Cisco IOS XE Bengaluru 17.5.x（Catalyst 9300 スイッチ）Quality of Service コンフィギュレーション ガイド

初版：2021 年 4 月 1 日

最終更新：2023 年 8 月 3 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

第 1 章

自動 QoS の設定 1

自動 QoS の前提条件 1

自動 QoS の制約事項 1

自動 QoS の設定に関する情報 2

自動 QoS の概要 2

自動 QoS 短縮機能の概要 3

自動 QoS グローバル設定テンプレート 3

自動 QoS ポリシーとクラス マップ 3

実行コンフィギュレーションでの自動 QoS の影響 3

実行コンフィギュレーションでの自動 QoS 短縮機能の影響 4

自動 QoS の設定方法 5

自動 QoS の設定 5

自動 QoS のアップグレード 8

自動 QoS 短縮機能のイネーブル化 10

自動 QoS の監視 11

自動 QoS に関するトラブルシューティング 11

自動 QoS の設定例 12

例：auto qos trust cos 12

例：auto qos trust dscp 14

例：auto qos video cts 16

例：auto qos video ip-camera 19

例：auto qos video media-player 21

例：auto qos voip trust 23

例：auto qos voip cisco-phone 25

例：auto qos voip cisco-softphone 28
例：auto qos global compact 32
自動 QoS の関連情報 32
自動 QoS の機能履歴 33

第 2 章 QoS の設定 35
QoS の前提条件 35
QoS コンポーネント 36
QoS の用語 37
QoS の概要 37
モジュラ QoS CLI 37
有線アクセスでサポートされる QoS 機能 38
階層型 QoS 38
QoS の実装 39
レイヤ 2 フレームのプライオリティ ビット 40
レイヤ 3 パケットのプライオリティ ビット 41
分類を使用したエンドツーエンドの QoS ソリューション 41
パケット分類 41
パケットと合わせて伝搬される情報に基づく分類 42
デバイス固有の情報に基づく分類 44
QoS 有線モデル 44
入力ポートのアクティビティ 44
出力ポートのアクティビティ 45
分類 45
アクセス コントロール リスト 45
クラス マップ 46
存続可能時間分類 47
レイヤ 3 パケット長分類 48
レイヤ 2 SRC-Miss または DST-Miss の分類 49
ポリシー マップ 49
物理ポートのポリシー マップ 50

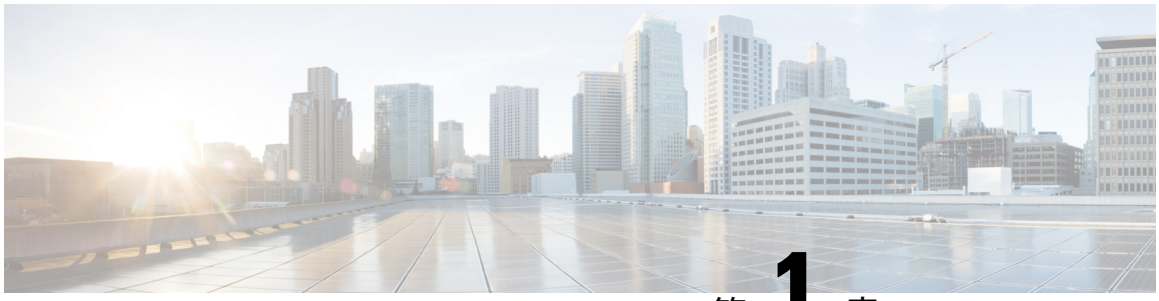
VLAN のポリシーマップ 51
QoS プロファイル 51
セキュリティグループ分類 51
SGT ベースの QoS 52
SGACL との DGID の共有 52
SGT ベースの QoS の制限 52
入力ポート FIFO パーサー 53
ポリシング 55
トークンバケット アルゴリズム 55
マーキング 56
パケット ヘッダーのマーキング 56
スイッチ固有の情報のマーキング 57
テーブル マップのマーキング 57
トラフィックの調整 58
ポリシング 59
シングルレート 2 カラー ポリシング 59
デュアルレート 3 カラー ポリシング 60
シェーピング 60
クラスベース トラフィック シェーピング 61
キューイングおよびスケジューリング 61
帯域幅 63
帯域幅の割合 63
残存帯域幅の割合 63
重み付けテール ドロップ 64
重み付けテール ドロップのデフォルト値 64
プライオリティ キュー 65
プライオリティ キュー ポリサー 66
キュー バッファ 66
キュー バッファの割り当て 68
ダイナミックなしきい値および拡張 68
統合バッファ共有 69

重み付けランダム早期検出 69
信頼動作 69
Cisco IP Phone の信頼境界機能のポート セキュリティ 69
有線ポートの信頼動作 70
標準 QoS のデフォルト設定 71
デフォルトの有線 QoS 設定 71
DSCP マップ 71
QoS の設定方法 73
クラス、ポリシー、およびマップの設定 73
トラフィック クラスの作成 73
トラフィック ポリシーの作成 76
クラスベース パケット マーキングの設定 80
トラフィック ポリシーのインターフェイスへの適用 85
ポリシー マップによる物理ポートのトラフィックの分類、ポリシング、およびマーキング 88
ポリシーマップによるトラフィックの分類およびマーキング 92
テーブル マップの設定 95
有線ターゲットの QoS に関する制約事項 98
QoS の特性と機能の設定 101
帯域幅の設定 101
ポリシングの設定 103
プライオリティの設定 106
SGT ベースの QoS の設定 109
キューとシェーピングの設定 111
出力キューの特性の設定 111
キュー バッファの設定 111
キュー制限の設定 115
シェーピングの設定 118
シャープ プロファイル キューイングの設定 119
QoS のモニタリング 122
QoS の設定例 122

例：TCP プロトコル分類 122
例：UDP プロトコル分類 123
例：RTP プロトコル分類 124
例：アクセス コントロール リストによる分類 125
例：サービス クラス レイヤ 2 の分類 125
例：サービス クラス DSCP の分類 125
例：VLAN ID レイヤ 2 の分類 125
例：DSCP 値または precedence 値による分類 126
例：階層型分類 126
例：階層型ポリシーの設定 126
例：音声およびビデオの分類 127
例：平均レート シェーピングの設定 129
例：キュー制限の設定 129
例：キュー バッファの設定 130
例：ポリシング アクションの設定 131
例：ポリサーの VLAN 設定 131
例：ポリシングの単位 132
例：シングルレート 2 カラー ポリシング設定 132
例：デュアルレート 3 カラー ポリシング設定 132
例：テーブル マップのマーキング設定 133
例：CoS マーキングを保持するテーブル マップの設定 134
次の作業 134
QoS に関する追加情報 134
QoS の機能履歴 135

第 3 章 **重み付けランダム早期検出の設定 137**
ネットワーク輻輳の回避 137
テール ドロップ 137
重み付けランダム早期検出 138
WRED の仕組み 138
WRED 重み計算 138

WRED 設定の制限 139
WRED 使用上の注意事項 139
WRED の設定 140
DSCP 値に基づく WRED の設定 140
サービス クラス値に基づく WRED の設定 141
IP プレシデンス値に基づく WRED の設定 143
WRED の設定例 144
階層化 QoS を使用した WRED のサポート 144
WRED 設定の確認 145
WRED 設定のベスト プラクティス 146
重み付けランダム早期検出の機能履歴 148



第 1 章

自動 QoS の設定

- 自動 QoS の前提条件 （1 ページ）
- 自動 QoS の制約事項 （1 ページ）
- 自動 QoS の設定に関する情報 （2 ページ）
- 自動 QoS の設定方法 （5 ページ）
- 自動 QoS の監視 （11 ページ）
- 自動 QoS に関するトラブルシューティング （11 ページ）
- 自動 QoS の設定例 （12 ページ）
- 自動 QoS の関連情報 （32 ページ）
- 自動 QoS の機能履歴 （33 ページ）

自動 QoS の前提条件

自動 QoS の前提条件は標準 QoS の前提条件と同じです。

自動 QoS の制約事項

次に、自動 QoS の制約事項を示します。

- 自動 QoS は、SVI インターフェイスではサポートされません。
- ビデオをサポートしている IP フォンには、 **auto qos voip cisco-phone** オプションを設定しないでください。ビデオパケットには Expedited Forwarding（EF; 完全優先転送）プライオリティが設定されていないため、このオプションを使用すると、ビデオパケットの DSCP マーキングが上書きされ、これらのパケットが class-default クラスに分類されます。
- 自動 QoS が **auto qos voip cisco-phone** コマンドを使用するスタートアップ コンフィギュレーションから実行コンフィギュレーションにプッシュされた場合、自動 QoS によって設定は生成されません。これは予期された動作であり、これにより、 **auto qos voip cisco-phone** コマンドがスタートアップ コンフィギュレーションからプッシュされるたびに、ユーザーが作成したカスタマイズ済みの QoS ポリシーがデフォルト設定（ある場合）で上書きされないようにします。

この制限に対し、次のいずれかの回避策を使用できます。

- スイッチのインターフェイスで **auto qos voip cisco-phone** コマンドを手動で設定します。
- 新しいスイッチでは、スタートアップ コンフィギュレーションから自動 QoS コマンドをプッシュする場合は、コマンドに標準テンプレートの一部として次の項目をそれぞれ含める必要があります。
  1. インターフェイス レベル：
     - **trust device cisco-phone**
     - **auto qos voip cisco-phone**
     - **service-policy input** AutoQos-4.0-CiscoPhone-Input-Policy
     - **service-policy output** AutoQos-4.0-Output-Policy
  2. グローバル レベル：
     - クラスマップ
     - ポリシーマップ
     - ACL（ACE）
- **auto qos voip cisco-phone** コマンドがインターフェイスですでに設定されているが、ポリシーが生成されていない場合は、すべてのインターフェイスからコマンドを無効にして、各インターフェイスでコマンドを手動で再設定します。

自動 QoS の設定に関する情報

自動 QoS の概要

自動 QoS 機能を使用して、QoS 機能の配置を容易にできます。自動 QoS は、ネットワーク設計を確認し、スイッチがさまざまなトラフィック フローに優先度を指定できるように QoS 設定をイネーブルにします。

スイッチは MQC モデルを採用しています。これは、特定のグローバル コンフィギュレーションを使用する代わりに、スイッチ上のインターフェイスに適用された自動 QoS が複数のグローバル クラス マップとポリシー マップを設定することを意味します。

自動 QoS はトラフィックを照合し、各一致パケットを qos-group に割り当てます。これにより、出力ポリシー マップは、プライオリティ キューを含む特定のキューに、特定の qos-group を配置できます。

QoS は、着信と発信の両方向で必要です。着信時に、スイッチ ポートは、パケットの DSCP を信頼する必要があります（デフォルトで実行されます）。発信時に、スイッチ ポートは、音

声パケットに「front of line」プライオリティを付与する必要があります。音声が発信キューの他のパケットの後ろで待機して、遅延が長くなりすぎる場合、パケットの受信時間の範囲外となるため、エンド ホストは、そのパケットをドロップします。

自動 QoS 短縮機能の概要

自動QoSコマンドを入力すると、CLIからコマンドを入力する場合と同様に、生成されたすべてのコマンドがスイッチにより表示されます。自動 QoS 短縮機能を使用して、実行コンフィギュレーションから自動 QoS が生成したコマンドを非表示にできます。これにより、実行コンフィギュレーションを容易に把握でき、またメモリをより効率的に使用できるようになります。

自動 QoS グローバル設定テンプレート

一般に、自動 QoS コマンドは、ACL または DSCP で一致する、またはアプリケーション クラスに送信されるトラフィックを識別するCoS値で一致する一連のクラスマップを生成します。また、生成されたクラスに一致する入力ポリシーや、設定されている帯域幅にクラスをポリシングする入力ポリシーも生成されます。8つの出力キュー クラスマップが生成されます。実際の出力の出力ポリシーは、この 8 つの出力キュー クラス マップのそれぞれにキューを割り当てます。

自動 QoS コマンドは、必要なテンプレートだけを生成します。たとえば、新しい自動 QoS コマンドを初めて使用するときに、8つのキュー出力サービス ポリシーを定義するグローバル設定が生成されます。この時点から、他のインターフェイスに適用された自動QoSコマンドは、出力キューのテンプレートを生成しません。これは、新しい自動 QoS コマンドが最初に使用されてから生成された同じ 8 つのキュー モデルに、すべての自動 QoS コマンドが依存しているためです。

自動 QoS ポリシーとクラス マップ

適切な自動 QoS コマンドを入力すると、次のアクションが実行されます。

- 特定のクラス マップが作成されます。
- 特定のポリシー マップ（入力および出力）が作成されます。
- 指定したインターフェイスにポリシー マップが適用されます。
- インターフェイスの信頼レベルが設定されます。

実行コンフィギュレーションでの自動 QoS の影響

自動 QoS がイネーブルになると、**auto qos** インターフェイス コンフィギュレーション コマンドおよび生成されたグローバル コンフィギュレーションが実行コンフィギュレーションに追加されます。

スイッチは、自動 QoS が生成したコマンドを、CLI から入力したように適用します。既存のユーザー設定では、生成されたコマンドの適用に失敗することがあります。また、生成されたコマンドで既存の設定が上書きされることもあります。これらのアクションが警告なしで発生する可能性があります。生成されたコマンドがすべて正常に適用された場合、上書きされなかったユーザー入力の設定は実行コンフィギュレーション内に残ります。上書きされたユーザー入力の設定は、現在の設定をメモリに保存せずに、スイッチをリロードすると復元できます。生成コマンドが適用されなかった場合、以前の実行コンフィギュレーションが復元されます。

実行コンフィギュレーションでの自動 QoS 短縮機能の影響

自動 QoS 短縮機能をイネーブルにした場合：

- CLI から入力された自動 QoS コマンドだけが実行コンフィギュレーションに表示されます。
- 生成されるグローバル コンフィギュレーションおよびインターフェイス コンフィギュレーションは表示されません。
- コンフィギュレーションを保存するときに、入力した自動 QoS コマンドだけが保存されます（非表示のコンフィギュレーションは保存されません）。
- スイッチをリロードすると、保存された自動 QoS コマンドがシステムにより検出、再実行され、AutoQoS SRND4.0 に準拠したコンフィギュレーション セットが生成されます。

（注） 自動 QoS 短縮機能がイネーブルである場合は、自動 QoS 生成コマンドを変更しないでください。これは、スイッチのリロード時にユーザー変更がオーバーライドされるためです。

自動 QoS グローバル短縮機能をイネーブルにした場合：

- **show derived-config** 非表示の AQC 派生コマンドを表示するには、コマンドを使用します。
- AQC コマンドはメモリに保存されません。これらは、スイッチがリロードされるたびに再生成されます。
- 短縮機能がイネーブルである場合、自動 QoS により生成されたコマンドは変更しないでください。
- 自動 QoS でインターフェイスが設定されており、AQC をディセーブルにする必要がある場合は、最初に自動 QoS をインターフェイス レベルでディセーブルにする必要があります。

自動 QoS の設定方法

自動 QoS の設定

QoS パフォーマンスを最適化するには、ネットワーク内のすべてのデバイスで自動 QoS を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface***interface-id* 例： `Device(config)# interface HundredGigE 1/0/1`	VoIP ポートやビデオ デバイスに接続されているポート、またはネットワーク内部の他の信頼できるスイッチまたはルータに接続されているアップリンク ポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	自動 QoS 設定によって、次のコマンドの 1 つを使用します。 • **auto qos voip** {**cisco-phone** | **cisco-softphone** | **trust**} • **auto qos video** {**cts** | **ip-camera** | **media-player**} • **auto qos classify** [**police**] • **auto qos trust** {**cos** | **dscp**} 例：	次のコマンドによって、VoIP 用の自動 QoS が有効になります。 • **auto qos voip cisco-phone**：ポートが Cisco IP Phone に接続されている場合、着信パケットの QoS ラベルは電話機が検出された場合だけ信頼されます（CDP を介して条件付き信頼）。

自動 QoS の設定

	コマンドまたはアクション	目的
	`Device(config-if)# auto qos trust dscp`	（注） ビデオをサポートしている IP フォンには、**auto qos voip cisco-phone** オプションを設定しないでください。ビデオ パケットには Expedited Forwarding（EF;完全優先転送）プライオリティが設定されていないため、このオプションを使用すると、ビデオ パケットの DSCP マーキングが上書きされ、これらのパケットが class-default クラスに分類されます。 • **auto qos voip cisco-softphone**：ポートが Cisco SoftPhone 機能を実行するデバイスに接続されています。このコマンドによって Cisco IP SoftPhone アプリケーションおよびマーキングを実行する PC に接続しているインターフェイスの QoS 設定が生成され、そのようなインターフェイスからのトラフィックをマーキングおよびポリシングします。このコマンドで設定されたポートは、信頼できないと見なされます。 • **auto qos voip trust**：アップリンク ポートが信頼性のあるスイッチまたはルータに接続されていて、入力パケットの VoIP トラフィック分類が信頼されています。 次のコマンドは、指定されたビデオ デバイス（システム、カメラ、メディアプレーヤー）用の自動 QoS を有効にします。 • **auto qos video cts**：Cisco Telepresence System に接続されているポート。着信パケットの QoS ラベルは Cisco TelePresence が検出された場合だけ

