token consente all'origine di inviare un determinato numero di bit alla rete.

- Per inviare un pacchetto, il regolatore del traffico deve essere in grado di rimuovere dal bucket un certo numero di token uguale alla dimensione del pacchetto in termini di rappresentazione.

- Se nel bucket non sono presenti token sufficienti per inviare un pacchetto, il pacchetto attende finché il bucket non dispone di token sufficienti (nel caso di uno shaper), oppure il pacchetto viene scartato o contrassegnato (nel caso di un policer).

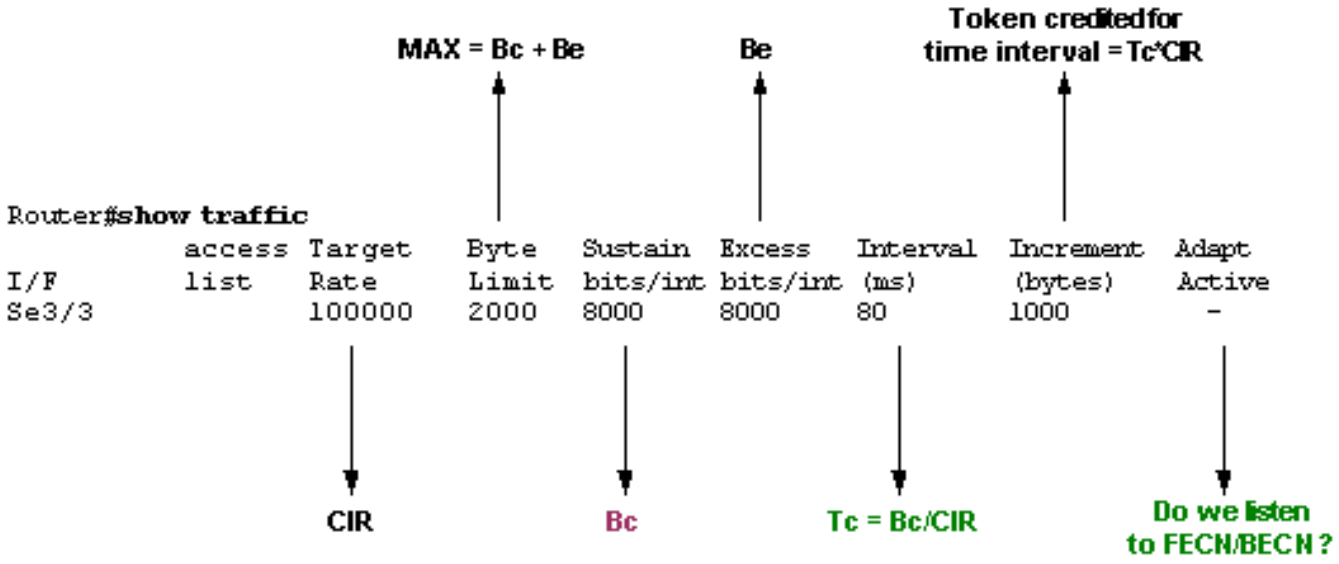
- Il bucket stesso ha una capacità specificata. Se il bucket riempie la capacità, i nuovi token in arrivo vengono eliminati e non sono disponibili per i pacchetti futuri. Pertanto, in qualsiasi momento, il burst di dimensioni maggiori che una sorgente può inviare alla rete è all'incirca proporzionale alla dimensione del bucket. Un token bucket consente l'esplosione ma la limita.

Il comando Shaping incrementa il bucket di token a intervalli di tempo che utilizzano un valore di bit al secondo (bps). Uno shaper utilizza la formula seguente:

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

In questa equazione, Bc rappresenta il burst garantito e CIR indica la banda minima garantita. (Per ulteriori informazioni, consultare Configurazione del Frame Relay Traffic Shaping.) Il valore di Tc definisce l'intervallo temporale durante il quale vengono inviati i bit Bc per mantenere la velocità media del CIR in secondi.

L'intervallo per Tc è compreso tra 10 ms e 125 ms. Con DTS (Distributed Traffic Shaping) su Cisco serie 7500, il Tc minimo è di 4 ms. Il router calcola internamente questo valore in base ai valori di CIR e Bc. Se Bc / CIR è inferiore a 125 ms, utilizza il Tc calcolato da tale equazione. Se il valore Bc/CIR è superiore o uguale a 125 ms, viene utilizzato un valore Tc interno se Cisco IOS® determina che il flusso del traffico può essere più stabile con un intervallo inferiore. Utilizzare il comando **show traffic-shape** per determinare se il router utilizza un valore interno per Tc o il valore configurato nella riga di comando. L'output di esempio successivo del comando **show traffic-shape** è illustrato in Show Commands for Frame Relay Traffic Shaping.



