

帯域幅制限に関するトラフィック ポリシングと トラフィック シェーピングの比較

目次

[概要](#)

[はじめに](#)

[表記法](#)

[前提条件](#)

[使用するコンポーネント](#)

[ポリシングとシェーピング](#)

[選択基準](#)

[トークンのリフレッシュ レート](#)

[トラフィック シェーピング](#)

[トラフィック ポリシング](#)

[最小帯域幅制御と最大帯域幅制御](#)

[関連情報](#)

概要

この文書では、シェーピングとポリシングの機能的な相違点を明らかにします。どちらのメカニズムもトークン バケットをトラフィック メータとして使用してパケット レートを測定しますが、機能面で重要な違いがあります。（トークン バケットについては、『[トークン バケットとは](#)』を参照してください）。

[はじめに](#)

[表記法](#)

ドキュメント表記の詳細は、『[シスコ テクニカル ティップスの表記法](#)』を参照してください。

[前提条件](#)

このドキュメントに関する固有の要件はありません。

[使用するコンポーネント](#)

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始して

います。対象のネットワークが実稼働中である場合には、どのような作業についても、その潜在的な影響について確実に理解しておく必要があります。

ポリシングとシェーピング

次の図は、その主な違いを示しています。トラフィックポリシングはバーストを伝搬します。トラフィックレートが最大レートの設定値に達すると、超過トラフィックが廃棄（またはマーキング）されます。その結果、出力レートは頂上と谷間のある鋸歯状になります。ポリシングと異なり、トラフィックシェーピングでは、超過パケットはキューまたは漏出バケットに保持され、一定の時間間隔でスケジュールされてから遅れて伝送されます。トラフィックシェーピングの結果、パケットの出力レートは平滑化されます。

シェーピングでは遅延パケットをバッファリングするためのキューと十分なメモリが存在する必要がありますが、ポリシングではそれらは必要ありません。キューイングは発信に関する概念です。インターフェイスから送出されるパケットはキューイングされ、シェーピングを適用できます。インターフェイスでは着信トラフィックに対してポリシングだけを適用できます。シェーピングを有効にする場合は十分なメモリがあることを確認してください。また、シェーピングには遅延パケットを後で伝送するためのスケジューリング機能が必要です。このスケジューリング機能を使用すると、シェーピングキューを別々のキューに編成できます。スケジューリング機能の例としては、Class Based Weighted Fair Queuing (CBWFQ; クラスベースの重み付け均等化キューイング) や低遅延キューイング (LLQ) などがあります。

選択基準

次の表はシェーピングとポリシングの相違点をまとめたもので、最適なソリューションを選択する際に役立ちます。

	シェーピング	ポリシング
目標	認定レートを超える超過パケットをバッファリングし、キューに格納する。	認定レートを超える超過パケットを廃棄（またはマーキング）する。バッファリングしない。*
トークンのリフレッシュレート	時間間隔の開始時に補充される。（最小の数の時間間隔が必要）	次の式に基づき継続する。 $1 /$
トークン値	ビット/秒で設定する。	バイトで設定する。
設定オプション	• クラスベースシェーピングを実装する場合は、モジュラ QoS コマンドライン インターフェイス (MQC) で <code>shape</code> コ	• クラスベースシェーピングを実装する場合は、MQC で <code>police</code> コマンドを使用する。

	<p>マンドを使用する。</p> <ul style="list-style-type: none"> • Frame Relay Traffic Shaping (FRTS; フレームリレートラフィックシェーピング) を実装する場合は、 frame-relay traffic-shape コマンドを使用する。 • Generic Traffic Shaping (GTS; ジェネリックトラフィックシェーピング) を実装する場合は、 traffic-shape コマンドを使用する。 	<ul style="list-style-type: none"> • Committed Access Rate (CAR; 専用アクセスレート) を実装する場合は、 rate-limit コマンドを使用する。
着信時の適用	なし	○
発信時の適用	○	○
バースト	8 回の時間間隔以上にわたって出力レートを平滑化することにより、バーストを制御する。漏出パケットを使用してトラフィックを遅らせることで、平滑化効果を達成する。	バーストを伝搬する。 ○。平滑化は行わない。 ○。
利点	超過パケットがバッファリングされるため、超過パケットが廃棄される可能性は少ない (パケットは最大でキューの長さまでバッファリングされる。超過トラフィックが高いレートで持続した場合は廃棄される可能性がある)。一般に、パケットの廃棄による再転送が起こらなくなる。	パケットの廃棄によって出力レートを制御する。キューイングによる遅延が起こらなくなる。
短所	キューイングによる遅延が発生するおそれがある (特にキューが深い場合)。	超過パケットが廃棄され (設定している場合)、TCP ウィンドウサイズが絞られて、該当するトラフィックストリームの全体的な出力レートが下がる。バースト