	コマンドまたはアクション	目的
		信頼されます（CDP を介した条件付き信頼） • **auto qos video ip-camera**：Cisco ビデオ監視カメラに接続されているポート。着信パケットの QoS ラベルは Cisco カメラが検出された場合だけ信頼されます（CDP を介した条件付き信頼） • **auto qos video media-player**：CDP 対応 Cisco Digital Media Player に接続されているポート。着信パケットの QoS ラベルはデジタル メディア プレイヤーが検出された場合だけ信頼されます（CDP を介した条件付き信頼）。 次のコマンドは、分類の自動 QoS を有効にします。 • **auto qos classify police**：このコマンドは、信頼できないインターフェイスの QoS 設定を生成します。この設定では、信頼できないデスクトップ/デバイスから着信するトラフィックを分類してマークするため、サービス ポリシーがインターフェイスに適用されます。生成されたサービス ポリシーは、ポリシングを実行します。 次のコマンドによって、信頼できるインターフェイス用の自動 QoS が有効になります。 • **auto qos trust cos**：サービス クラス • **auto qos trust dscp**：DiffServ コードポイント。
ステップ 4	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 5	**show auto qos interface***interface-id* 例： `Device# show auto qos interface HundredGigE 1/0/1`	（任意）自動 QoS がイネーブルであるインターフェイス上の自動 QoS コマンドを表示します。自動 QoS 設定およびユーザー変更を表示する場合は、**show running-config** コマンドを使用します。

自動 QoS のアップグレード

始める前に

アップグレードを行う前に、スイッチ上のすべての自動QoS設定を削除する必要があります。この例では、その手順について説明します。

この例の手順を実行した後で、新しいソフトウェアイメージまたはアップグレード後のソフトウェア イメージのスイッチをリブートし、自動 QoS を再設定する必要があります。

手順

ステップ 1 **show auto qos**

例：

```
Device# show auto qos

TwentyFiveGigE1/0/1
auto qos trust dscp

TwentyFiveGigE1/0/2
auto qos trust cos
```

特権 EXEC モードでこのコマンドを入力して、現在の自動 QoS 設定をすべて記録します。

ステップ 2 **no auto qos**

例：

```
Device(config-if)#no auto qos
```

インターフェイス コンフィギュレーション モードで、自動 QoS 設定が行われている各インターフェイスで適切な **no auto qos** コマンドを実行します。

ステップ 3 **show running-config | i autoQos**

例：

```
Device# show running-config | i autoQos
```

特権EXECモードに戻り、このコマンドを入力して、残りの自動QoSマップ、クラスマップ、ポリシー マップ、アクセス リスト、テーブル マップ、またはその他の設定を記録します。

ステップ 4 **no policy-map** *policy-map_name*

例：

```
Device(config)# no policy-map pmap_101
Device(config)# no class-map cmap_101
Device(config)# no ip access-list extended AutoQos-101
Device(config)# no table-map 101
Device(config)# no table-map policed-dscp
```

グローバル コンフィギュレーション モードでこのコマンドを入力して、QoS クラス マップ、ポリシー マップ、アクセス リスト、テーブル マップ、およびその他の自動 QoS 設定を削除します。

- **no policy-map** *policy-map-name*
- **no class-map** *class-map-name*
- **no ip access-list extended** *Auto-QoS-x*
- **no table-map** *table-map-name*
- **no table-map policed-dscp**

ステップ 5 **show running-config | i autoQos**

例：

```
Device#show running-config | i autoQos
```

特権 EXEC モードに戻り、このコマンドを実行して、自動 QoS 設定がないこと、または自動 QoS 設定の残りの部分がないことを確認します。

ステップ 6 **show auto qos**

例：

```
Device# show auto qos
```

このコマンドを実行して、自動 QoS 設定がないこと、または設定の残りの部分がないことを確認します。

ステップ 7 **write memory**

例：

```
Device# write memory
```

write memory コマンドを入力して、自動QoS設定に対する変更をNVメモリに書き込みます。

次のタスク

新しいソフトウェア イメージまたはアップグレード後のソフトウェア イメージでスイッチをリブートします。

新しいソフトウェアイメージまたはアップグレード後のソフトウェアイメージでリブートしたら、ステップ 1 で説明した **show auto qos** コマンドを実行した結果に基づいて、適切なスイッチインターフェイスの自動 QoS を再設定します。

（注） スイッチまたはスタックごとに、マークダウンの超過用に1つのテーブルマップ、マークダウンの違反用に1つのテーブルマップが存在します。超過アクションのテーブルマップがスイッチにすでに存在している場合は、自動 QoS ポリシーを適用できません。

自動 QoS 短縮機能のイネーブル化

自動 QoS 短縮機能をイネーブルにするには、次のコマンドを入力します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**auto qos global compact** 例： `Device(config)# auto qos global compact`	自動QoS短縮機能がイネーブルになり、自動 QoS のグローバル コンフィギュレーション（非表示）が生成されます。 その後、インターフェイス コンフィギュレーション モードで設定する自動 QoS コマンドを入力できます。システムにより生成されるインターフェイス コマンドも非表示になります。 適用された自動 QoS 設定を表示するには、次の特権 EXEC コマンドを使用します。 • **show derived-config** • **show policy-map** • **show access-list** • **show class-map**

	コマンドまたはアクション	目的
		• **show table-map** • **show auto qos** • **show policy-map interface** • **show ip access-lists** これらのコマンドにはキーワード「**AutoQos-**」が含まれます。

次のタスク

自動 QoS 短縮機能をディセーブルにするには、対応する自動 QoS コマンドの **no** 形式を入力して自動 QoS インスタンスをすべてのインターフェイスから削除し、次に **no auto qos global compact** グローバル コンフィギュレーション コマンドを実行します。

自動 QoS の監視

表 *1:* 自動 *QoS* の監視用コマンド

コマンド	説明
show auto qos [**interface** [*interface-id*]]	最初の自動 QoS 設定を表示します。 **show auto qos** コマンド出力と **show running-config** コマンド出力を比較してユーザー定義の QoS 設定を比較できます。
show running-config	自動 QoS によって影響されるかもしれない QoS 設定に関する情報を表示します。 **show auto qos** コマンド出力と **show running-config** コマンド出力を比較してユーザー定義の QoS 設定を比較できます。
show derived-config	自動qosテンプレートにより実行コンフィギュレーションとともに設定される非表示の **mls qos** コマンドを表示します。

自動 QoS に関するトラブルシューティング

自動 QoS のトラブルシューティングを行うには、**debug auto qos** 特権 EXEC コマンドを使用します。詳細については、このリリースに対応するコマンドリファレンスにある **debug auto qos** コマンドを参照してください。

ポートで自動 QoS を無効にするには、**auto qos** インターフェイス コンフィギュレーション コマンドの **no** 形式（**no auto qos voip** など）を使用します。このポートに対して、auto-QoS によって生成されたインターフェイス コンフィギュレーション コマンドだけが削除されます。auto-QoS をイネーブルにした最後のポートで、**no auto qos voip** コマンドを入力すると、auto-QoS によって生成されたグローバル コンフィギュレーション コマンドが残っている場合でも、auto-QoS はディセーブルと見なされます（グローバル コンフィギュレーションによって影響を受ける他のポートでのトラフィックの中断を避けるため）。

自動 QoS の設定例

例：auto qos trust cos

次に、**auto qos trust cos** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-Trust-Cos-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigE1/0/2
Device(config-if)# auto qos trust cos
Device(config-if)# end
Device# show policy-map interface HundredGigE1/0/2

HundredGigE1/0/2

  Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy

  Class-map: class-default (match-any)
  0 packets
  Match: any
  QoS Set
  dscp dscp table AutoQos-4.0-Trust-Dscp-Table
```

```
Service-policy output: AutoQos-4.0-Output-Policy

  queue stats for all priority classes:
  Queueing
  priority level 1

  (total drops) 0
  (bytes output) 0

Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
  0 packets
  Match: dscp cs4 (32) cs5 (40) ef (46)
  Match: cos 5
  Priority: 30% (30000000 kbps), burst bytes 750000000,

  Priority Level: 1

Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
  0 packets
  Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
  Match: cos 3
  Queueing

  queue-limit dscp 16 percent 80
  queue-limit dscp 24 percent 90
  queue-limit dscp 48 percent 100
  queue-limit dscp 56 percent 100
  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%

  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
  0 packets
  Match: dscp af41 (34) af42 (36) af43 (38)
  Match: cos 4
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
  0 packets
  Match: dscp af21 (18) af22 (20) af23 (22)
  Match: cos 2
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
  0 packets
  Match: dscp af11 (10) af12 (12) af13 (14)
  Match: cos 1
  Queueing

  (total drops) 0
  (bytes output) 0
```

```
  bandwidth remaining 4%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  0 packets
  Match: dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 1%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  0 packets
  Match: dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  0 packets
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 25%
  queue-buffers ratio 25
```

例：auto qos trust dscp

次に、**auto qos trust dscp** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-Trust-Dscp-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）

- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigE1/0/2
Device(config-if)# auto qos trust dscp
Device(config-if)# end
Device#show policy-map interface HundredGigE1/0/2

HundredGigE1/0/2

  Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy

    Class-map: class-default (match-any)
      0 packets
      Match: any
      QoS Set
        dscp dscp table AutoQos-4.0-Trust-Dscp-Table

  Service-policy output: AutoQos-4.0-Output-Policy

    queue stats for all priority classes:
      Queueing
      priority level 1

      (total drops) 0
      (bytes output) 0

    Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
      0 packets
      Match:  dscp cs4 (32) cs5 (40) ef (46)
      Match: cos  5
      Priority: 30% (30000000 kbps), burst bytes 750000000,

      Priority Level: 1

    Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
      0 packets
      Match:  dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
      Match: cos  3
      Queueing
      queue-limit dscp 16 percent 80
      queue-limit dscp 24 percent 90
      queue-limit dscp 48 percent 100
      queue-limit dscp 56 percent 100
      (total drops) 0
      (bytes output) 0
      bandwidth remaining 10%

      queue-buffers ratio 10

    Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
      0 packets
      Match:  dscp af41 (34) af42 (36) af43 (38)
      Match: cos  4
      Queueing

      (total drops) 0
      (bytes output) 0
      bandwidth remaining 10%
      queue-buffers ratio 10

    Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
      0 packets
```

```
  Match:  dscp af21 (18) af22 (20) af23 (22)
  Match: cos  2
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
  0 packets
  Match:  dscp af11 (10) af12 (12) af13 (14)
  Match: cos  1
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 4%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  0 packets
  Match:  dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 1%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  0 packets
  Match:  dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  0 packets
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 25%
  queue-buffers ratio 25
```

例：auto qos video cts

次に、**auto qos video cts** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-Trust-Cos-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigabitEthernet1/0/2
Device(config-if)# auto qos video cts
Device(config-if)# end
Device# show policy-map interface HundredGigabitEthernet1/0/2

HundredGigabitEthernet1/0/2

  Service-policy input: AutoQos-4.0-Trust-Cos-Input-Policy

    Class-map: class-default (match-any)
      Match: any
      QoS Set
        cos cos table AutoQos-4.0-Trust-Cos-Table

  Service-policy output: AutoQos-4.0-Output-Policy

    queue stats for all priority classes:
      Queueing
      priority level 1

      (total drops) 0
      (bytes output) 0

    Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
      Match:  dscp cs4 (32) cs5 (40) ef (46)
      Match: cos  5
      Priority: 30% (300000 kbps), burst bytes 7500000,

      Priority Level: 1

    Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
      Match:  dscp cs3 (24) cs6 (48) cs7 (56)
      Match: cos  3
      Queueing
      queue-limit dscp 16 percent 80
      queue-limit dscp 24 percent 90
      queue-limit dscp 48 percent 100

      (total drops) 0
      (bytes output) 0
      bandwidth remaining 10%
```

```
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
  Match:  dscp af41 (34) af42 (36) af43 (38)
  Match: cos  4
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
  Match:  dscp af21 (18) af22 (20) af23 (22)
  Match: cos  2
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
  Match:  dscp af11 (10) af12 (12) af13 (14)
  Match: cos  1
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 4%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  Match:  dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 1%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  Match:  dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 25%
  queue-buffers ratio 25
```

例：auto qos video ip-camera

次に、**auto qos video ip-camera** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-Trust-Dscp-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigabitE1/0/2
Device(config-if)# auto qos video ip-camera
Device(config-if)# end
Device# show policy-map interface HundredGigabitE1/0/2

HundredGigabitE1/0/2

  Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy

    Class-map: class-default (match-any)
      Match: any
      QoS Set
        dscp dscp table AutoQos-4.0-Trust-Dscp-Table

  Service-policy output: AutoQos-4.0-Output-Policy

    queue stats for all priority classes:
      Queueing
      priority level 1

      (total drops) 0
      (bytes output) 0

    Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
      Match:  dscp cs4 (32) cs5 (40) ef (46)
      Match: cos  5
      Priority: 30% (300000 kbps), burst bytes 7500000,

      Priority Level: 1
```

例：**auto qos video ip-camera**

```
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
  Match:  dscp cs3 (24) cs6 (48) cs7 (56)
  Match: cos  3
  Queueing
  queue-limit dscp 16 percent 80
  queue-limit dscp 24 percent 90
  queue-limit dscp 48 percent 100

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%

  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
  Match:  dscp af41 (34) af42 (36) af43 (38)
  Match: cos  4
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
  Match:  dscp af21 (18) af22 (20) af23 (22)
  Match: cos  2
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
  Match:  dscp af11 (10) af12 (12) af13 (14)
  Match: cos  1
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 4%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  Match:  dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 1%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  Match:  dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  Match: any
```

```
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 25%
    queue-buffers ratio 25
```

例：auto qos video media-player

次に、**auto qos video media-player** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-Trust-Dscp-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigabitE1/0/2
Device(config-if)# auto qos video media-player
Device(config-if)# end
Device# show policy-map interface HundredGigabitE1/0/2

HundredGigabitE1/0/2

  Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy

    Class-map: class-default (match-any)
      Match: any
      QoS Set
        dscp dscp table AutoQos-4.0-Trust-Dscp-Table

  Service-policy output: AutoQos-4.0-Output-Policy

    queue stats for all priority classes:
      Queueing
```

```
  priority level 1

  (total drops) 0
  (bytes output) 0

Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
  Match:  dscp cs4 (32) cs5 (40) ef (46)
  Match: cos  5
  Priority: 30% (300000 kbps), burst bytes 7500000,

  Priority Level: 1

Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
  Match:  dscp cs3 (24) cs6 (48) cs7 (56)
  Match: cos  3
  Queueing
  queue-limit dscp 16 percent 80
  queue-limit dscp 24 percent 90
  queue-limit dscp 48 percent 100

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%

  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
  Match:  dscp af41 (34) af42 (36) af43 (38)
  Match: cos  4
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
  Match:  dscp af21 (18) af22 (20) af23 (22)
  Match: cos  2
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
  Match:  dscp af11 (10) af12 (12) af13 (14)
  Match: cos  1
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 4%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  Match:  dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 1%
  queue-buffers ratio 10
```

```
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  Match:  dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 25%
  queue-buffers ratio 25
```

例：auto qos voip trust

次に、**auto qos voip trust** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-Trust-Cos-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigabitE1/0/3
Device(config-if)# auto qos voip trust
Device(config-if)# end
Device# show policy-map interface HundredGigabitE1/0/3

HundredGigabitE1/0/3

  Service-policy input: AutoQos-4.0-Trust-Cos-Input-Policy
```

```
  Class-map: class-default (match-any)
    Match: any
    QoS Set
      cos cos table AutoQos-4.0-Trust-Cos-Table

Service-policy output: AutoQos-4.0-Output-Policy

  queue stats for all priority classes:
    Queueing
    priority level 1

    (total drops) 0
    (bytes output) 0

  Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
    Match:  dscp cs4 (32) cs5 (40) ef (46)
    Match: cos  5
    Priority: 30% (300000 kbps), burst bytes 7500000,

    Priority Level: 1

  Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
    Match:  dscp cs3 (24) cs6 (48) cs7 (56)
    Match: cos  3
    Queueing
    queue-limit dscp 16 percent 80
    queue-limit dscp 24 percent 90
    queue-limit dscp 48 percent 100

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%

    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
    Match:  dscp af41 (34) af42 (36) af43 (38)
    Match: cos  4
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%
    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
    Match:  dscp af21 (18) af22 (20) af23 (22)
    Match: cos  2
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%
    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
    Match:  dscp af11 (10) af12 (12) af13 (14)
    Match: cos  1
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 4%
    queue-buffers ratio 10
```

```
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  Match:  dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 1%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  Match:  dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 25%
  queue-buffers ratio 25
```

例：auto qos voip cisco-phone

次に、**auto qos voip cisco-phone** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-CiscoPhone-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- AutoQos-4.0-Voip-Data-Class（match-any）
- AutoQos-4.0-Voip-Signal-Class（match-any）
- AutoQos-4.0-Default-Class（match-any）
- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）

- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
(config)# interface gigabitEthernet1/0/5
(config-if)# auto qos voip cisco-phone
(config-if)# end
# show policy-map interface gigabitEthernet1/0/5

 GigabitEthernet1/0/5

  Service-policy input: AutoQos-4.0-CiscoPhone-Input-Policy

    Class-map: AutoQos-4.0-Voip-Data-Class (match-any)
      Match: ip dscp ef (46)
      QoS Set
        ip dscp ef
      police:
          cir 128000 bps, bc 8000 bytes, be 8000 bytes
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          set-dscp-transmit dscp table policed-dscp
        violated 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: AutoQos-4.0-Voip-Signal-Class (match-any)
      Match: ip dscp cs3 (24)
      QoS Set
        ip dscp cs3
      police:
          cir 32000 bps, bc 8000 bytes, be 8000 bytes
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          set-dscp-transmit dscp table policed-dscp
        violated 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: AutoQos-4.0-Default-Class (match-any)
      Match: access-group name AutoQos-4.0-Acl-Default
      QoS Set
        dscp default
      police:
          cir 10000000 bps, bc 8000 bytes, be 8000 bytes
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          set-dscp-transmit dscp table policed-dscp
        violated 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: class-default (match-any)
      Match: any

  Service-policy output: AutoQos-4.0-Output-Policy
```

```
queue stats for all priority classes:
  Queueing
  priority level 1

  (total drops) 0
  (bytes output) 0

Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
  Match:  dscp cs4 (32) cs5 (40) ef (46)
  Match: cos  5
  Priority: 30% (300000 kbps), burst bytes 7500000,