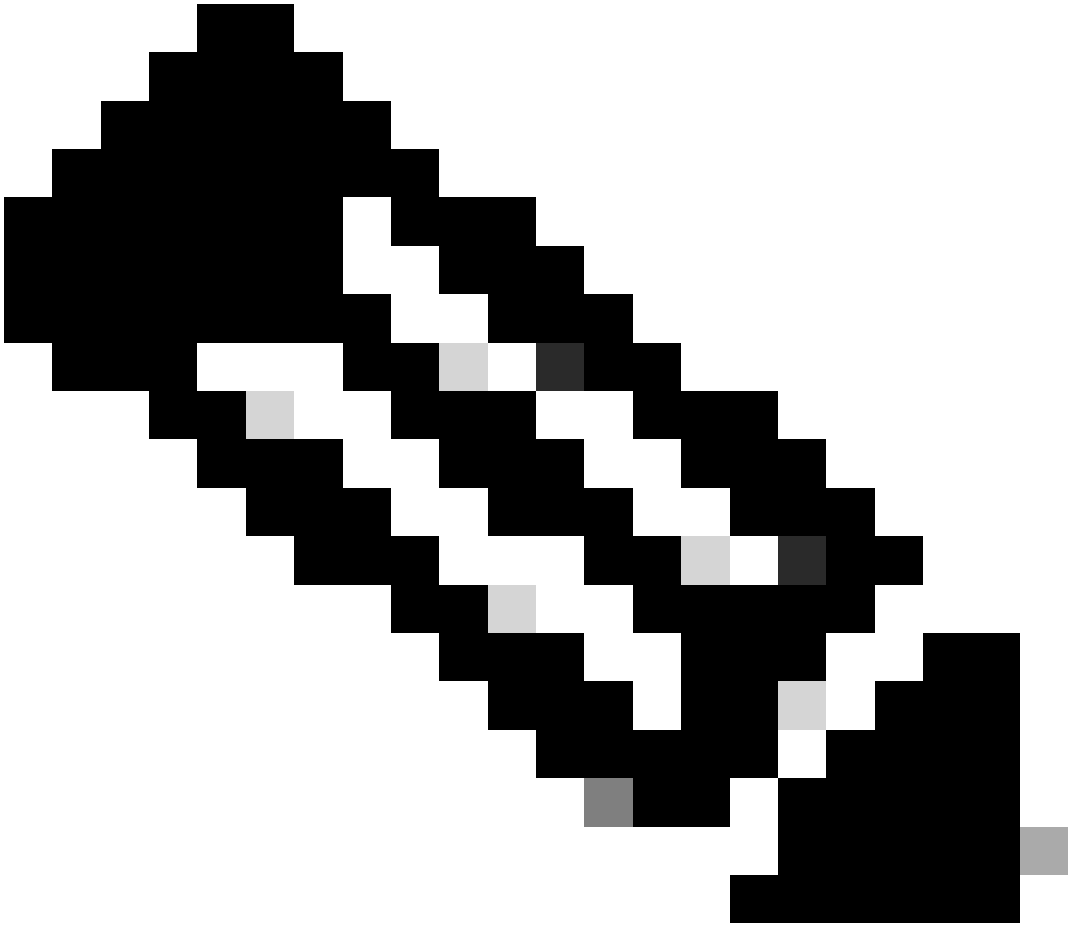
mostra output traffico

Quando il burst in eccesso (Be) è configurato su un valore diverso da 0, il traffic shaping consente di memorizzare i token nel bucket, fino a Bc + Be. Il valore massimo che il token bucket possa mai raggiungere è Bc + Be e i token in overflow vengono eliminati. L'unico modo per avere un valore maggiore dei token Bc nel bucket è non utilizzare tutti i token Bc durante uno o più Tc. Poiché il token bucket viene ricaricato a ogni Tc con token Bc, è possibile accumulare token inutilizzati per un uso successivo per un valore massimo pari a Bc + Be.

Al contrario, il policing basato su classi e la velocità limiting aggiungono continuamente token al bucket. In particolare, la velocità di arrivo dei token viene calcolata come segue:

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

In altre parole, se l'arrivo precedente del pacchetto era a t1 e il momento corrente è t, il bucket viene aggiornato con byte di valore t-t1 in base alla velocità di arrivo del token.



Nota: un regolatore del traffico utilizza i valori burst specificati in byte e la formula precedente converte i bit in byte.