		サイズをあまりにも大きく設定すると、超過パケットの廃棄が起こり、特に TCP ベースのフローの場合に全体的な出力レートが抑制されるおそれがある。
オプションの パケット マーキング	なし	(レガシー CAR 機能 使用時)

* ポリシングではバッファリングは行われませんが、物理インターフェイスでシリアルライズされるのを待つ間キューに入れる必要がある「適合」パケットには、設定されたキューイングメカニズムが適用されます。

トークンのリフレッシュレート

シェーピングとポリシングの主な違いはトークンが補充される速度にあります。この項では、その違いについて見ていきます。

簡単に言えば、シェーピングとポリシングはどちらもトークンバケットメタファを使用します。トークンバケット自体には、廃棄ポリシーまたは優先順位ポリシーはありません。トークンバケットメタファは次のように動作します。

- トークンはある一定のレートでバケットに入れられます。
- 各トークンは、発信元がある一定のビット数をネットワークに送信するための許可を表します。
- パケットを送信するには、トラフィック調整機能によって、パケットサイズと表現上等しい数のトークンをバケットから削除する必要があります。
- パケットを送信するために十分なトークンがバケットにない場合は、バケットに十分な数のトークンが溜まるまで待機するか (シェーパの場合)、廃棄またはマークダウンされます (ポリサーの場合)。
- バケット自体には指定された容量があります。バケットの容量がいっぱいになると、新しく着信したトークンは廃棄されて、今後のパケットのためにそれらのトークンを使用できなくなります。したがって、発信元がネットワークに送信できる最大のバーストは常にバケットのサイズにほぼ比例します。トークンバケットでは、バーストが許可されますが、制限があります。

トークンバケットメタファを念頭に置きながら、シェーピングとポリシングがどのようにしてバケットにトークンを追加するのかを見ていきましょう。

シェーピングでは、ビット/秒 (bps) 値によって定められた時間間隔でトークンバケットが補充されます。シェーパは次の数式を使用します。

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

この数式では、Bc は Committed Burst (認定バースト) を表し、CIR は Committed Information Rate (認定情報レート) を表します。(詳細は、「[フレームリレートラフィックシェーピングの設定](#)」を参照してください)。Tc の値は、秒単位での CIR の平均レートを維持するために Bc ビットを送信する時間間隔を定義します。

Tc の範囲は、10 ミリ秒から 125 ミリ秒の間です。Cisco 7500 シリーズの Distributed Traffic Shaping (DTS; 分散トラフィックシェーピング) の最小 Tc は 4 ミリ秒です。ルータは、CIR 値と Bc 値に基づいて、この値を内部で計算します。Bc/CIR が 125 ミリ秒よりも小さい場合は、この式から算出された Tc が使用されます。Bc/CIR が 125 ミリ秒と等しいか、それよりも大きい場合は、トラフィックフローがより短い時間間隔において安定すると Cisco IOS(R) によって判断されたときに、内部 Tc 値が使用されます。ルータで Tc の内部値を使用しているか、コマンドラインで設定された値を使用しているかを確認するには、**show traffic-shape** コマンドを使用します。次の show traffic-shape コマンドの出力例に関する詳細は、『[フレームリレートラフィックシェーピングのための show コマンド](#)』を参照してください。

超過バースト (Be) がゼロ以外の値に設定されている場合、シェーパは Bc + Be に達するまでトークンをバケットに保持します。トークンバケットの上限値は Bc + Be であり、オーバーフローしたトークンは廃棄されます。Bc を超えるトークンをバケットに保持する唯一の方法は、1 回または複数回の Tc の間にすべての Bc トークンを使用しないようにすることです。トークンバケットには Bc のトークンが Tc ごとに補充されるため、未使用のトークンを Bc + Be まで蓄積して後で使用できます。

それに対して、クラスベースのポリシングでは、トークンがバケットに継続的に追加されます。具体的には、トークン到達レートは次のようにして計算されます。

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

つまり、前のパケットが到達した時間を t1、現在の時間を t とした場合、トークン到達レートに基づいて t-t1 に相当するバイト数だけバケットが更新されます。トラフィックポリサーはバイトで指定されたバースト値を使用するため、上記の数式ではビットからバイトに換算されている点に注意してください。

CIR (つまり、ポリサーレート) として 8000 bps、通常バーストとして 1000 バイトを使用した例を次に示します。

```
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
```

トークンバケットには最初 1000 バイト満たされています。450 バイトのパケットが到達した場合、トークンバケットには十分なバイト数があるため、パケットは適合します。したがって適合アクション (伝送) が実行され、トークンバケットから 450 バイトが削除されます (残り 550 バイト)。次のパケットが .25 秒後に到達したとすると、次の数式に基づいてトークンバケットに 250 バイトが追加されます。