  Priority Level: 1

Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
  Match:  dscp cs3 (24) cs6 (48) cs7 (56)
  Match: cos  3
  Queueing
  queue-limit dscp 16 percent 80
  queue-limit dscp 24 percent 90
  queue-limit dscp 48 percent 100

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%

  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
  Match:  dscp af41 (34) af42 (36) af43 (38)
  Match: cos  4
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
  Match:  dscp af21 (18) af22 (20) af23 (22)
  Match: cos  2
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
  Match:  dscp af11 (10) af12 (12) af13 (14)
  Match: cos  1
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 4%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
  Match:  dscp cs1 (8)
  Queueing

  (total drops) 0
  (bytes output) 0
```

```
  bandwidth remaining 1%
  queue-buffers ratio 10

Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
  Match:  dscp af31 (26) af32 (28) af33 (30)
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 10%
  queue-buffers ratio 10

Class-map: class-default (match-any)
  Match: any
  Queueing

  (total drops) 0
  (bytes output) 0
  bandwidth remaining 25%
  queue-buffers ratio 25
```

例：auto qos voip cisco-softphone

次に、**auto qos voip cisco-softphone** コマンドと、適用されるポリシーとクラスマップの例を示します。

このコマンドを実行すると、次のポリシー マップが作成されて適用されます。

- AutoQos-4.0-CiscoSoftPhone-Input-Policy
- AutoQos-4.0-Output-Policy

このコマンドを実行すると、次のクラス マップが作成されて適用されます。

- AutoQos-4.0-Voip-Data-Class（match-any）
- AutoQos-4.0-Voip-Signal-Class（match-any）
- AutoQos-4.0-Multimedia-Conf-Class（match-any）
- AutoQos-4.0-Bulk-Data-Class（match-any）
- AutoQos-4.0-Transaction-Class（match-any）
- AutoQos-4.0-Scavanger-Class（match-any）
- AutoQos-4.0-Signaling-Class（match-any）
- AutoQos-4.0-Default-Class（match-any）
- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）

- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface HundredGigE1/0/20
Device(config-if)# auto qos voip cisco-softphone
Device(config-if)# end
Device# show policy-map interface HundredGigE1/0/20

 HundredGigE1/0/20

  Service-policy input: AutoQos-4.0-CiscoSoftPhone-Input-Policy

    Class-map: AutoQos-4.0-Voip-Data-Class (match-any)
      Match: ip dscp ef (46)
      QoS Set
        ip dscp ef
      police:
          cir 128000 bps, bc 8000 bytes, be 8000 bytes
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          set-dscp-transmit dscp table policed-dscp
        violated 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: AutoQos-4.0-Voip-Signal-Class (match-any)
      Match: ip dscp cs3 (24)
      QoS Set
        ip dscp cs3
      police:
          cir 32000 bps, bc 8000 bytes, be 8000 bytes
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          set-dscp-transmit dscp table policed-dscp
        violated 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: AutoQos-4.0-Multimedia-Conf-Class (match-any)
      Match: access-group name AutoQos-4.0-Acl-MultiEnhanced-Conf
      QoS Set
        dscp af41
      police:
          cir 5000000 bps, bc 8000 bytes, be 8000 bytes
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          set-dscp-transmit dscp table policed-dscp
        violated 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: AutoQos-4.0-Bulk-Data-Class (match-any)
      Match: access-group name AutoQos-4.0-Acl-Bulk-Data
      QoS Set
```

```
    dscp af11
  police:
      cir 10000000 bps, bc 8000 bytes, be 8000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      set-dscp-transmit dscp table policed-dscp
    violated 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps, violate 0000 bps

Class-map: AutoQos-4.0-Transaction-Class (match-any)
  Match: access-group name AutoQos-4.0-Acl-Transactional-Data
  QoS Set
    dscp af21
  police:
      cir 10000000 bps, bc 8000 bytes, be 8000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      set-dscp-transmit dscp table policed-dscp
    violated 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps, violate 0000 bps

Class-map: AutoQos-4.0-Scavanger-Class (match-any)
  Match: access-group name AutoQos-4.0-Acl-Scavanger
  QoS Set
    dscp cs1
  police:
      cir 10000000 bps, bc 8000 bytes, be 8000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      set-dscp-transmit dscp table policed-dscp
    violated 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps, violate 0000 bps

Class-map: AutoQos-4.0-Signaling-Class (match-any)
  Match: access-group name AutoQos-4.0-Acl-Signaling
  QoS Set
    dscp cs3
  police:
      cir 32000 bps, bc 8000 bytes, be 8000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      set-dscp-transmit dscp table policed-dscp
    violated 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps, violate 0000 bps

Class-map: AutoQos-4.0-Default-Class (match-any)
  Match: access-group name AutoQos-4.0-Acl-Default
  QoS Set
    dscp default
  police:
      cir 10000000 bps, bc 8000 bytes, be 8000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      set-dscp-transmit dscp table policed-dscp
    violated 0 bytes; actions:
```

```
        drop
      conformed 0000 bps, exceed 0000 bps, violate 0000 bps

  Class-map: class-default (match-any)
    Match: any

Service-policy output: AutoQos-4.0-Output-Policy

  queue stats for all priority classes:
    Queueing
    priority level 1

    (total drops) 0
    (bytes output) 0

  Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
    Match:  dscp cs4 (32) cs5 (40) ef (46)
    Match: cos  5
    Priority: 30% (300000 kbps), burst bytes 7500000,

    Priority Level: 1

  Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
    Match:  dscp cs3 (24) cs6 (48) cs7 (56)
    Match: cos  3
    Queueing
    queue-limit dscp 16 percent 80
    queue-limit dscp 24 percent 90
    queue-limit dscp 48 percent 100

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%

    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
    Match:  dscp af41 (34) af42 (36) af43 (38)
    Match: cos  4
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%
    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
    Match:  dscp af21 (18) af22 (20) af23 (22)
    Match: cos  2
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%
    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
    Match:  dscp af11 (10) af12 (12) af13 (14)
    Match: cos  1
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 4%
```

```
    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
    Match:  dscp cs1 (8)
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 1%
    queue-buffers ratio 10

  Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
    Match:  dscp af31 (26) af32 (28) af33 (30)
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 10%
    queue-buffers ratio 10

  Class-map: class-default (match-any)
    Match: any
    Queueing

    (total drops) 0
    (bytes output) 0
    bandwidth remaining 25%
    queue-buffers ratio 25
```

例：auto qos global compact

次に、**auto qos global compact** コマンドの例を示します。

```
Device# configure terminal
Device(config)# auto qos global compact
Device(config)# interface HundredGigE1/0/2
Device(config-if)# auto qos voip cisco-phone

Device# show auto qos
HundredGigE1/0/2
auto qos voip cisco-phone

Device# show running-config interface HundredGigE1/0/2
interface HundredGigE1/0/2
auto qos voip cisco-phone
end
```

自動 QoS の関連情報

自動 QoS 設定で特定の QoS の変更をする必要がある場合は、QoS のマニュアルを確認してください。

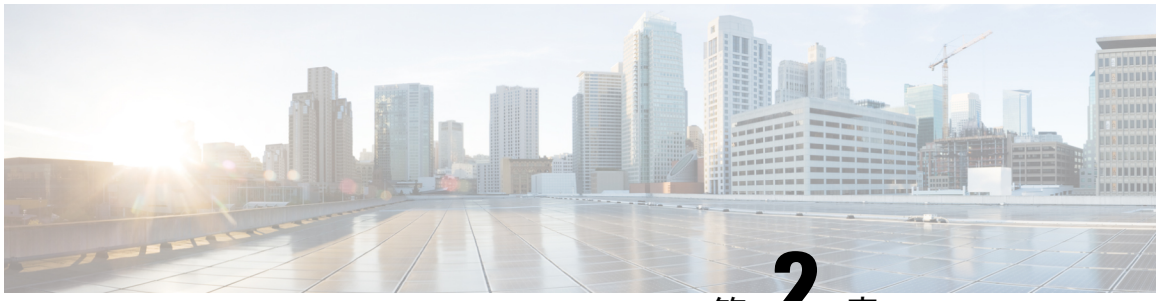
自動 QoS の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	自動 QoS	自動 QoS 機能により、QoS 機能の導入が簡素化されます。この機能は、ネットワーク設計を確認し、スイッチがさまざまなトラフィックフローに優先度を指定できるように QoS 設定をイネーブルにします。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 2 章

QoS の設定

- QoS の前提条件 （35 ページ）
- QoS コンポーネント （36 ページ）
- QoS の用語 （37 ページ）
- QoS の概要 （37 ページ）
- QoS の実装 （39 ページ）
- QoS 有線モデル （44 ページ）
- 分類 （45 ページ）
- 入力ポート FIFO パーサー （53 ページ）
- ポリシング （55 ページ）
- マーキング （56 ページ）
- トラフィックの調整 （58 ページ）
- キューイングおよびスケジューリング （61 ページ）
- 信頼動作 （69 ページ）
- 標準 QoS のデフォルト設定 （71 ページ）
- QoS の設定方法 （73 ページ）
- QoS のモニタリング （122 ページ）
- QoS の設定例 （122 ページ）
- 次の作業 （134 ページ）
- QoS に関する追加情報 （134 ページ）
- QoS の機能履歴 （135 ページ）

QoS の前提条件

標準 Quality of Service（QoS）を設定する前に、次の事項を十分に理解しておく必要があります。

- 標準 QoS の概念
- 従来の Cisco IOS QoS。
- モジュラ QoS CLI（MQC）

- QoS 実装について。
- 使用するアプリケーションのタイプおよびネットワークのトラフィックパターン
- トラフィックの特性およびネットワークのニーズ。たとえば、ネットワークのトラフィックがバーストであるかどうか。音声およびビデオストリーム用の帯域幅確保の必要性
- ネットワークの帯域幅要件および速度
- ネットワーク上の輻輳発生箇所

QoS コンポーネント

Quality of Service（QoS）は、次の主要コンポーネントで構成されています。

- 分類：分類は、アクセスコントロールリスト（ACL）、DiffServ コードポイント（DSCP）、サービスクラス（CoS）、およびその他の要因に基づいて、トラフィックの 1 つのタイプを区別するプロセスです。
- マーキングと変換：マーキングは、特定の情報をネットワークのダウンストリームデバイスに伝送するか、デバイス内の 1 つのインターフェイスから別のインターフェイスに情報を伝送するためにトラフィック上で使用されます。トラフィックをマークすると、そのトラフィックの QoS 動作が適用されます。これは、**set** コマンドを直接使用するか、テーブルマップ経由で入力値を受け取って出力の値に直接変換することで実行します。
- シェーピングとポリシング：シェーピングはダウンストリームデバイスで輻輳が発生しないようにトラフィックレートを調整しながら、トラフィックの最大レートを強制するプロセスのことです。最も一般的な形式のシェーピングは、物理または論理インターフェイスから送信されるトラフィックを制限するために使用されます。ポリシングは、トラフィック クラスに最大レートを強制するために使用されます。レートを超過した場合は、イベント発生直後に特定のアクションが実行されます。
- キューイング：キューイングは、トラフィックの輻輳を防止するために使用されます。トラフィックは、帯域割り当てに基づいて処理およびスケジューリングするために、特定のキューに送信されます。次に、トラフィックはポートを介してスケジュールまたは送信されます。
- 帯域幅：帯域幅の割り当てにより、QoS ポリシーが適用されるトラフィックで使用可能な容量が決まります。
- 信頼：信頼により、トラフィックがデバイスを通過できるようになります。明示的なポリシー設定がない場合、エンドポイントから、またはエンドポイントへの DiffServ コードポイント（DSCP）値、プレシデンス値、または CoS 値は保持されます。

QoS の用語

この QoS コンフィギュレーション ガイドでは、次の用語が同じ意味で使用されます。

- アップストリーム（デバイスに対する方向）は、入力と同じ意味です。
- ダウンストリーム（デバイスに対する方向）は、出力と同じ意味です。

QoS の概要

Quality of Service（QoS）を設定することで、他のトラフィック タイプの代わりに特定のトラフィック タイプを優先的に処理できます。QoS を設定しない場合、デバイスはパケットの内容やサイズに関係なく、各パケットにベストエフォート型のサービスを提供します。デバイスは信頼性、遅延限界、スループットを保証せずにパケットを送信します。

次に、QoS が提供する具体的な機能を示します。

- 低遅延
- 帯域幅保証
- バッファリング能力とドロップ分野
- トラフィック ポリシング
- フレームまたはパケット ヘッダーの属性変更のイネーブル化
- 関連サービス

モジュラ QoS CLI

デバイスでは、QoS 機能はモジュラ QoS コマンドライン インターフェイス（MQC）を使用してイネーブルにできます。MQC はコマンドライン インターフェイス（CLI）構造を採用しています。これを使用すると、トラフィック ポリシーを作成し、作成したポリシーをインターフェイスにアタッチできます。1 つのトラフィック ポリシーには、1 つのトラフィック クラスと 1 つ以上の QoS 機能が含まれます。トラフィック クラスがトラフィックを分類するために使用されるのに対して、トラフィック ポリシーの QoS 機能は分類されたトラフィックの処理方法を決定します。MQC の主な目的の 1 つは、プラットフォームに依存しないインターフェイスを提供することにより、シスコ プラットフォーム全体の QoS を設定することです。

有線アクセスでサポートされる QoS 機能

次の表に、有線アクセスでサポートされる QoS 機能について説明します。

表 *2:* 有線アクセスでサポートされる *QoS* 機能

機能	説明
サポートされるターゲット	• ギガビット イーサネット • 10 ギガビット イーサネット • 40 ギガビット イーサネット • VLAN
設定手順	**service-policy** コマンドを使用してインストールされる QoS ポリシー。
ポート レベルでサポートされるキューの数	ポートでは最大 8 つのキューがサポートされます。

サポートされる分類メカニズム	• DSCP
	• IP precedence
	• CoS
	• QoS-group
	• 次を含む ACL のメンバーシップ：
	  • IPv4 ACL
	  • IPv6 ACL
	  • MAC ACL

階層型 QoS

デバイスは階層型 QoS（HQoS）をサポートします。HQoS を使用すると、次の作業を実行できます。

- 階層型分類：トラフィック分類は、他のクラスに基づいています。
- 階層型ポリシング：階層型ポリシーの複数のレベルでポリシングを設定するプロセス。
- 階層型シェーピング：シェーピングは、階層の複数のレベルで設定できます。

（注） 階層型シェーピングは、ポート シェーパーでのみサポートされます。ポート シェーパーでは、親に対してクラス デフォルトの設定だけが可能で、クラス デフォルトのアクションはシェーピングだけです。

QoS の実装

ネットワークは通常、ベストエフォート型の配信方式で動作します。したがって、すべてのトラフィックに等しいプライオリティが与えられ、適度なタイミングで配信される可能性はどのトラフィックでも同等です。輻輳が発生すると、すべてのトラフィックが等しくドロップされます。

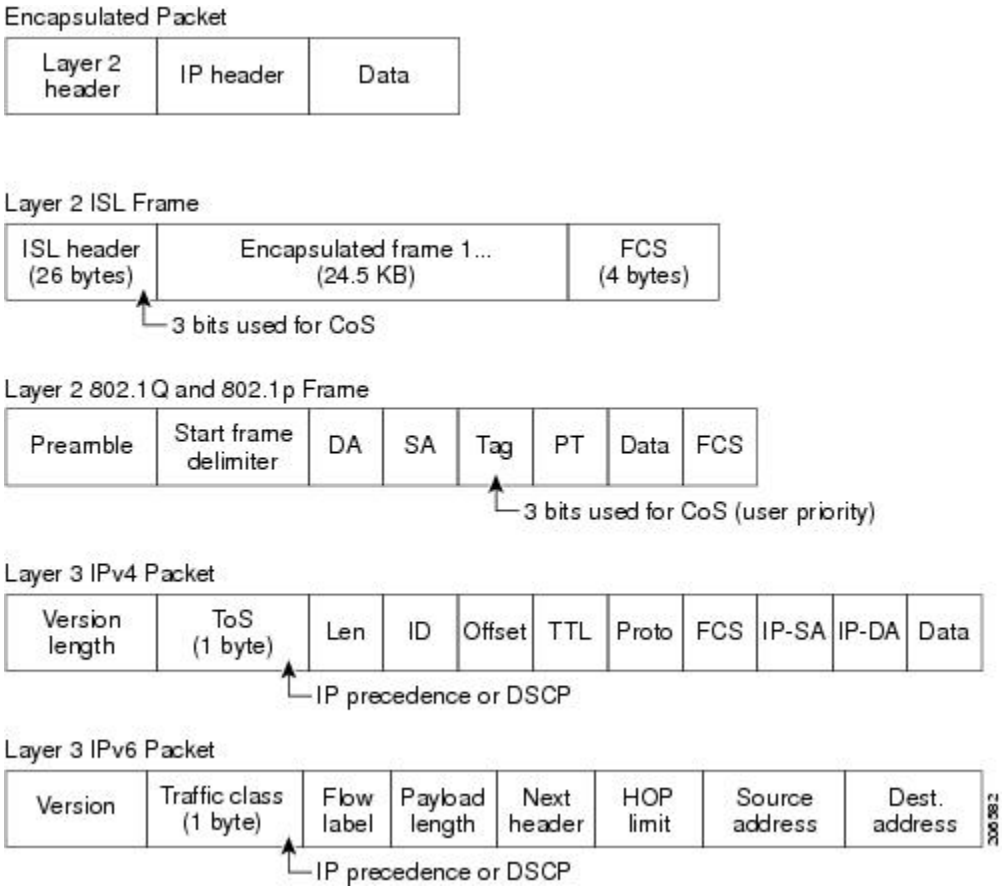
QoS 機能を設定すると、特定のネットワーク トラフィックを選択し、相対的な重要性に応じてそのトラフィックに優先度を指定し、輻輳管理および輻輳回避技術を使用して、優先処理を実行できます。ネットワークに QoS を実装すると、ネットワーク パフォーマンスがさらに予測しやすくなり、帯域幅をより効率的に利用できるようになります。