Questo è un esempio in cui si usa un CIR (o frequenza policer) di 8000 bps e un normale burst di 1000 byte:

<#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
```

I bucket di token iniziano completi a 1000 byte. Un pacchetto da 450 byte in arrivo è conforme perché nel token bucket sono disponibili abbastanza byte. L'azione di conformità (trasmissione) viene effettuata dal pacchetto e 450 byte vengono rimossi dal bucket di token (e rimangono 550 byte). Se il pacchetto successivo arriva 0,25 secondi dopo, vengono aggiunti 250 byte al bucket del token, come indicato nella formula seguente:

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

Il calcolo lascia 700 byte nel token bucket. Se il pacchetto successivo è di 800 byte, questo risulta in eccesso e viene eseguita l'azione exceed action (drop). Nessun byte viene preso dal token bucket.

Traffic Shaping

Cisco® IOS supporta i seguenti metodi di traffic shaping:

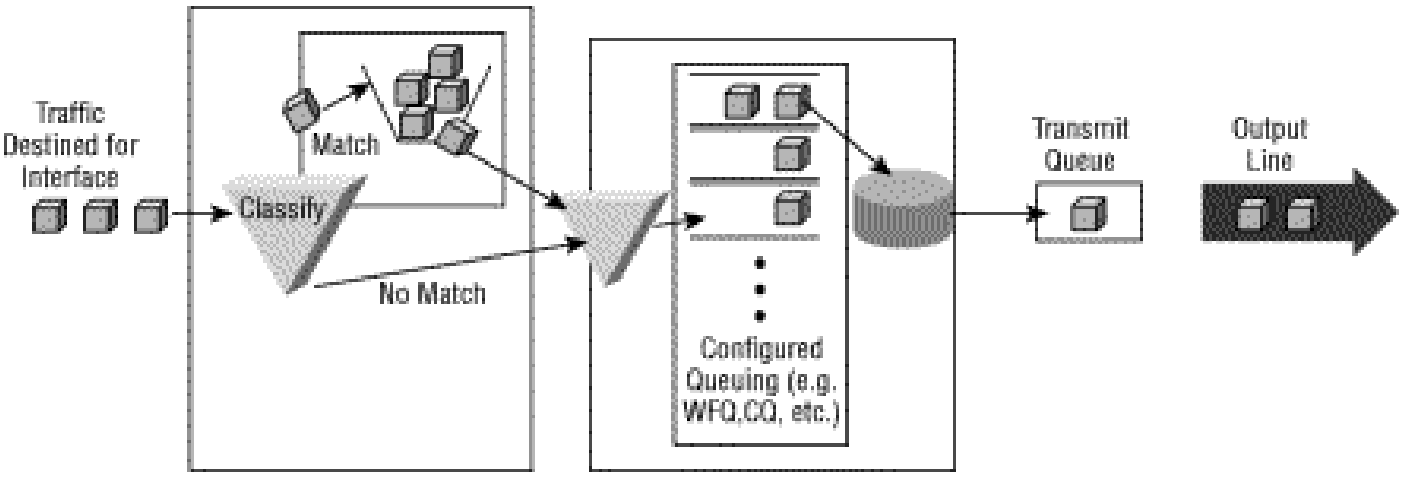
- GTS (Generic Traffic Shaping)

- FRTS (Frame Relay Traffic Shaping)

- Class-Based Shaping e Distributed Class-Based Shaping

Tutti i metodi di traffic shaping presentano un'implementazione simile, anche se le interfacce a riga di comando (CLI) sono in qualche modo diverse, e utilizzano diversi tipi di code per contenere e controllare il traffico che viene rinviato. Cisco consiglia il class-based shaping e il distributed shaping, configurati con la CLI QoS modulare.

Nel diagramma successivo viene illustrato come un criterio QoS dispone il traffico in classi e accoda i pacchetti che superano le velocità di shaping configurate.



Traffic Policing

Cisco IOS supporta i seguenti metodi di monitoraggio del traffico:

- CAR (Committed Access Rate)

- Class-Based Policing

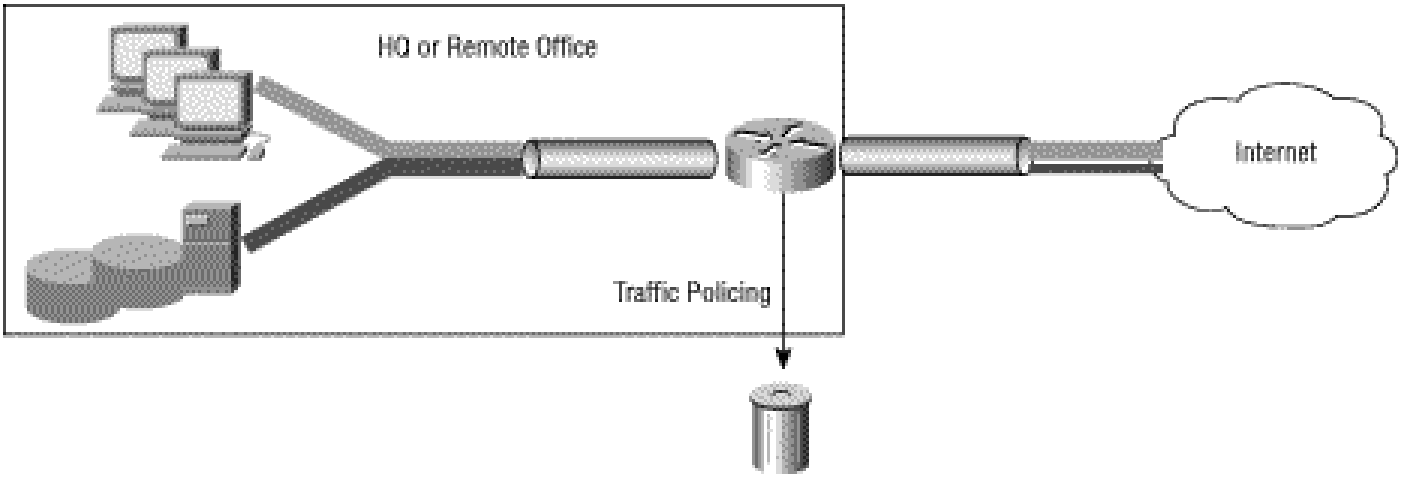
I due meccanismi presentano differenze funzionali importanti, come illustrato in Confronta classi di Policing e Committed Access Rate. Cisco consiglia il policing basato su classi e altre funzionalità della CLI QoS modulare quando vengono applicati i criteri QoS.

Usare il comando **Police** per specificare che a una classe di traffico deve essere imposta una velocità massima e, se questa velocità viene

superata, è necessario eseguire un'azione immediata. In altre parole, con il comando **police, non è possibile salvare il pacchetto nel buffer e inviarlo in un secondo momento, come nel caso del comando shape.**

Inoltre, con il traffic policing, il token bucket determina se un pacchetto è in eccesso o è conforme alla velocità applicata. In entrambi i casi, il policing implementa un'azione configurabile, che include la precedenza IP o il DSCP (Differentiated Services Code Point).

Il diagramma successivo mostra un'applicazione comune del traffic policing in un punto di congestione, dove normalmente si applicano le funzionalità QoS.