$(0.25 * 8000) / 8$

この計算により、トークンバケットの内容は 700 バイトになります。次のパケットが 800 バイトであれば、パケットは超過し、超過アクション (廃棄) が実行されます。トークンバケットからは 1 バイトも削除されません。

[トラフィックシェーピング](#)

Cisco IOS は次のトラフィックシェーピング方式をサポートしています。

- [ジェネリックトラフィックシェーピング](#)

- [フレームリレートラフィックシェーピング](#)
- [クラスベースシェーピングと分散クラスベースシェーピング](#)

どのトラフィックシェーピング方式も実装の点ではよく似ていますが、それぞれの Command Line Interface (CLI; コマンド行インターフェイス) は若干異なります。各方式は遅延されるトラフィックを格納およびシェーピングするために異なるタイプのキューを使用します。シスコでは、クラスベースシェーピングと分散シェーピングを推奨しています。これらはモジュラ QoS CLI を使用して設定されます。

次の図は、どのようにして QoS ポリシーがトラフィックをクラスに分類し、設定されたシェーピングレートを超えるパケットをキューイングするかを示しています。

[トラフィックポリシング](#)

Cisco IOS は次のトラフィックポリシング方式をサポートしています。

- [専用アクセスレート](#)
- [クラスベースポリシング](#)

「[クラスベースポリシングおよび専用アクセスレートの比較](#)」で説明されているように、これらの 2 つのメカニズムには、機能面で重要な違いがあります。シスコでは、QoS ポリシーを適用する際は、クラスベースポリシングと、モジュラ QoS CLI の他の機能を推奨しています。

トラフィックのクラスに課された最大レートと、そのレートを超えたときにただちに実行する必要があるアクションを指定するには、`police` コマンドを使用します。つまり、`police` コマンドには、`shape` コマンドとは異なり、パケットをバッファリングして後で送出するためのオプションはありません。

また、ポリシングでは、パケットが適用レートを超えているか、または適用レートに適合しているかは、トークンパケットによって決まります。どちらの場合でも、IP 優先順位や Differentiated Services Code Point (DSCP) の設定などの設定可能なアクションが実装されます。

次の図は、輻輳ポイント (QoS 機能は一般にこのポイントで適用される) でのトラフィックポリシングの一般的な適用方法を示しています。

[最小帯域幅制御と最大帯域幅制御](#)

`shape` コマンドも `police` コマンドも、出力レートを最大の kbps 値に制限します。重要なのは、どちらのメカニズムも輻輳時に最小帯域幅を保証しない点です。このような最小帯域幅の保証を提供するには、`bandwidth` または `priority` コマンドを使用します。

階層ポリシーでは 2 つのサービスポリシーが使用されます。1 つは、トラフィック集約に QoS メカニズムを適用する親ポリシー、もう 1 つは集約の中の個々のフローまたはサブセットに QoS メカニズムを適用する子ポリシーです。サブインターフェイスやトンネルインターフェイスなどの論理インターフェイスには、親レベルでのトラフィック制限機能とそれよりも低いレベルでのキューイングを使用した階層ポリシーが必要です。トラフィック制限機能を使用すると、超過パケットをキューイングしたときに見られるように、出力レートが低下し、(おそらく) 輻輳が発生します。

下記の設定は最適ではありませんが、トラフィック集約 (このケースでは `class-default`) を最大レートに制限する際の、`police` コマンドと `shape` コマンドの違いを示すために記載しています。

この設定の **police** コマンドでは、パケットのサイズ、および適合トークン バケットと超過トークン バケットに残っているバイト数に基づいて子クラスからパケットを送信しています (「[トラフィック ポリシング](#)」を参照してください)。その結果、Voice over IP (VoIP) クラスと Internet Protocol (IP; インターネット プロトコル) クラスに設定されたレートは保証されない可能性があります。これは、**priority** 機能によって与えられた保証が、**police** 機能によって無効になるためです。

しかし、**shape** コマンドを使用すると、階層的キューイング システムが実現され、すべてが保証されます。つまり、シェープ レートを超える負荷が発生すると、VoIP および IP クラスはそれぞれのレートが保証され、class-default トラフィックが (子レベルで) 廃棄されます。

注意 : この設定を使用することは推奨されません。この設定は、トラフィック集約を制限する際の、**police** コマンドと **shape** コマンドの違いを示す目的で記載されているだけです。

```
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
```

上記の設定を有効なものにするには、ポリシングをシェーピングに置き換える必要があります。次に、例を示します。

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

親ポリシーと子ポリシーの詳細については、『[プライオリティ クラスの QoS 子サービス ポリシ](#)
[二](#)』を参照してください。

[関連情報](#)

- [QoS に関するサポート ページ](#)
- [テクニカルサポート - Cisco Systems](#)