QoS は、インターネット技術特別調査委員会（IETF）の規格である Differentiated Services（Diff-Serv）アーキテクチャに基づいて実装されます。このアーキテクチャでは、ネットワークに入るときに各パケットを分類することが規定されています。

この分類は IP パケット ヘッダーに格納され、推奨されない IP タイプ オブ サービス（ToS）フィールドの 6 ビットを使用して、分類（クラス）情報として伝達されます。分類情報をレイヤ 2 フレームでも伝達できます。

次の図にレイヤ 2 フレームまたはレイヤ 3 パケットの特殊ビットを示します。

図 *1:* フレームおよびパケットにおける *QoS* 分類レイヤ



レイヤ 2 フレームのプライオリティ ビット

レイヤ 2 の ISL（スイッチ間リンク）フレーム ヘッダーには、下位 3 ビットで IEEE 802.1p サービス クラス（CoS）値を伝達する 1 バイトのユーザ フィールドがあります。レイヤ 2 ISL トランクとして設定されたポートでは、すべてのトラフィックが ISL フレームに収められます。

レイヤ 2 802.1Q フレーム ヘッダーには、2 バイトのタグ制御情報フィールドがあり、上位 3 ビット（ユーザ プライオリティ ビット）で CoS 値が伝達されます。レイヤ 2 802.1Q トランクとして設定されたポートでは、ネイティブ Virtual LAN（VLAN）のトラフィックを除くすべてのトラフィックが 802.1Q フレームに収められます。

他のフレーム タイプでレイヤ 2 CoS 値を伝達することはできません。

レイヤ 2 CoS 値の範囲は、0（ロー プライオリティ）～ 7（ハイ プライオリティ）です。

レイヤ 3 パケットのプライオリティ ビット

レイヤ 3 IP パケットは、IP precedence 値または Diffserv コード ポイント（DSCP）値のいずれかを伝送できます。DSCP 値は IP precedence 値と下位互換性があるので、QoS ではどちらの値も使用できます。

IP precedence 値の範囲は 0 ～ 7 です。DSCP 値の範囲は 0 ～ 63 です。

分類を使用したエンドツーエンドの QoS ソリューション

インターネットにアクセスするすべてのスイッチおよびルータはクラス情報に基づいて、同じクラス情報が与えられているパケットは同じ扱いで転送を処理し、異なるクラス情報のパケットはそれぞれ異なる扱いをします。パケットのクラス情報は、設定されているポリシー、パケットの詳細な検証、またはその両方に基づいて、エンドホストが割り当てるか、または伝送中にスイッチまたはルータで割り当てることができます。パケットの詳細な検証は、コア スイッチおよびルータの負荷が重くならないように、ネットワークのエッジ付近で行います。

パス上のスイッチおよびルータは、クラス情報を使用して、個々のトラフィック クラスに割り当てるリソースの量を制限できます。Diff-Serv アーキテクチャでトラフィックを処理するときの、各デバイスの動作をホップ単位動作といいます。パス上のすべてのデバイスに一貫性のあるホップ単位動作をさせることによって、エンドツーエンドの QoS ソリューションを構築できます。

ネットワーク上で QoS を実装する作業は、インターネットワーキング デバイスが提供する QoS 機能、ネットワークのトラフィック タイプおよびパターン、さらには着信および発信トラフィックに求める制御のきめ細かさによって、簡単にも複雑にもなります。

パケット分類

パケット分類は、特定の基準に基づいて定義したポリシーの複数のクラスの 1 つに属するものとしてパケットを識別するプロセスです。モジュラ QoS CLI（MQC）は、ポリシークラス ベースの言語です。ポリシー クラスの言語は、次の定義に使用されています。

- 1 つまたは複数の一致基準があるクラス マップ テンプレート
- 1 つまたは複数のクラスがポリシー マップに関連付けられているポリシーマップ テンプレート

ポリシーマップテンプレートは、デバイスの 1 つまたは複数のインターフェイスに関連付けられます。

パケット分類は、ポリシーマップで定義されたクラスの 1 つに属するものとしてパケットを識別するプロセスです。分類プロセスは、処理されるパケットがクラス内の特定のフィルタに一致した場合に終了します。これは、最初の一致による終了と呼ばれます。つまり、ポリシーマップ内のクラスの順序に関係なく、パケットがポリシー内の複数のクラスに一致する場合、最初のクラスの一致後に分類プロセスが終了します。

パケットがポリシーのクラスと一致しない場合は、ポリシーのデフォルトクラスに分類されます。すべてのポリシー マップには、システム定義のクラスのデフォルト クラスがあり、どのユーザー定義クラスにも一致しないパケットに一致します。

パケット分類は次のタイプに分類できます。

- パケットと合わせて伝搬される情報に基づく分類
- デバイス固有の情報に基づく分類
- 階層型分類

パケットと合わせて伝搬される情報に基づく分類

パケットの一部としてエンドツーエンドまたはホップ間で伝搬される情報に基づく分類には、一般的に次のものがあります。

- レイヤ 3 またはレイヤ 4 ヘッダーに基づく分類
- レイヤ 2 情報に基づく分類

レイヤ 3 またはレイヤ 4 ヘッダーに基づく分類

これは最も一般的な導入シナリオです。レイヤ 3 およびレイヤ 4 ヘッダーの多くのフィールドは、パケット分類に使用できます。

最もきめ細かいレベルでは、この分類方法はフロー全体を照合するために使用できます。この導入タイプで、アクセス コントロール リスト（ACL）を使用できます。ACL は、フローのさまざまなサブセット（送信元 IP アドレスのみ、宛先 IP アドレスのみ、または両方の組み合わせなど）に基づく照合に使用することもできます。

分類は、IP ヘッダーの precedence 値または DSCP 値に基づいて実行することもできます。IP precedence フィールドは、特定のパケットを処理する必要がある相対プライオリティを示すために使用されます。これは、IP ヘッダー内のタイプ オブ サービス（ToS）バイトの 3 ビットで構成されます。

次の表に、さまざまな IP precedence ビット値と名前を示します。

表 *3 : IP precedence* 値と名前

IP precedence 値	IP precedence ビット	IP precedence の名前
0	000	ルーチン
1	001	プライオリティ
2	010	即時
3	011	フラッシュ
4	100	フラッシュ オーバーライド

IP precedence 値	IP precedence ビット	IP precedence の名前
5	101	重大
6	110	インターネットワーク制御
7	111	ネットワーク制御

（注） ネットワークのルーティング制御トラフィックすべては、IP precedence 値 6 をデフォルトで使用します。また、IP precedence 値 7 は、ネットワーク制御トラフィック用に予約されています。したがって、IP precedence 値 6 および 7 はユーザー トラフィック用に推奨されません。

DSCP フィールドは、IP ヘッダーの 6 ビットで構成され、インターネット技術特別調査委員会（IETF）の DiffServ ワーキング グルーにより標準化されています。DSCP ビットが含まれた元の ToS バイトは、DSCP バイトの名前を変更しました。DSCP フィールドは、IP precedence と同様に IP ヘッダーの一部です。DSCP フィールドは、IP precedence フィールドのスーパー セットです。したがって、DSCP フィールドは、IP precedence に関連して説明した内容と同様の方法で使用され、設定されます。

（注） DSCP フィールド定義は IP precedence 値と下位互換性があります。

レイヤ 2 ヘッダー内の一部のフィールドは、ポリシーを使用して設定することもできます。

レイヤ 2 ヘッダーに基づく分類

レイヤ2ヘッダー情報に基づく分類は、さまざまな方法で実行できます。最も一般的な方法は次のとおりです。

- MAC アドレスベースの分類（アクセス グループの場合のみ）：分類は送信元 MAC アドレス（入力方向のポリシー用）および宛先 MAC アドレス（出力方向のポリシー用）に基づいています。
- サービス クラス：分類は、IEEE 802.1p 標準に基づくレイヤ 2 ヘッダーの 3 ビットに基づいて行われます。これは通常、IP ヘッダーの ToS バイトにマッピングします。
- VLAN ID：分類は、パケットの VLAN ID に基づいて行われます。

（注） レイヤ2ヘッダー内のこれらフィールドの一部は、ポリシーを使用して設定することもできます。

デバイス固有の情報に基づく分類

デバイスは分類がパケットヘッダーまたはペイロードの情報に基づいていない場合に使用できる分類メカニズムを提供します。

複数の入力インターフェイスから出力インターフェイスの特定のクラスに送信されるトラフィックを集約する必要が生じる場合があります。たとえば、複数のカスタマーエッジルータが、異なるインターフェイスの同じアクセスデバイスに接続される可能性があります。サービスプロバイダは、特定のレートでコアに送信されるすべての集約音声トラフィックをポリシングする場合があります。ただし、異なるカスタマーからの音声トラフィックには、異なるToS設定がなされている可能性があります。QoSグループベースの分類は、次のシナリオで役立つ機能です。

入力インターフェイスで設定されたポリシーは、QoSグループを特定の値に設定します。この値は出力インターフェイスでイネーブルになっているポリシーのパケットの分類に使用できます。

QoSグループは、デバイス内部のパケットデータ構造内のフィールドです。QoSグループは、デバイスの内部ラベルであり、パケットヘッダーの一部ではないことに注意してください。

QoS 有線モデル

QoS を実装するには、デバイスで次のタスクを実行する必要があります。

- トラフィック分類：パケットまたはフローを相互に区別します。
- トラフィックマーキングおよびポリシング：パケットがデバイスを移動するときに、特定の QoS を示すラベルを割り当て、パケットが設定されたリソース使用率制限に準拠するようにします。
- キューイングおよびスケジューリング：リソース競合があるすべての状況で、異なる処理を行います。
- シェーピング：デバイスから送信されるトラフィックが、特定のトラフィックプロファイルに適合するようにします。

入力ポートのアクティビティ

次のアクティビティはデバイスの入力ポートで発生します。

- 分類：パケットと QoS ラベルを関連付けて、パケットごとに異なるパスを分類します。たとえば、デバイスは、ある種類のトラフィックを別の種類のトラフィックと区別するためにパケット内の CoS または DSCP を QoS ラベルにマッピングします。生成された QoS ラベルは、このパケットでこれ以降に実行されるすべてのQoSアクションを識別します。
- ポリシング：ポリシングでは、着信トラフィックのレートを設定済みポリサーと比較して、パケットが適合か不適合かを判別します。ポリサーは、トラフィック フローで消費される帯域幅を制限します。その判別結果がマーカーに渡されます。

- マーキング：マーキングでは、パケットが不適合の場合の対処法に関して、ポリサーおよび設定情報を検討し、パケットの扱い（パケットを変更しないで通過させるか、パケットのQoSラベルをマークダウンするか、またはパケットをドロップするか）を決定します。

出力ポートのアクティビティ

次のアクティビティはデバイスの出力ポートで発生します。

- ポリシング：ポリシングでは、着信トラフィックのレートを設定済みポリサーと比較して、パケットが適合か不適合かを判別します。ポリサーは、トラフィック フローで消費される帯域幅を制限します。その判別結果がマーカーに渡されます。
- マーキング：マーキングでは、パケットが不適合の場合の対処法に関して、ポリサーおよび設定情報を検討し、パケットの扱い（パケットを変更しないで通過させるか、パケットのQoSラベルをマークダウンするか、またはパケットをドロップするか）を決定します。
- キューイング：キューイングでは、使用する出力キューを選択する前に、QoS パケット ラベルと対応する DSCP 値または CoS 値を評価します。複数の入力ポートが 1 つの出力ポートに同時にデータを送信すると輻輳が発生することがあるため、重み付けテール ドロップ（WTD）によってトラフィック クラスを区別し、QoS ラベルに基づいてパケットに別々のしきい値を適用します。しきい値を超過している場合、パケットはドロップされます。

分類

分類とは、パケットのフィールドを検証して、トラフィックの種類を区別するプロセスです。QoSがデバイス上でイネーブルになっている場合にのみ、分類はイネーブルです。デフォルトにより、QoS はデバイスでイネーブルになっています。

分類時に、デバイスは検索処理を実行し、パケットに QoS ラベルを割り当てます。QoS ラベルは、パケットに対して実行するすべての QoS アクション、およびパケットの送信元キューを識別します。

アクセス コントロール リスト

IP標準ACL、IP拡張ACL、またはレイヤ2MAC ACLを使用すると、同じ特性を備えたパケット グループ（クラス）を定義できます。また IPv6 ACL に基づいて IP トラフィックを分類することもできます。

QoSのコンテキストでは、アクセスコントロールエントリ（ACE）の許可および拒否アクションの意味が、セキュリティ ACL の場合とは異なります。

- 許可アクションとの一致が検出されると（最初の一致の原則）、指定のQoS関連アクションが実行されます。

- 拒否アクションと一致した場合は、処理中の ACL がスキップされ、次の ACL が処理されます。
- 許可アクションとの一致が検出されないまま、すべての ACE の検証が終了した場合、そのパケットでは QoS 処理は実行されず、デバイスがベストエフォート型サービスを実行します。
- ポートに複数の ACL が設定されている場合に、許可アクションを含む最初の ACL とパケットの一致が見つかると、それ以降の検索処理は中止され、QoS 処理が開始されます。

（注） アクセス リストを作成するときは、アクセス リストの末尾に暗黙の拒否ステートメントがデフォルトで存在し、それ以前のステートメントで一致が見つからなかったすべてのパケットに適用されることに注意してください。

ACL でトラフィック クラスを定義した後で、そのトラフィック クラスにポリシーを結合できます。ポリシーにはそれぞれにアクションを指定した複数のクラスを含めることができます。ポリシーには、特定の集約としてクラスを分類する（DSCP を割り当てるなど）コマンドまたはクラスのレート制限を実施するコマンドを含めることができます。このポリシーを特定のポートに結合すると、そのポートでポリシーが有効になります。

IP ACL を実装して IP トラフィックを分類する場合は、**access-list** グローバル コンフィギュレーション コマンドを使用します。レイヤ 2 MAC ACL を実装して非 IP トラフィックを分類する場合は、**mac access-list extended** グローバル コンフィギュレーション コマンドを使用します。

クラス マップ

クラス マップは、特定のトラフィック フロー（またはクラス）に名前を付けて、他のすべてのトラフィックと区別するためのメカニズムです。クラスマップでは、さらに細かく分類するために、特定のトラフィック フローと照合する条件を定義します。この条件には、ACL で定義されたアクセス グループとの照合や、DSCP 値または IP precedence 値あるいは CoS 値の特定のリストとの照合を含めることができます。複数のトラフィック タイプを分類する場合は、別のクラス マップを作成し、異なる名前を使用できます。パケットをクラス マップ条件と照合した後で、ポリシー マップを使用してさらに分類します。

（注） 同じクラスマップに IPv4 と IPv6 の分類基準を同時に設定することはできません。ただし、同じポリシー内の異なるクラスマップで設定することは可能です。

クラスマップを作成するには、**class-map** グローバル コンフィギュレーション コマンドまたは **class** ポリシー マップ コンフィギュレーション コマンドを使用します。複数のポリシー間でマップを共有する場合には、**class-map** コマンドを使用する必要があります。**class-map** コマンドを入力すると、デバイスによってクラスマップ コンフィギュレーション モードが開始されます。

class class-default ポリシーマップ コンフィギュレーション コマンドを使用して、デフォルト クラスを作成できます。デフォルト クラスはシステム定義であり、設定することはできません。分類されていないトラフィック（トラフィック クラスで指定された一致基準を満たさないトラフィック）は、デフォルト トラフィックとして処理されます。

存続可能時間分類

ACLマップに基づいてパケットを分類できます。ACLリストの基準として存続可能時間（TTL）を設定し、着信パケットの TTL チェックを実行できます。アクセス コントロール エントリは、IPv4 TTLをチェックし、着信パケットの値を照合するために使用されます。分類されたパケットは、ポリシーマップアクションに基づいてマーキングまたはポリシングされます。この分類ではキューイングを設定できません。

次に、TTL 分類の例を示します。

```
policy-map TTL_MATCH
 class IPV4_TTL
  police rate 6000000000
  set dscp af23

ip access-list extended IPV4_TTL
 permit ip any any ttl eq 100
 permit tcp any any ttl ne 150

!
Device#show run class-map IPV4_TTL
class-map match-all IPV4_TTL
 match access-group name IPV4_TTL
!

Device#show policy-map interface hun1/0/47

HundredGigE1/0/47

  Service-policy output: TTL_MATCH

    Class-map: IPV4_TTL (match-all)
    553567424 packets
    Match: access-group name IPV4_TTL
    police:
    rate 6000000000 bps, burst 187500000 bytes
    conformed 22983406600 bytes; actions:
    transmit
    exceeded 32375773000 bytes; actions:
    drop
    conformed 588922000 bps, exceeded 830894000 bps
    QoS Set
    dscp af23

    Class-map: class-default (match-any)
    2184433710 packets
    Match: any
```

レイヤ 3 パケット長分類

この機能は、IP ヘッダーのレイヤ 3 パケット長に基づいて、トラフィックを照合して分類する機能を提供します。レイヤ 3 パケット長とは、IP データグラム長と IP ヘッダー長の合計です。クラスポリシーマップの一致基準としてパケット長を設定し、着信パケットで値を照合することができます。分類されたパケットは、ポリシーマップアクションに基づいてマーキングまたはポリシングされます。この機能は、IPv6 パケットでは機能しません。