Controlli per la larghezza di banda minima e massima

Entrambi i comandi **shape e police limitano la velocità di trasmissione a un valore massimo di kbps.** È importante sottolineare che nessuno dei due meccanismi fornisce una garanzia di larghezza di banda minima durante i periodi di congestione. Utilizzare il comando **bandwidth o priority per fornire tali garanzie.**

Un criterio gerarchico utilizza due criteri di servizio: un criterio padre per applicare un meccanismo QoS a un'aggregazione di traffico e un criterio figlio per applicare un meccanismo QoS a un flusso o a un subset dell'aggregazione. Le interfacce logiche, ad esempio le sottointerfacce e le interfacce tunnel, richiedono un criterio gerarchico con la limiting funzionalità traffico al livello superiore e l'accodamento ai livelli inferiori. La limiting funzione traffic-feature riduce la velocità di uscita e (presumibilmente) crea una congestione, come dimostrano i pacchetti in queuing eccesso.

La configurazione successiva non è ottimale e mostra la differenza tra il comando **Police** e il comando **shape** quando il traffico viene aggregato, in questo caso limiting una classe predefinita, a una velocità massima. In questa configurazione, il comando **Police** invia i pacchetti dalle classi figlie in base alle dimensioni del pacchetto e al numero di byte che rimangono nei bucket di conformità e che superano i bucket dei token. (Consultare Traffic Policing.) Di conseguenza, le tariffe fornite alle classi VoIP (Voice over IP) e IP (Internet Protocol) non possono essere garantite, poiché la funzionalità di polizia ignora le garanzie fornite dalla funzionalità di priorità.

Tuttavia, se si utilizza il comando shape, il risultato è un sistema di accodamento gerarchico e tutte le garanzie vengono fornite. In altre parole, quando il carico offerto supera la velocità di shaping, la velocità delle classi VoIP e IP è garantita e il traffico di classe predefinita (al livello secondario) subisce eventuali perdite.

⚠ **Attenzione**: questa configurazione non è consigliata e viene mostrata per illustrare la differenza tra il comando **Police** e il comando **shape** quando limita un'aggregazione di traffico.

```
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
```

Affinché la configurazione precedente abbia senso, è necessario sostituire la policy con la shaping.

Ad esempio:

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

Nota: per ulteriori informazioni sui criteri padre e figlio, vedere Criteri servizio figlio QoS per classe di priorità.

Informazioni correlate

- **Supporto tecnologico per Qualità del servizio (QoS)**
- **Documentazione e supporto tecnico – Cisco Systems**

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).