次に、レイヤ 3 パケット長の分類の例を示します。

```
Service-policy output: PACKET_MATCH1

Class-map: class-default (match-any)
 16281588 packets
  Match: any

Service-policy : L3_MATCH

Class-map: PACKET_LENGTH_1 (match-any)
  9910510 packets
  Match: packet length 7582
  Match: packet length 5000
  QoS Set
  dscp cs2
  police:
 rate 3 %
 rate 1200000000 bps, burst 37500000 bytes
 conformed 10000 bytes; actions:
   transmit
 exceeded 112121 bytes; actions:
   drop
 conformed 500 bps, exceeded 3434 bps

Class-map: PACKET_LENGTH_2 (match-all)
 6371042 packets
  Match: dscp cs4 (32)
  Match: packet length 7759
  police:
 rate 12000000000 bps, burst 375000000 bytes
 conformed 44545 bytes; actions:
 transmit
 exceeded 34343 bytes; actions:
 drop
 conformed 1211 bps, exceeded 11211 bps

Class-map: class-default (match-any)
 36 packets
  Match: any
 QoS Set
   precedence 3
Device#

class-map match-any PACKET_LENGTH_1
match packet length min 7582 max 7582
match packet length min 5000 max 5000

class-map match-all PACKET_LENGTH_2
match dscp cs4
match packet length min 7759 max 7759
```

レイヤ 2 SRC-Miss または DST-Miss の分類

トラフィックは、送信元 MAC アドレスまたは宛先 MAC アドレスについて、MAC アドレステーブルに見つからない MAC アドレスで分類できます。L2-Miss 分類によるポリシーマップは、入力方向でレイヤ2インターフェイスに適用できます。ポリシング、マーキング、または再マーキングアクションは、この分類を使用して適用できます。L2-Miss 分類は、レイヤ 3 インターフェイスには適用できません。この分類ではキューイングを設定できません。

次に、L2-Miss 分類の例を示します。

```
Device #show run class-map DST-MISS
    class-map match-any DST-MISS
    match l2 dst-mac miss

Device #show run class-map SRC-MISS
    class-map match-all SRC-MISS
    match l2 src-mac miss

Device #show policy-map L2-MISS
Policy Map L2-MISS
    Class DST-MISS
        set dscp af22
    police cir percent 10
        conform-action transmit
        exceed-action drop
    Class SRC-MISS
        set precedence 1
    police rate percent 20
        conform-action transmit
        exceed-action drop
!
end

Device#
```

ポリシー マップ

ポリシー マップでは、作用対象のトラフィック クラスを指定します。アクションには次が含まれます。

- トラフィック クラスに特定の DSCP 値または IP precedence 値を設定する
- トラフィック クラスに CoS 値を設定する
- QoS グループを設定する
- トラフィックがアウト オブ プロファイルになった場合の、トラフィックの帯域幅制限やアクションを指定する

ポリシー マップを効率的に機能させるには、ポートにポリシー マップを結合する必要があります。

ポリシーマップは、**policy-map** グローバル コンフィギュレーション コマンドを使用して作成し、名前を付けます。このコマンドを入力すると、デバイスはポリシーマップ コンフィギュレーション モードを開始します。このモードでは、**class** または **set** ポリシーマップ コンフィ

ギュレーション コマンドおよびポリシーマップ クラス コンフィギュレーション コマンドを使用して、特定のトラフィッククラスに対して実行するアクションを指定します。

ポリシーマップは、ポリシーマップ クラス コンフィギュレーション コマンド**police** と **bandwidth** を使用して設定することもできます。これらのコマンドは、ポリサー、トラフィックの帯域幅制限、および制限を超過した場合のアクションを定義します。加えて、ポリシーマップは、**priority** ポリシーマップ クラス コンフィギュレーション コマンド（クラスの優先順位をスケジューリングする）、またはキューイング ポリシーマップ クラス コンフィギュレーション コマンド（**queue-buffers** および **queue-limit**）を使用すると、より詳細に設定できます。

ポリシーマップを有効にするには、**service-policy** インターフェイス コンフィギュレーション コマンドを使用してポートにマップを結合します。

（注） **priority** と **set** の両方をポリシーマップに設定することはできません。これらのコマンド両方をポリシー マップに設定すると、ポリシー マップをインターフェイスに適用した際に、エラーメッセージが表示されます。次に、この制限の例を示します。

```
Device# configure terminal
Device(config)# class-map cmap
Device(config-cmap)# exit
Device(config)# class-map classmap1
Device(config-cmap)# exit
Device(config)# policy-map pmap
Device(config-pmap)# class cmap
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# exit
Device(config-pmap)# class classmap1
Device(config-pmap-c)# set dscp 10
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface HundredGigE1/0/2
Device(config-if)# service-policy output pmap

Non-queuing action only is unsupported in a queuing policy!!!
%QOS-6-POLICY_INST_FAILED:
Service policy installation failed
```

物理ポートのポリシー マップ

実行対象となるトラフィック クラスを指定する非階層型ポリシー マップを、物理ポート上に設定できます。アクションには、特定の DSCP あるいはトラフィック クラスでの IP プレシデンス値または QoS の値の設定、一致する各トラフィック クラス（ポリサー）に対するトラフィックの帯域幅限度の指定、トラフィックがアウト オブ プロファイル（マーキング）の場合の処理などが含まれます。

ポリシー マップには、次の特性もあります。

- 1 つのポリシーマップに、それぞれ異なる一致条件とポリサーを指定した複数のクラス ステートメントを指定できます。
- ポリシー マップには、事前に定義されたデフォルトのトラフィック クラスを含めることができます。デフォルトのトラフィック クラスはマップの末尾に明示的に配置されます。

class class-default ポリシーマップ コンフィギュレーション コマンドを使用してデフォルトのトラフィッククラスを設定すると、未分類トラフィック（トラフィッククラスで指定された一致基準に一致しないトラフィック）はデフォルトのトラフィッククラス（**class-default**）として処理されます。

- 1 つのポートから受信されたトラフィック タイプごとに、別々のポリシー マップ クラスを設定できます。

VLAN のポリシーマップ

デバイスは、VLAN の QoS 機能をサポートします。これにより、ユーザーは、着信フレームの VLAN 情報を使用して VLAN レベルで QoS 処理（分類と QoS アクション）を実行できます。VLAN ベースの QoS では、サービス ポリシーが SVI インターフェイスに適用されます。VLAN ポリシー マップに属するすべての物理インターフェイスは、ポートベースのポリシーマップの代わりに VLAN ベースのポリシー マップが表示されるようにプログラムする必要があります。

ポリシーマップは VLAN SVI に適用されますが、ポート単位で実行できるのはマーキングまたは再マーキングアクションのみです。複数の物理ポートからのトラフィックの合計が認識されるようにポリサーを設定できません。各ポートは、そのポートに着信するトラフィックを制御する別のポリサーを必要とします。

QoS プロファイル

デバイスは、Ternary Content Addressable Memory（TCAM）を使用して分類ルールを保存します。TCAM リソースの使用を最適化するには、QoS プロファイルを使用して、使用頻度の低い機能の一部をオフにし、必要に応じてオンにします。

qos profile {**default** | **extended**} コマンドを使用して、必要な分類機能セットを選択できます。**default** キーワードは、共通の分類機能のみをロードします。**extended** キーワードは、デバイスで使用可能な完全な分類機能セット（ただしスケールは縮小されています）をロードします。デフォルトでは、一般的に使用される分類機能のみがデバイスに設定されます。

qos profile extended は、共通の分類機能とともに TCP フラグを有効にします。

show platform software fed active qos profile コマンドを使用して、デバイスに設定されている QoS プロファイルを確認できます。

例

```
device# show platform software fed active qos profile
Using default - Common Classification Features
```

セキュリティグループ分類

セキュリティグループの分類には、送信元セキュリティグループタグ（SGT）と宛先セキュリティグループタグ（DGT）によって指定される送信元グループと宛先グループの両方が含まれます。

SGT QoS 分類の目的は、ユーザーグループを活用してポリシーの粒度を高めることです。これにより、ポリシーはアプリケーションに対応するだけでなく、ユーザーアイデンティティ（またはユーザーが属するユーザーのグループ）に基づいて差別化されたサービスを提供します。

SGT または DGT に基づく出力 QoS 分類はサポートされていません。

SGT ベースの QoS

SGT ベースの QoS 機能は、定義されたユーザーグループまたはデバイスに対して、QoS ポリシーおよびアクションに基づくトラフィックのクラスに特別な処理を提供します。この機能により、異なるユーザーグループによって開始されたアプリケーションやトラフィックに複数の QoS ポリシーを割り当てることができます。各ユーザーグループは一意の SGT 値によって定義され、MQC ベースの QoS 構成をサポートできます。

SGT ベースの QoS 機能は、SGT-DGT ベースのパケット分類に使用するユーザーグループおよびデバイスベースの QoS サービスレベルの両方に適用できます。これは、QoS ポリシーの優先度設定に使用するコンテキスト情報に基づくユーザーグループの定義をサポートできる可能性もあります。

SGACL との DGID の共有

リソースの制限により、4096 のセキュリティグループ宛先タグ（DGT）のみがサポートされます。DGT に基づく分類は、DGID と呼ばれるセキュリティ宛先タグ ID によって実現されます。DGID はグローバルリソースであり、SGACL と共有されます。DGID 割り当ては、検出された順序で行われます。デバイスでは、起動時に、QoS ポリシー設定の前に SGACL 設定が適用されます。したがって、DGID は最初に SGACL に割り当てられ、次に QoS ポリシーに割り当てられます。

show platform software fed sw active sgacl detail コマンドは、DGT から DGID へのマッピングを表示します。

例

```
device# show platform software fed active sgacl detail

Global Enforcement: On
*Refcnt: for the non-SGACL feature
========================== DGID Table ====================================
SGT/Refcnt     DGT     DGID   hash  test_cell monitor   permitted  denied
==========================================================================
*/1            24        1     24
 24            24        1     24       Off     Off        0          0
```

SGT ベースの QoS の制限

次に、SGT ベースの QoS 機能の制限事項を示します。

- SGT ベースの QoS は、トンネルインターフェイスではサポートされません。
- 4096 のセキュリティ宛先タグと 65539 のセキュリティ送信元タグのみがサポートされます。

- SGT ベースのポリシーは、インターフェイスの入力方向にのみ適用できます。

アップグレードまたはダウングレードの制約事項

- 以前のリリースから Cisco IOS XE リリース 16.12.x 以降にアップグレードする場合、サポートされる DGID の最大数は 256 です。この問題を解決するには、スイッチをリロードします。
- Cisco IOS XE リリース 17.1.x から IOS XE 16.12.x リリースへのダウングレードの場合、割り当てられた DGID は 4096 と表示されますが、256 個の DGID のみがサポートされます。この問題を解決するには、スイッチをリロードします。
- tcp フラグがポリシーで設定されている場合、In-Service Software Upgrade（ISSU）は失敗します。ISSU を実行するには、最初に tcp フラグ設定を削除します。
- インターフェイスにアタッチされているポリシーマップが tcp フラグに基づいてトラフィックを分類すると、ISSU のアップグレードは失敗します。ISSU を実行するには、ポリシーマップをインターフェイスから切り離すか、tcp フラグの分類を削除します。

入力ポート FIFO パーサー

入力ポート FIFO（IPF）は、着信ネットワークトラフィックを解析して、フレームをさまざまなプライオリティレベルに分類します。トラフィッククラスはさまざまなパケット形式から抽出されます。たとえば、トラフィッククラスは、IP パケットの場合は Differentiated Services Code Point（DSCP; DiffServ コードポイント）から、dot1q タグパケットの場合はサービスクラス（CoS）から抽出できます。これらのトラフィッククラスはさらにプライオリティレベルにマッピングされます。このプライオリティレベルは、輻輳発生時にドロップを決定するために使用されます。

IPF パーサーは、グローバルモードと分離モード（ポートレベルでのハイおよびロープライオリティ設定）で使用できます。デフォルトでは、分離モードです。分離モードでは、プライオリティの区別はシステムレベルではなくポートレベルで行われます。

IPF パーサーをグローバルモードで設定するには、次のコマンドを使用します。

```
configure port-ingress-fifo mode global
```

次に、トラフィッククラスとプライオリティのマッピングを表示する **show** コマンドの例を示します。

```
Device# show platform hardware fed active qos ipf interface twentyFiveGigE 1/0/1 cos-map

IPF cos to traffic class map for Interface [cos : traffic-class]:
------------------------------------------------------------------------
0 : 0           1 : 1           2 : 2           3 : 3
4 : 4           5 : 5           6 : 6           7 : 7
8 : 4           9 : 4           10 : 4          11 : 4
12 : 4          13 : 4          14 : 4          15 : 4

Device# show platform hardware fed active qos ipf interface twentyFiveGigE 1/0/1 dscp-map

IPF dscp to traffic class map for Interface [dscp : traffic-class]:
```

```
------------------------------------------------------------------
0 : 0           1 : 0           2 : 0           3 : 0
4 : 0           5 : 0           6 : 0           7 : 0
8 : 1           9 : 1           10 : 1          11 : 1
12 : 1          13 : 1          14 : 1          15 : 1
16 : 2          17 : 4          18 : 4          19 : 4
20 : 4          21 : 4          22 : 4          23 : 4
24 : 3          25 : 4          26 : 4          27 : 4
28 : 4          29 : 4          30 : 4          31 : 4
32 : 4          33 : 4          34 : 4          35 : 4
36 : 4          37 : 4          38 : 4          39 : 4
40 : 4          41 : 4          42 : 4          43 : 4
44 : 4          45 : 4          46 : 5          47 : 4
48 : 6          49 : 4          50 : 4          51 : 4
52 : 4          53 : 4          54 : 4          55 : 4
56 : 7          57 : 4          58 : 4          59 : 4
60 : 4          61 : 4          62 : 4          63 : 4
```

```
Device#show platform hardware fed active qos ipf interface twentyFiveGigE 1/0/1 exp-map

IPF exp to traffic class map for Interface [exp : traffic-class]:
------------------------------------------------------------------
0 : 0           1 : 1           2 : 2           3 : 3
4 : 4           5 : 5           6 : 6           7 : 7
```

```
Device#show platform hardware qos ipf interface twentyFiveGigE 1/0/1 ipf-parse-cfg
IPF configration for Interface:
--------------------------------
Port Trust:           Enabled
Default TC:           0
Dscp based parsing:   Disabled
Exp based parsing:    Disabled
Fdcos based parsing:  Enabled
cos based parsing:    Disabled
```

```
Device#show platform hardware fed active qos ipf tc-to-pri asic 0
IPF traffic class to priority for[Asic:Core:TlaInst]::[0:0:0]
-------------------------------------------------------------
Priority                Traffic Classes
--------                ---------------
Low Pri :               0  1  4
High Pri:               2  3  5  6  7
IPF traffic class to priority for[Asic:Core:TlaInst]::[0:0:1]
-------------------------------------------------------------
Priority                Traffic Classes
--------                ---------------
Low Pri :               0  1  4
High Pri:               2  3  5  6  7
```

統計情報の show コマンド：

```
Device#show platform hardware fed active qos ipf statistics asic 0
Ipf Statistics:[Asic|Core|Tla] : [0 | 0 | 0] - Global Mode
-----------------------------------------------
Ipf misc packet drops:                          0
Ipf Drop Statistics
-------------------
low pri Frames drop:                        0
low pri mop Frames drop:                    0
high pri Frames drop:                       0
almost full Frames drop:                    0
RCP Frames drop:                            0

Ipf Statistics:[Asic|Core|Tla] : [0 | 0 | 1] - Global Mode
-----------------------------------------------
Ipf misc packet drops:                          0
```

```
Ipf Drop Statistics
-------------------
low pri Frames drop:                    0
low pri mop Frames drop:                0
high pri Frames drop:                   0
almost full Frames drop:                0
RCP Frames drop:                        0
```

ポリシング

パケットが分類され、DSCP ベース、CoS ベース、または QoS グループのラベルが割り当てられると、ポリシングおよびマーキング プロセスを開始できます。

ポリシングには、トラフィックの帯域幅限度を指定するポリサーの作成が伴います。制限を超えるパケットは、「アウトオブプロファイル」または「不適合」になります。各ポリサーはパケットごとに、パケットが適合か不適合かを判別し、パケットに対するアクションを指定します。これらのアクションはマーカーによって実行されます。パケットを変更しないで通過させるアクション、パケットをドロップするアクション、またはパケットに割り当てられた DSCP または CoS 値を変更（マークダウン）してパケットの通過を許可するアクションなどがあります。

パケットの混乱を避けるため、通常、適合トラフィックも不適合トラフィックも同じキューを通過します。

（注） すべてのトラフィックは、ブリッジングされるかルーティングされるかに関係なく、ポリサーの影響を受けます（ポリサーが設定されている場合）。その結果、ブリッジングされたパケットは、ポリシングまたはマーキングが行われたときにドロップされたり、DSCP または CoS フィールドが変更されたりすることがあります。

物理ポートでのみポリシングを設定できます。

ポリシーマップおよびポリシングアクションを設定したら、**service-policy** インターフェイス コンフィギュレーション コマンドを使用して、入力ポートまたは出力ポートにポリシーマップを付加します。

トークンバケット アルゴリズム

ポリシングはトークンバケットアルゴリズムを使用します。各フレームがデバイスに着信すると、バケットにトークンが追加されます。バケットにはホールがあり、平均トラフィック レートとして指定されたレート（ビット/秒）で送信されます。バケットにトークンが追加されるたびに、デバイスは、バケット内に十分なスペースがあるかを確認します。十分なスペースがなければ、パケットは不適合とマーキングされ、指定されたポリサーアクション（ドロップまたはマークダウン）が実行されます。

バケットが満たされる速度は、バケット深度（burst-byte）、トークンが削除されるレート（rate-bps）、および平均レートを上回るバースト期間によって決まります。バケットのサイズ

によってバースト長に上限が設定され、バックツーバックで送信できるフレーム数が制限されます。バースト期間が短い場合、バケットはオーバーフローせず、トラフィック フローに何のアクションも実行されません。ただし、バースト期間が長く、レートが高い場合、バケットはオーバーフローし、そのバーストのフレームに対してポリシングアクションが実行されます。

バケットの深さ（バケットがオーバーフローするまでの許容最大バースト）を設定するには、**police** ポリシーマップ クラス コンフィギュレーション コマンドの burst-byte オプションを使用します。トークンがバケットから削除される速度（平均レート）を設定するには、**police** ポリシーマップ クラス コンフィギュレーション コマンドの rate オプションを使用します。

マーキング

マーキングは、特定の情報をネットワークのダウンストリームデバイスに伝送するか、デバイス内の 1 つのインターフェイスから別のインターフェイスに情報を伝送するために使用します。

マーキングは、パケットヘッダーの特定のフィールド/ビットを設定するか、デバイス内部のパケット構造内の特定のフィールドを設定するために使用できます。さらに、マーキング機能はフィールド間のマッピングの定義に使用できます。QoS では次のマーキング方法を使用できます。

- パケット ヘッダー
- デバイス固有の情報
- テーブル マップ

パケット ヘッダーのマーキング

パケット ヘッダー フィールドのマーキングは 2 種類の一般的なカテゴリに分類できます。

- IPv4/v6 ヘッダー ビット マーキング
- レイヤ 2 ヘッダー ビット マーキング

IP レベルのマーキング機能は、precedence を設定したり、IP ヘッダー内の DSCP を特定の値に設定したりして、ダウンストリームデバイス（スイッチまたはルータ）で特定のホップごとの動作を実行するために使用されます。また、異なる入力インターフェイスからのトラフィックを、出力インターフェイス内の単一のクラスに集約するためにも使用できます。この機能は現在、IPv4 および IPv6 ヘッダーでサポートされています。

レイヤ 2 ヘッダーのマーキングは、通常、ダウンストリーム デバイス（スイッチまたはルータ）のドロップ動作に影響を与えるために使用されます。これは、レイヤ2ヘッダーの一致と並行して動作します。ポリシーマップを使用して設定されるレイヤ2ヘッダーのビットはサービス クラスです。

スイッチ固有の情報のマーキング

この形式のマーキングには、パケットヘッダーの一部ではないパケットデータ構造内のフィールドのマーキングが含まれます。これにより、後でデータパスでマーキングを使用できるようになります。これはスイッチ間で伝搬されません。QoS グループのマーキングはこのカテゴリに分類されます。この形式のマーキングは、入力インターフェイスで有効になっているポリシーだけでサポートされます。対応する照合機能を同じスイッチの出力インターフェイスでイネーブルにし、適切な QoS アクションを適用することができます。

テーブル マップのマーキング

テーブル マップ マーキングは変換表を使用したフィールド間のマッピングおよび変換を可能にします。この変換表はテーブル マップと呼ばれます。

インターフェイスに接続されているテーブルマップに応じて、パケット内の CoS、DSCP、およびPrecedence値が書き換えられます。デバイスにより、入力のテーブルマップポリシーと出力のテーブルマップポリシーの両方を設定できます。

たとえば、テーブル マップは、レイヤ 2 CoS 設定をレイヤ 3 の precedence 値にマッピングするのに使用できます。この機能により、マッピングを実行する方法を示す1つのテーブルに複数の **set** コマンドを組み合わせて使用することができます。このテーブルは複数のポリシーで参照するか、または同じポリシー内で複数回参照することができます。

テーブル マップ ベースのポリシーでは、次の機能がサポートされています。

- 変換：1 つの DSCP 値セットから別の DSCP 値セットにマッピングするテーブル マップを利用できます。また、このテーブル マップは出力ポートに付加できます。
- 書き換え：入力パケットは設定されたテーブル マップに基づいて書き換えられます。
- マッピング：テーブル マップ ベースのポリシーは、set ポリシーの代わりに使用できます。

テーブル マップ マーキングには、次の手順が必要です。

1. テーブルマップの定義：**table-map** グローバル コンフィギュレーション コマンドを使用して値をマッピングします。テーブルが使用されるクラスまたはポリシーは認識されません。テーブルマップのデフォルトのコマンドは、「from」フィールドで一致がない場合に値が「to」フィールドにコピーされることを示すために使用されます。
2. ポリシー マップの定義：テーブル マップを使用するポリシー マップを定義します。
3. ポリシーをインターフェイスに関連付けます。

（注） 入力ポートのテーブル マップ ポリシーによって、そのポートの信頼設定が qos-marking の「from」タイプに変更されます。

（注） dscp 値以外の値を信頼するには、テーブルマップを入力方向でデフォルトのコピーとともに使用します。

（注） 出力のテーブルマップポリシーで QoS グループを DSCP 値にマッピングすると、QoS グループは DSCP に対応する同等の COS 値をマッピングしません。QoS グループをゼロ以外の COS 値に定義する場合は、COS テーブルマップに別の QoS グループを設定します。

トラフィックの調整

ネットワークでQoSをサポートするには、サービスプロバイダ ネットワークに入るトラフィックをネットワーク境界ルータでポリシングし、トラフィック レートがサービス範囲内に収まるようにする必要があります。ネットワーク コアのプロビジョニングで処理できるように設定されているトラフィックよりも多くのトラフィックがネットワーク境界のいくつかのルータから送信開始されると、トラフィック負荷の増加によってネットワーク輻輳が発生します。ネットワークのパフォーマンスが低下すると、すべてのネットワーク トラフィックで QoS を提供することが困難になります。

トラフィック ポリシング機能（ポリシング機能を使用）およびシェーピング機能（トラフィック シェーピング機能を使用）はトラフィック レートを管理しますが、トークンが不足した場合のトラフィックの処理方法が異なります。トークンの概念は、トークン バケット方式、トラフィック測定機能に基づいています。

（注） ネットワーク トラフィックでQoSテストを実行すると、シェーパー データとポリシング データで異なる結果が生じることがあります。シェーピングからのネットワーク トラフィック データの方が、より正確な結果が得られます。

この表は、ポリシングとシェーピングの機能を比較します。

表 *4:* ポリシングとシェーピングの機能の比較

ポリシング機能	シェーピング機能
適合するトラフィックをライン レートで送信し、バーストを許可します。	トラフィックが固定レートでスムーズに送信されます。
トークンが不足すると、アクションがただちに実行されます。	トークンが不足すると、パケットをバッファし、後でトークンが使用可能になった時点で送信します。シェーピングを使用するクラスにはキューが関連付けられており、このキューを使用してパケットがバッファされます。

ポリシング機能	シェーピング機能
ポリシングは、ビット/秒、パケット/秒、およびセル/秒など複数の単位で設定できます。	シェーピングの設定単位はビット/秒だけです。
ポリシングには、イベントに複数の可能なアクションが関連付けられています。このようなアクションの例としては、イベント、マーキング、ドロッピングなどがあります。	シェーピングはプロファイルを満たさないパケットをマークできません。
入出力両方のトラフィックで機能します。	出力トラフィックに対してのみ実装されます。
ウィンドウ サイズを小さくしたためにパケット ドロップが発生すると、伝送制御プロトコル（TCP）は、回線速度でラインを検出しますが、設定されたレートに適合します。	TCP は低速回線があることを検出し、再送信タイマーを適切に調整できます。これにより、再送信の範囲が狭くなり、TCP に負担をかけません。

ポリシング

QoS ポリシング機能は、トラフィック クラスに最大レートを強制するために使用されます。QoS ポリシング機能は、プライオリティ機能と合わせて、プライオリティ トラフィックを制限するためにも使用できます。レートを超過した場合は、イベント発生直後に特定のアクションが実行されます。レート（認定情報レート [CIR] および最大情報レート [PIR]）とバースト パラメータ（適合バースト サイズ [B_c] および拡張バースト サイズ [B_e]）は、すべてバイト/秒で設定されます。

QoS では次のポリシング形式またはポリサーがサポートされます。

- シングルレート 2 カラー ポリシング
- デュアルレート 3 カラー ポリシング

（注） シングルレート 3 カラー ポリシングはサポートされません。

シングルレート 2 カラー ポリシング

シングルレート 2 カラー ポリサーは、CIR と B_c だけを設定するモードです。

B_c は任意のパラメータであり、これが指定されていない場合、デフォルトで計算されます。このモードでは、着信パケットに十分なトークンがある場合、パケットは適合すると見なされます。パケットの到着時に、十分なトークンが B_c の範囲内で使用できない場合、パケットは設定レートを超えたと見なされます。

（注） トークンバケットアルゴリズムの詳細については、トークンバケットアルゴリズム（55 ページ）を参照してください。

デュアルレート 3 カラー ポリシング

デュアルレートポリサーでは、デバイスはカラーブラインドモードのみをサポートします。このモードでは、認定情報レート（CIR）および最大情報レート（PIR）を設定します。名前からわかるように、この場合、最大レート用に 1 つ、認定レート用に 1 つの、合わせて 2 つのトークン バケットがあります。

（注） トークンバケットアルゴリズムの詳細については、トークンバケットアルゴリズム（55 ページ）を参照してください。

カラー ブラインド モードでは、最大レートのバケットの着信パケットが最初にチェックされます。十分な数のトークンがない場合、パケットはレートに違反していると見なされます。十分な数のトークンがある場合、次に適合レートのバケットのトークンをチェックして、十分な数のトークンがあるかどうかを判別します。最大レートのバケットにあるトークンは、パケットのサイズによって減少します。十分な数のトークンがない場合、パケットが設定されているレートを超過していると見なされます。十分な数のトークンがある場合、パケットは適合すると見なされ、両方のバケットのトークンは、パケットのサイズによって減少します。

トークン補充レートは着信パケットによって異なります。あるパケットが時間 T1 に着信し、次のパケットが時間 T2 に着信したとします。T1 と T2 間の時間間隔は、トークン バケットに追加される必要があるトークンの数を決定します。これは次のように計算されます。

パケットの時間間隔（T2-T1） * CIR）/8 バイト

シェーピング

シェーピングは、ダウンストリーム スイッチおよびルータで輻輳が発生しないようにトラフィック レートを調整しながら、トラフィックの最大レートを強制するプロセスのことです。最も一般的な形式のシェーピングは、物理または論理インターフェイスから送信されるトラフィックを制限するために使用されます。

シェーピングにはバッファが関連付けられており、十分なトークンがないパケットがすぐにドロップされずにバッファされます。シェーピングされるトラフィックのサブセットで使用可能なバッファ数は制限され、さまざまな要因に基づいて計算されます。使用可能なバッファの数は、特定の QoS コマンドを使用して調整できます。パケットはドロップされずに、バッファが使用可能になった時点でバッファされます。

クラスベース トラフィック シェーピング

デバイスではクラスベースのトラフィック シェーピングを使用します。このシェーピング機能は、インターフェイスに関連付けられたポリシーのクラスでイネーブルになります。シェーピングが設定されたクラスには、トークンがないパケットを保持する複数のバッファが割り当てられます。バッファされたパケットは FIFO を使用してクラスから送信されます。最も一般的な形式の使用では、クラスベースのシェーピングを使用して、全体として物理インターフェイスまたは論理インターフェイスの最大レートを強制します。クラスでは次のシェーピング形式がサポートされます。

- 平均レート シェーピング
- 階層型シェーピング

シェーピングは、トークン バケットを使用して実行されます。CIR、B_c、B_e の値は、パケットが送信されるレートと、トークンが補充されるレートを決定します。

（注） トークン バケット アルゴリズムの詳細については、トークンバケットアルゴリズム（55 ページ）を参照してください。

平均レート シェーピング

平均レートシェーピングを設定するには、**shape average** ポリシーマップ クラス コマンドを使用します。

このコマンドは、特定のクラスの最大帯域幅を設定します。キューの帯域幅は、ポートでさらに使用できる帯域幅があってもこの値に制限されます。デバイスでは、割合またはターゲットビットレート値でシェーピング平均を設定できます。

階層型シェーピング

シェーピングは、階層内の複数のレベルで設定することもできます。これは、シェーピングを設定した親ポリシーを作成して、追加のシェーピングを設定した子ポリシーを親ポリシーに付加することで実現できます。

ポート シェーパーでは、クラス デフォルトが使用され、親で実行できるアクションはシェーピングだけです。キューイング アクションはポート シェーパーがある子で実行されます。ユーザー設定のシェーピングを使用すると、子のキューイング アクションを設定することはできません。

キューイングおよびスケジューリング

デバイスは、トラフィックの輻輳を防止するためにキューイングおよびスケジューリングを使用します。デバイスは、次のキューイングおよびスケジューリング機能をサポートします。

- 帯域幅

- 重み付けテール ドロップ
- プライオリティ キュー
- キュー バッファ
- 重み付けランダム早期検出

ポートにキューイング ポリシーを定義すると、制御パケットは、しきい値が最も高いベスト プライオリティ キューにマッピングされます。制御パケットのキュー マッピングは、以下の状況では異なって機能します。

- Quality of Service（QoS）ポリシーなし：QoS ポリシーが設定されていない場合、DSCP 値が 16、24、48、および 56 の制御パケットは、最も高いしきい値 threshold2 を持つキュー 0 にマッピングされます。
- ユーザー定義のポリシーあり：出力ポートに設定されているユーザー定義のキューイング ポリシーは、制御パケットのデフォルトのプライオリティ キューの設定に影響する可能性があります。

（注） 出力方向のキューイングポリシーは **match access-group** 分類をサポートしません。

制御トラフィックは、次のルールに基づいて最適なキューにリダイレクトされます。

1. ユーザー ポリシーで定義されている場合、最高レベルのプライオリティ キューがベスト キューとして常に選択されます。
2. プライオリティ キューがない場合、Cisco IOS ソフトウェアは、ベスト キューとしてキュー 0 を選択します。ソフトウェアがベスト キューとしてキュー 0 を選択した場合は、コントロール プレーン トラフィックに最適な QoS 処理を提供するために、このキューに最大帯域幅を定義する必要があります。
3. しきい値がベストキューで設定されていない場合、Cisco IOS ソフトウェアは、DiffServ コード ポイント（DSCP）値が 16、24、48、および 56 の制御パケットを threshold2 にマッピングされるように割り当て、ベスト キュー内の残りの制御トラフィックを threshold1 に再割り当てします。

ポリシーが制御トラフィックに対して明示的に設定されていない場合、Cisco IOS ソフトウェアはすべての一致しない制御トラフィックを threshold2 を持つベスト キューにマッピングし、一致する制御トラフィックはポリシーで設定されたキューにマッピングされます。

（注） レイヤ 3 パケットに適切な QoS を提供するために、パケットが適切なキューに明示的に分類されていることを確認する必要があります。ソフトウェアはデフォルト キューで DSCP 値を検出すると、自動的にこのキューをベスト キューとして再割り当てします。

帯域幅

デバイスは次の帯域幅設定をサポートしています。

- 帯域幅
- 帯域幅の割合
- 残存帯域幅の割合

帯域幅の割合

特定のクラスに最小帯域幅を割り当てるには、**bandwidth percent** ポリシーマップ クラス コマンドを使用します。合計が 100 % を超えることはできず、合計が 100 % 未満である場合は、残りの帯域幅がすべての帯域幅キューで均等に分割されます。

（注） キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。

ポリシー マップで帯域幅タイプを混在させることはできません。たとえば、1 つのポリシー マップで帯域幅の割合と kbps の両方を使用して、帯域幅を設定することはできません。

残存帯域幅の割合

指定されたキューでの未使用帯域幅の割合を作成するには、**bandwidth remaining percent** ポリシーマップ クラス コマンドを使用します。未使用帯域幅は、これら指定されたキューにより、設定で指定されている割合で使用されます。このコマンドは、**priority** コマンドがポリシー内の特定のキューでも使用される場合に使用します。

割合を割り当てる場合には、これらの割合に従って、キューに特定の重みが割り当てられます。

0 ー 100 の割合を指定できます。たとえば、1 つのクラスの帯域幅余剰割合を 2 に設定し、別のクラスで帯域幅余剰割合 4 のキューを設定できます。帯域幅余剰割合 4 は、帯域幅余剰割合 2 の 2 倍の頻度でスケジュールされます。

ポリシーの全帯域幅の割合の割り当ては 100 を超えることができます。たとえば、1 つのキューの帯域幅余剰割合を 50 に設定し、別のキューに帯域幅余剰割合 100 を設定できます。

重み付けテール ドロップ

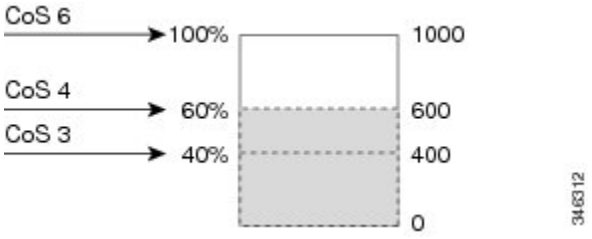
デバイス出力キューは、重み付けテールドロップ（WTD）と呼ばれるテールドロップ輻輳回避メカニズムの拡張バージョンを使用します。WTD はキュー長を管理したり、トラフィック分類ごとにドロップ優先順位を設定したりするために実装されています。

フレームが特定のキューにキューイングされると、WTD はフレームに割り当てられた QoS ラベルを使用して、それぞれ異なるしきい値を適用します。この QoS ラベルのしきい値を超えると（宛先キューの空きスペースがフレームサイズより小さくなると）、フレームはドロップされます。

各キューには 3 種類の設定可能なしきい値があります。QoS ラベルは、3 つのしきい値のうちのどれがフレームの影響を受けるかを決定します。

図 *2 : WTD* およびキューの動作

次の図は、サイズが 1000 フレームであるキューでの WTD の動作の例を示しています。ドロップ割合は次のように設定されています。40%（400 フレーム）、60%（600 フレーム）、および 100%（1000 フレーム）です。これらのパーセンテージは、40% しきい値の場合は最大 400 フレーム、60% しきい値の場合は最大 600 フレーム、100% しきい値の場合は最大 1000 フレーム



をキューイングできるという意味です。

例では、CoS 値 6 は他の CoS 値よりも重要度が高く、100 %のドロップしきい値（キューフルステート）に割り当てられます。CoS 値 4 は 60 % しきい値に、CoS 値 3 は 40 % しきい値に割り当てられます。これらのしきい値の割り当てはすべて、 **queue-limit cos** コマンドを使用します。

600 のフレームが格納されているキューに、新しいフレームが着信したとします。これは CoS 値 4 を使用し、60 % のしきい値が適用されます。このフレームがキューに追加されると、しきい値を超過するため、フレームはドロップされます。

重み付けテール ドロップのデフォルト値

次に、重み付けテール ドロップ（WTD）のデフォルト値と、WTD しきい値を設定するためのルールを示します。

- WTD に対して 2 つ以下のキュー制限割合を設定する場合、WTD のデフォルト値はこれらのしきい値に割り当てられます。

次に、WTD しきい値のデフォルト値を示します。

表 *5 : WTD* しきい値のデフォルト値

しきい値	デフォルト値の割合
0	80
1	90
2	400

- 異なる 3 つの WTD しきい値が設定されている場合、キューは設定どおりにプログラムされます。
- 2 つの WTD しきい値が設定されている場合、最大値の割合は 400 です。
- 1 つの WTD しきい値が x として設定されている場合、最大値の割合は 400 です。
  - x の値が 90 未満の場合、threshold1 = 90 および threshold 0= x です。
  - x の値が 90 の場合、threshold1 = 90、threshold 0 = 80 です。
  - x の値が 90 より大きい場合、threshold1 = x、threshold 0 = 80 です。

プライオリティ キュー

各ポートは 8 つの出力キューをサポートし、そのうち 2 つにプライオリティを設定できます。

2 つのクラスのプライオリティを設定するには、**priority level** ポリシー クラスマップ コマンドを使用します。1 つのクラスにプライオリティ キュー レベル 1 を設定し、別のクラスにプライオリティ キュー レベル 2 を設定する必要があります。これら 2 つのキューのパケットは、他のキューと比較して、低遅延になります。

プライオリティキューが設定されている場合は、100% のラインレートトラフィックを送信できません。プライオリティキューが設定されている場合、ラインレートトラフィックは 99.6% にしかならないため、遅延は 20 マイクロ秒未満になります。

（注） プライオリティは 1 つのレベルのみ設定できます。

1 つのポリシーマップで使用できる完全プライオリティまたはレベル付きプライオリティは 1 つだけです。kbps または割合のない同じプライオリティ レベルが設定された複数のプライオリティは、ポリシングですべてが設定された場合にのみ使用できます。

プライオリティ キュー ポリサー

このスイッチは、プライオリティキューのポリシングレートの設定をサポートします。プライオリティ キュー ポリサーは、シングルレート 2 カラーポリシングのみをサポートします。

（注） テーブルマップを使用したポリシングはサポートされません。

プライオリティ キュー ポリサーの設定例

例 1

```
Policy Map priority-1
    Class prio1
      priority level 1
     police rate percent 10
       conform-action transmit
       exceed-action drop
    Class prio2
      priority level 2
     police rate percent 5
       conform-action transmit
       exceed-action drop
    Class new
      bandwidth 20 (%)
```

例 2

```
Policy Map priority-1
    Class prio1
      priority level 1 20 (%)
     police rate percent 10
       conform-action transmit
       exceed-action drop
    Class prio2
      priority level 2 25 (%)
     police rate percent 5
       conform-action transmit
       exceed-action drop
    Class new
      bandwidth 20 (%)
```

キュー バッファ

ブート時に有線ポートでイネーブルになっているポリシーマップがない場合、デフォルトで作成される 2 つのキューがあります。有線ポートには、MQC ベースのポリシーを使用して最大 8 つのキューを設定できます。次の表に、どのパケットがどのキューに入っているかを示します。

表 *6 : DSCP*、*Precedence*、*CoS* : キューのしきい値のマッピング テーブル

DSCP、Precedence、CoS	キュー	しきい値
制御パケット	0	2

DSCP、Precedence、CoS	キュー	しきい値
他のパケット	1	2

（注） バッファのアベイラビリティを保証し、ドロップしきい値を設定し、キューの最大メモリ割り当てを設定できます。キューバッファを設定するには、 **queue-buffers** ポリシーマップ クラス コマンドを使用します。最大しきい値を設定するには、**queue-limit** ポリシーマップ クラス コマンドを使用します。

バッファ割り当ては2種類あります。キューに明示的に予約される厳格なバッファと、特定のポートで未使用時に他のポートで利用可能な柔軟なバッファです。

有線ポートのデフォルトでは、キュー0には、厳格なバッファとしてインターフェイスで利用可能なバッファの40%が割り当てられます。つまり、1ギガビットポートにおいては、キュー0に対して200バッファが割り当てられ、10ギガビットポートにおいては、600バッファが割り当てられます。このキューの柔軟な最大値は厳格なバッファの4倍に設定されます。つまり、1ギガビットポートの場合は800、10ギガビットポートの場合は2400、40ギガビットポートの場合は9600に設定されます。

キュー1に割り当てられた厳格なバッファはありません。柔軟なバッファの最小割り当ては、1ギガビットポートの場合は300バッファ、10ギガビットポートの場合は900バッファ、40ギガビットポートの場合は3600バッファです。キュー1の柔軟なバッファの最大割り当ては、柔軟なバッファの最小割り当ての4倍に設定されます。つまり、1ギガビットポートの場合は1200バッファ、10ギガビットポートの場合は3600バッファ、40ギガビットポートの場合は14400バッファです。

（注） デフォルトでは、キュー0はプライオリティキューではありません。ポリシーマップでは、**priority level** コマンドを使用して、キュー0をプライオリティキューにすることができます。キュー0にプライオリティレベル1が割り当てられている場合、このキューのソフト最大制限はハード最大制限と同じ値に自動的に設定されます。

C9300-24UB、C9300-24UXB、および C9300-48UB スイッチでのキューバッファ管理

Cisco Catalyst 9300 スイッチ、C9300-24UB、C9300-24UXB、および C9300-48UB は、C9300 シリーズの他のスイッチと比較して、速度の不一致によるマイクロ バースト トラフィックおよび輻輳を処理するパケットバッファ機能を拡張するために大きなバッファがプロビジョニングされています。スタッキングモードと非スタッキングモードを選択することにより、大きなバッファサイズを有効または無効にできます。

C9300-24UBおよびC9300-48UBスイッチのスタッキングモードと非スタッキングモードのバッファは、**qos stack-buffer** コマンドを使用して管理できます。

（注） C9300-24UXB は **qos stack-buffer** コマンドをサポートしていません。C9300-24UXB は常にスタッキングモードで起動します。

デフォルトでは、C9300-24UB、C9300-24UXB、C9300-48UB スイッチの起動時にスタッキングが有効になります。**qos stack-buffer disable** コマンドを実行してからスイッチをリロードすると、C9300-24UB および C9300-48UB スイッチをスタンドアロンモード（非スタッキングモード）で起動できます。スタンドアロンモードでは、スイッチのスタッキング機能は無効です。スタッキングモードに切り替えるには、**qos stack-buffer enable** コマンドを実行し、**write memory** コマンドを使用して設定を保存してから、デバイスをリロードします。

（注） C9300-24UB、C9300-24UXB、および C9300-48UB を C9300 シリーズの他のスイッチとスタックすることはできません。C9300-24UB、C9300-24UXB、および C9300-48UB スイッチは、相互にのみスタックできます。

キュー バッファの割り当て

キューに対するバッファ割り当ては、**queue-buffers ratio** ポリシーマップ クラス コンフィギュレーション コマンドを使用して調整できます。

ダイナミックなしきい値および拡張

従来、予約バッファは各キューに静的に割り当てられていました。キューがアクティブかどうかにかかわらず、バッファはキューに保持されます。さらに、キューの数が増えるに従って、各キューに割り当てられた予約バッファの部分が徐々に短くなることがあります。最終的に、すべてのキューのジャンボ フレームをサポートするのに十分な予約バッファがなくなる可能性があります。

デバイスは、バッファ リソースを公平かつ効率的に割り当てる機能として、ダイナミックなしきい値および拡張（DTS）をサポートしています。輻輳が発生すると、この DTS 機能はグローバル/ポート リソースの占有に基づいて、着信データにバッファを柔軟に割り当てます。概念上、DTS は、リソースを他のキューが使用できるように、キュー バッファの割り当てを徐々に縮小します。逆も同様です。この柔軟な方法によって、バッファをより効率的かつ公平に利用できるようになります。

前の項で説明したように、キューには厳格な制限と柔軟な制限の 2 つの制限が設定されています。

厳格な制限は DTS の一部ではありません。これらのバッファはそのキューにだけ使用できます。厳格な制限の合計は、グローバルに設定された厳格な最大制限未満である必要があります。出力キューイングのために設定されたグローバルな厳格な制限は、現在 5705 に設定されています。MQC ポリシーが設定されていないデフォルトのシナリオでは、24 の 1 ギガビット ポートが 24 * 67 = 1608 を使用し、4 つの 10 ギガビット ポートが 4 * 720 = 2880 を使用し、合計 4488 のバッファを使用して、設定に基づいてより厳格なバッファを割り当てることができます。

柔軟なバッファ制限は DTS プロセスに参加します。さらに、柔軟なバッファ割り当ての一部は、グローバルな柔軟な制限の割り当てを超えることができます。出力キューイング用のグローバルな柔軟な制限は、現在27024に設定されています。厳格な制限と柔軟な制限の合計は39696になり、10.1 MB に変換されます。柔軟なバッファ割り当ての合計がグローバルな制限を超える場合があるため、システムの負荷が軽ければ、特定のキューで多数のバッファを使用できるようになります。DTSプロセスはシステムの負荷が増大するにしたがって、キュー単位の割り当てを動的に調整します。

統合バッファ共有

Cisco IOS XE 17.2.1 リリース以降では、同じ ASIC 内の 2 つのコア間でアクティブキュー管理（AQM）バッファの共有を設定できます。バッファ共有が設定されたポートは、AQM バッファがマッピングされているコアに関係なく、使用可能な AQM バッファのいずれかを使用できます。これにより、単一の AQM コアのバッファでは飽和していた大量のトラフィックのバーストを管理できます。

この機能を有効にするには、**qos share-buffer** コマンドを使用します。**show plat hardware fed active qos queue config interface** コマンドを使用すると、バッファ共有が有効になっているかどうかを確認できます。これは、システム全体に影響するグローバルコンフィギュレーションになります。バッファ共有を無効にするには、**no qos share-buffer** コマンドの **no** 形式を使用します。

重み付けランダム早期検出

重み付けランダム早期検出（WRED）は、ネットワークでの輻輳を回避するメカニズムです。WREDは、出力インターフェイスにネットワーク混雑の兆候が表れた際に、選択的にパケットをドロップしてテールドロップの確率を減らし、多数のパケットが一度にドロップされないようにします。

WRED の詳細については、次を参照してください。重み付けランダム早期検出の設定 （137 ページ）

信頼動作

Cisco IP Phone の信頼境界機能のポート セキュリティ

一般的なネットワークでは、デバイスポートに Cisco IP Phone を接続し、電話の背後からデータパケットを生成するデバイスをカスケードします。Cisco IP Phone では、音声パケット CoS レベルをハイ プライオリティ（CoS＝5）にマーキングし、データ パケットをロー プライオリティ（CoS＝0）にマーキングすることで、共有データ リンクを通して音声品質を保証しています。電話からデバイスに送信されたトラフィックは通常 802.1Q ヘッダーを使用するタグでマーキングされています。ヘッダーには VLAN 情報およびパケットのプライオリティになる CoS の 3 ビット フィールドが含まれています。

ほとんどの Cisco IP Phone 設定では、電話からデバイスへ送信されるトラフィックは、音声トラフィックがネットワーク内の他のタイプのトラフィックに対して適切にプライオリティ付けがされていることを保証するように信頼されています。 **trust device** インターフェイス コンフィギュレーション コマンドを使用して、電話の接続先のデバイス ポートが受信トラフィックを信頼するように設定します。

（注） インターフェイス コンフィギュレーション モードで使用可能な **trust device** *device_type* コマンドは、デバイスでのスタンドアロン コマンドです。このコマンドを AutoQoS 設定で使用するときに、接続されているピア デバイスが対応デバイス（信頼ポリシーに一致するデバイスとして定義されているデバイス）ではない場合、CoS 値と DSCP 値の両方が「0」に設定され、いずれの入力ポリシーも有効になりません。接続されているピア デバイスが対応するデバイスである場合は、入力ポリシーが有効になります。

信頼設定により、ユーザが電話をバイパスして PC を直接デバイスに接続する場合に、ハイプライオリティ キューの誤使用を避けるため信頼境界機能も使用できます。信頼境界機能を使用しないと、（信頼性のある CoS 設定により）PC が生成した CoS ラベルがデバイスで信頼されてしまいます。それに対して、信頼境界機能は CDP を使用してデバイス ポートにある Cisco IP Phone（Cisco IP Phone 7910、7935、7940、7960 など）の存在を検出します。電話が検出されない場合、信頼境界機能がハイプライオリティ キューの誤使用を避けるためにデバイス ポートの信頼設定をディセーブルにします。信頼境界機能は、PC および Cisco IP Phone がデバイスに接続されているハブに接続されている場合は機能しないことに注意してください。

有線ポートの信頼動作

次の表に、着信パケット タイプが発信パケット タイプと異なる場合の信頼動作およびキューイング動作を示します。ポートのデフォルトの信頼モードが DSCP ベースであることに注意してください。信頼モードは、着信パケットが純粋なレイヤ 2 パケットの場合、CoS に「フォールバック」します。また、信頼設定を DSCP から CoS に変更できます。この設定変更は、「set cos cos table default default-cos」アクションのクラス デフォルトがある MQC ポリシーによって実現されます。ここで、default-cos は作成されるテーブル マップ名です（デフォルト コピーだけを実行）。

デバイス（IP フォン、ラップトップ、カメラ、TelePresence ユニットなどのエンドポイント、またはその他のデバイス）に接続されている有線ポートの場合、インターフェイス上で信頼デバイス設定が有効になります。明示的なポリシー設定がない場合、これらのエンドポイントから、またはこれらエンドポイントへの DSCP 値、precedence 値、または CoS 値はデバイスで信頼されるため、保持されます。

パケットはデフォルトの初期設定ごとに適切なキューに入れられます。デフォルトでは、デバイスでのプライオリティ キューイングは実行されません。これは、ユニキャストおよびマルチキャスト パケットに当てはまります。

表 *7:* 信頼およびキューイング動作

着信パケット	発信パケット	信頼動作	キューイング動作
レイヤ 3	レイヤ 3	DSCP/Precedence の保持	DSCP に基づく
レイヤ 2	レイヤ 2	N/A	CoS に基づく
タグ付き	タグ付き	DSCP および CoS の保持	DSCP に基づく（信頼 DSCP が優先）
レイヤ 3	タグ付き	DSCP の保持、すなわち CoS が 0 に設定される	DSCP に基づく

標準 QoS のデフォルト設定

デフォルトの有線 QoS 設定

デバイスの各有線インターフェイスでは、デフォルトで2つのキューが設定されます。すべての制御トラフィックはキュー0を通過し、処理されます。その他すべてのトラフィックはキュー1を通過し、処理されます。

DSCP マップ

デフォルトの CoS/DSCP マップ

DSCP 透過モードを無効にすると、DSCP 値は次の表に従って CoS から抽出されます。これらの値が使用しているネットワークに適さない場合は、値を変更する必要があります。

（注） DSCP 透過モードはデフォルトでは無効になっています。これがイネーブルになっている場合（**mls qos rewrite ip dscp** コンフィギュレーション コマンド）、DSCP の書き換えは実行されません。

表 *8:* デフォルトの *CoS/DSCP* マップ

CoS 値	DSCP 値
0	0
1	8
2	16
3	24
4	32
5	40

CoS 値	DSCP 値
6	48
7	56

デフォルトの IP Precedence/DSCP マップ

着信パケットの IP precedence 値を、QoS がトラフィックのプライオリティを表すために内部使用する DSCP 値にマッピングするには、IP precedence/DSCP マップを使用します。次の表は、デフォルトの IP Precedence/DSCP マップを示しています。これらの値が使用しているネットワークに適さない場合は、値を変更する必要があります。

表 *9*：デフォルトの *IP Precedence/DSCP* マップ

IP precedence 値	DSCP 値
0	0
1	8
2	16
3	24
4	32
5	40
6	48
7	56

デフォルトの DSCP/CoS マップ

4 つの出力キューのうち 1 つを選択するために使用される CoS 値を生成するには、DSCP/CoS マップを使用します。次の表に、デフォルトの DSCP/CoS マップを示します。これらの値が使用しているネットワークに適さない場合は、値を変更する必要があります。

表 *10*：デフォルトの *DSCP/CoS* マップ

DSCP 値	CoS 値
0 ～ 7	0
8 ～ 15	1
16 ～ 23	2
24 ～ 31	3
32 ～ 39	4

DSCP 値	CoS 値
40 ～ 47	5
48 ～ 55	6
56 ～ 63	7

QoS の設定方法

クラス、ポリシー、およびマップの設定

トラフィック クラスの作成

一致基準が含まれるトラフィッククラスを作成するには、**class-map** コマンドを使用してトラフィッククラス名を指定し、必要に応じて、次の **match** コマンドをクラスマップ コンフィギュレーション モードで使用します。

始める前に

この設定作業で指定するすべての match コマンドの使用は任意ですが、1 つのクラスに少なくとも 1 つの一致基準を設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**class-map** *class-map name* {**match-any** | **match-all**}
	例：
	`Device(config)# class-map test_1000`
	`Device(config-cmap)#`
		クラス マップ コンフィギュレーション モードを開始します。
		• 名前を指定したクラスとパケットとの照合に使用されるクラス マップを作成します。
		• **match-any**：トラフィック クラスで受信したトラフィックがその一部と分類されるには、一致基準のいずれかを満たす必要があります。

	コマンドまたはアクション	目的
		• **match-all**：トラフィック クラスで受信したトラフィックがトラフィック クラスの一部と分類されるには、すべての一致基準を満たす必要があります。 （注） これはデフォルトです。**match-any** または **match-all** が明示的に定義されていない場合、デフォルトで **match-all** が選択されます。

ステップ 3	**match access-group** {*index number* | *name*}
	例：
	`Device(config-cmap)# match access-group 100`
	`Device(config-cmap)#`
		このコマンドでは次のパラメータを使用できます。
		• access-group
		• cos
		• dscp
		• group-object
		• ip
		• mpls
		• non-client-nrt
		• precedence
		• protocol
		• qos-group
		• vlan
		• wlan
		（任意）この例では、アクセス グループ ID を入力します。
		• アクセス リスト インデックス（1 ～ 2799 の値）
		• 名前付きアクセス リスト

	コマンドまたはアクション	目的

ステップ 4	**match cos** *CoS* 値
	例：
	`Device(config-cmap)# match cos 2 3 4 5`
	`Device(config-cmap)#`
		（任意）IEEE 802.1Q または ISL サービス クラス（ユーザー）プライオリティ値に一致します。
		• 最大 4 つの CoS 値（0 ～ 7）をスペースで区切って入力します。

ステップ 5	**match dscp** *DSCP* 値
	例：
	`Device(config-cmap)# match dscp af11 af12`
	`Device(config-cmap)#`
		（任意）IPv4 および IPv6 パケットの DSCP 値に一致します。

ステップ 6	**match ip** {**dscp** *dscp value* | **precedence** *precedence value* }
	例：
	`Device(config-cmap)# match ip dscp af11 af12`
	`Device(config-cmap)#`
		（任意）次を含む IP 値に一致します。
		• **dscp**：IP DSCP（DiffServ コードポイント）に一致します。
		• **precedence**：IP precedence（0 ～ 7）に一致します。
		（注） CPU 生成パケットは出力時にマークされないので、パケットは設定されたクラスマップと一致しません。

ステップ 7	**match qos-group** *QoS グループ値*
	例：
	`Device(config-cmap)# match qos-group 10`
	`Device(config-cmap)#`
		（任意）QoS グループ値（0 ～ 31）に一致します。

ステップ 8	**match vlan** *vlan value*
	例：
	`Device(config-cmap)# match vlan 210`
	`Device(config-cmap)#`
		（任意）VLAN ID（1 ～ 4095）に一致します。

ステップ 9	**end** 例：	設定の変更内容を保存します。

	コマンドまたはアクション	目的
	`Device(config-cmap)# end`	

次のタスク

ポリシー マップを設定します。

トラフィック ポリシーの作成

トラフィックポリシーを作成するには、**policy-map** グローバル コンフィギュレーション コマンドを使用して、トラフィックポリシーの名前を指定します。

トラフィッククラスは、**class** コマンドを使用したときにトラフィックポリシーと関連付けられます。**class** コマンドは、ポリシー マップ コンフィギュレーション モードを開始した後に実行しなければなりません。**class** コマンドを入力すると、デバイスが自動的にポリシー マップ クラス コンフィギュレーション モードを開始します。ここでトラフィックポリシーの QoS ポリシーを定義します。

次のポリシー マップ クラスのアクションがサポートされます。

- bandwidth：帯域幅設定オプション。
- exit：QoS クラス アクション コンフィギュレーション モードを終了します。
- no：コマンドのデフォルト値を無効にするか、設定します。
- police：ポリシング機能の設定オプション。
- priority：このクラスの完全スケジューリング プライオリティの設定オプション。
- queue-buffers：キューのバッファ設定オプション。
- queue-limit：重み付けテール ドロップ（WTD）設定オプションのキューの最大しきい値。
- service-policy：QoS サービス ポリシーを設定します。
- set：次のオプションを使用して QoS 値を設定します。
  - CoS 値
  - DSCP 値
  - precedence 値
  - QoS グループ値
- shape：トラフィック シェーピング設定オプション。

始める前に

最初にクラス マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**policy-map** *policy-map name*
	例：
	`Device(config)# policy-map test_2000`
	`Device(config-pmap)#`
		ポリシーマップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシーマップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** {*class-name* | **class-default**}
	例：
	`Device(config-pmap)# class test_1000`
	`Device(config-pmap-c)#`
		ポリシーを作成または変更するクラスの名前を指定します。
		未分類のパケットのシステムデフォルト クラスも作成できます。

ステップ 4	**bandwidth** { *Kb/s* | **percent** *percentage* | **remaining** {*percent* | *ratio*} }
	例：
	`Device(config-pmap-c)# bandwidth 500`
	`Device(config-pmap-c)#`
		（任意）次のいずれかを使用して帯域幅を設定します。
		• *Kb/s*：キロビット/秒。Kb/s に 100 ～ 100000000 の値を入力します。
		• **percent**：このポリシーマップに使用される総帯域幅の割合を入力します。
		• **remaining**：残りの帯域幅の割合を入力します。
		このコマンドおよび使用の詳細な例については、帯域幅の設定（101 ページ）を参照してください。

ステップ 5	**exit**
	例：
	`Device(config-pmap-c)# exit`
	`Device(config-pmap-c)#`
		（任意）QoS クラス アクション コンフィギュレーション モードを終了します。

ステップ 6	**no** 例：	（任意）コマンドを無効にします。

	コマンドまたはアクション	目的
	`Device(config-pmap-c)# no` `Device(config-pmap-c)#`	

ステップ 7	**police** {*target_bit_rate* | **cir** | **rate**}
	例：
	`Device(config-pmap-c)# police 100000`
	`Device(config-pmap-c)#`
		（任意）ポリサーを設定します。
		• *target_bit_rate*：ビットレート/秒を入力します。8000 ～ 10000000000 の値を入力します。
		• **cir**：認定情報レート。
		• **rate**：ポリシングレート、階層型ポリシーの PCR、またはシングルレベルの ATM 4.0 ポリサー ポリシーの SCR を指定します。
		このコマンドおよび使用の詳細な例については、ポリシングの設定 （103 ページ）を参照してください。

ステップ 8	**priority** {*kb/s* | **level** *level value* | **percent** *percentage value*}
	例：
	`Device(config-pmap-c)# priority level 1 percent 50`
	`Device(config-pmap-c)#`
		（任意）このクラスに完全スケジューリング プライオリティを設定します。コマンド オプションは次のとおりです。
		• *kb/s*：キロビット/秒。1 ～ 2000000 の値を入力します。
		• **level**：マルチレベル プライオリティ キューを確立します。値を入力します（1 または 2）。
		• **percent**：このプライオリティの全帯域幅の割合を入力します。
		このコマンドおよび使用の詳細な例については、プライオリティの設定 （106 ページ）を参照してください。

ステップ 9	**queue-buffers ratio***ratio limit*
	例：
	`Device(config-pmap-c)# queue-buffers ratio 10`
	`Device(config-pmap-c)#`
		（任意）クラスのキューバッファを設定します。キューバッファの割合制限（0 ～ 100）を入力します。
		このコマンドおよび使用の詳細な例については、キューバッファの設定 （111 ページ）を参照してください。

	コマンドまたはアクション	目的

ステップ 10	**queue-limit** {*packets* | **cos** | **dscp** | **percent**}
	例：
	`Device(config-pmap-c)# queue-limit cos 7 percent 50`
	`Device(config-pmap-c)#`
		（任意）テール ドロップに対してキューの最大しきい値を指定します。
		• *packets*：デフォルトのパケット数。1 ～ 2000000 の値を入力します。
		• **cos**：各 CoS 値のパラメータを入力します。
		• **dscp**：各 DSCP 値のパラメータを入力します。
		• **percent**：しきい値の割合を入力します。
		このコマンドおよび使用の詳細な例については、キュー制限の設定（115 ページ）を参照してください。

ステップ 11	**service-policy** *policy-map name*
	例：
	`Device(config-pmap-c)# service-policy test_2000`
	`Device(config-pmap-c)#`
		（任意）QoS サービス ポリシーを設定します。

ステップ 12	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan**}
	例：
	`Device(config-pmap-c)# set cos 7`
	`Device(config-pmap-c)#`
		（任意）QoS 値を設定します。使用可能な QoS 設定値は次のとおりです。
		• **cos**：IEEE 802.1Q/ISL サービスクラスまたはユーザープライオリティを設定します。
		• **dscp**：IP(v4) および IPv6 パケットに DSCP を設定します。
		• **ip**：IP 固有の値を設定します。
		• **precedence**：IP(v4)、IPv6 パケットの優先順位を設定します。
		• **qos-group**：QoS グループを設定します。

ステップ 13	**shape average** {*target _bit_rate* | **percent**} 例：	（任意）トラフィック シェーピングを設定します。コマンドパラメータは次のとおりです。

	コマンドまたはアクション	目的
	`Device(config-pmap-c) #shape average percent 50` `Device(config-pmap-c) #`	• *target_bit_rate*：ターゲットビットレート。 • **percent**：認定情報レートのインターフェイス帯域幅の割合。 このコマンドおよび使用の詳細な例については、シェーピングの設定 （118 ページ）を参照してください。

ステップ 14	**end**
	例：
	`Device(config-pmap-c) #end`
	`Device(config-pmap-c) #`
		設定の変更内容を保存します。

次のタスク

インターフェイスを設定します。

クラスベース パケット マーキングの設定

この手順は、次のクラスベース パケット マーキング機能をデバイスで設定する方法を示します。

- CoS 値
- DSCP 値
- IP 値
- precedence 値
- QoS グループ値
- WLAN 値

始める前に

この手順を開始する前にクラス マップとポリシー マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device# configure terminal`	

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy1`
	`Device(config-pmap)#`
		ポリシーマップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシーマップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class1`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーションモードを開始します。ポリシーを作成または変更するクラスの名前を指定します。
		ポリシー クラス マップ コンフィギュレーションモードには、次のコマンドオプションが含まれます。
		• **bandwidth**：帯域幅設定オプション。
		• **exit**：QoS クラス アクション コンフィギュレーションモードを終了します。
		• **no**：コマンドのデフォルト値を無効にするか、設定します。
		• **police**：ポリシング機能の設定オプション。
		• **priority**：このクラスの完全スケジューリング プライオリティの設定オプション。
		• **queue-buffers**：キューのバッファ設定オプション。
		• **queue-limit**：重み付けテール ドロップ（WTD）設定オプションのキューの最大しきい値。
		• **service-policy**：QoS サービス ポリシーを設定します。
		• **set**：次のオプションを使用して QoS 値を設定します。

	コマンドまたはアクション	目的
		• CoS 値 • DSCP 値 • precedence 値 • QoS グループ値 • WLAN 値 • **shape**：トラフィック シェーピング設定オプション。 （注） この手順では、**set** コマンドオプションを使用して、使用可能な設定について説明します。その他のコマンドオプション（**bandwidth**）についてはこのマニュアルの他の項で説明します。このタスクでは、使用可能なすべての**set**コマンドが表示されますが、クラス単位でサポートされるのは 1 つの **set** コマンドだけです。

ステップ 4	例：
	`Device(config-pmap)# set cos 5`
	`Device(config-pmap)#`
		（任意）発信パケットの固有の IEEE 802.1Q レイヤ 2 CoS 値を設定します。値は 0 ～ 7 です。
		set cos コマンドを使用して次の値を設定することもできます。
		• **cos table**：CoS 値をテーブル マップに基づいて設定します。
		• **dscp table**：コード ポイント値をテーブルマップに基づいて設定します。
		• **precedence table**：コードポイント値をテーブルマップに基づいて設定します。
		• **qos-group table**：テーブル マップに基づいて QoS グループから CoS 値を設定します。

	コマンドまたはアクション	目的

ステップ 5	例：
	
	`Device(config-pmap)# set dscp af11`
	`Device(config-pmap)#`
		（任意）DSCP 値を設定します。
		
		特定の DSCP 値の設定に加えて、**set dscp** コマンドを使用して次を設定できます。
		
		• **default**：パケットをデフォルト DSCP 値（000000）と一致させます。
		
		• **dscp table**：テーブル マップに基づいて DSCP からパケットの DSCP 値を設定します。
		
		• **ef**：パケットを EF DSCP 値（101110）と一致させます。
		
		• **precedence table**：テーブル マップに基づいて優先順位からパケットの DSCP 値を設定します。
		
		• **qos-group table**：テーブル マップに基づいて QoS グループからパケットの DSCP 値を設定します。

ステップ 6	**set ip** {**dscp** | **precedence**}
	
	例：
	
	`Device(config-pmap)# set ip dscp c3`
	`Device(config-pmap)#`
		（任意）IP 固有の値を設定します。これらの値は、IP DSCP 値または IP precedence 値です。
		
		set ip dscp コマンドを使用して、次の値を設定することができます。
		
		• *dscp value*：特定の DSCP の値を設定します。
		
		• **default**：パケットをデフォルト DSCP 値（000000）と一致させます。
		
		• **dscp table**：テーブル マップに基づいて DSCP からパケットの DSCP 値を設定します。
		
		• **ef**：パケットを EF DSCP 値（101110）と一致させます。
		
		• **precedence table**：テーブル マップに基づいて優先順位からパケットの DSCP 値を設定します。

	コマンドまたはアクション	目的
		• **qos-group table**：テーブル マップに基づいて QoS グループからパケットの DSCP 値を設定します。 **set ip precedence** コマンドを使用して、次の値を設定することができます。 • *precedence value*：precedence 値を設定します（0 ～ 7）。 • **cos table**：テーブル マップに基づいてレイヤ 2 CoS からパケットの precedence 値を設定します。 • **dscp table**：テーブルマップに基づいて DSCP 値からパケットの precedence 値を設定します。 • **precedence table**：テーブルマップに基づいて優先順位から precedence 値を設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループから precedence 値を設定します。

ステップ 7	**set precedence** {*precedence value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name*}
	
	例：
	
	`Device(config-pmap)# set precedence 5`
	`Device(config-pmap)#`
		（任意）IPv4 と IPv6 パケットの precedence 値を設定します。
		
		set precedence コマンドを使用して、次の値を設定することができます。
		• *precedence value*：precedence 値を設定します（0 ～ 7）。
		• **cos table**：レイヤ 2 CoS からのパケットの precedence 値をテーブルマップに基づいて設定します。
		• **dscp table**：テーブルマップに基づいて DSCP 値からパケットの precedence 値を設定します。
		• **precedence table**：テーブルマップに基づいて優先順位から precedence 値を設定します。

	コマンドまたはアクション	目的
		• **qos-group table**：テーブル マップに基づいて QoS グループから precedence 値を設定します。

ステップ 8	**set qos-group** {*qos-group value* | **dscp table** *table-map name* | **precedence table** *table-map name*}
	例：
	`Device(config-pmap)# set qos-group 10`
	`Device(config-pmap)#`
		（任意）QoS グループ値を設定します。このコマンドを使用して次の値を設定できます。
		• *qos-group value*：1 から 31 までの数。
		• **dscp table**：テーブルマップに基づいて DSCP からコード ポイント値を設定します。
		• **precedence table**：テーブル マップに基づいて優先順位からコードポイント値を設定します。

ステップ 9	**end**
	例：
	`Device(config-pmap)# end`
	`Device#`
		設定変更を保存します。

ステップ 10	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

service-policy コマンドを使用して、インターフェイスにトラフィック ポリシーを付加します。

トラフィック ポリシーのインターフェイスへの適用

トラフィッククラスとトラフィックポリシーの作成後、**service-policy** インターフェイス コンフィギュレーション コマンドを使用して、トラフィックポリシーをインターフェイスに付加し、ポリシーを適用する方向を指定します（インターフェイスに着信するパケットまたはインターフェイスから送信されるパケット）。

始める前に

インターフェイスにトラフィック ポリシーを付加する前に、トラフィック クラスとトラフィック ポリシーを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *type*
	例：
	`Device(config)# interface GigabitEthernet1/0/1`
	`Device(config-if)#`
		インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。
		インターフェイス コンフィギュレーションのコマンド パラメータは次のとおりです。
		• **ANI**：自律型ネットワーキング仮想インターフェイス
		• **AccessTunnel**：アクセス トンネル インターフェイス
		• **Auto Template**：自動テンプレート インターフェイス
		• **CEM-PG**：保護グループを持つ回線エミュレーション インターフェイス
		• **FortyGigabitEthernet**：40 ギガビット イーサネット
		• **GigabitEthernet**：Gigabit Ethernet IEEE 802.3z
		• **Internal Interface**：内部インターフェイス
		• **LISP**：ロケータ ID 分離プロトコル仮想インターフェイス
		• **Loopback**：ループバック インターフェイス
		• **Null**：ヌル インターフェイス
		• **PROTECTION_GROUP**：保護グループ コントローラ
		• **Port-channel**：インターフェイスのイーサネット チャネル

	コマンドまたはアクション	目的
		• **SDH_ACR**：仮想 SDH-ACR コントローラ • **TLS-VIF**：TLS 仮想インターフェイス • **TenGigabitEthernet**：10 ギガビットイーサネット • **Tunnel**：トンネル インターフェイス • **Tunnel-tp**：MPLS トランスポート プロファイル インターフェイス • **Vlan**：Catalyst VLAN • **Range**：インターフェイス範囲 （注） トンネルインターフェイスはサポートされていません。

ステップ 3	**service-policy** { **input** *policy-map* | **output** *policy-map*}
	例：
	`Device(config-if)# service-policy output policy_map_01`
	`Device(config-if)#`
		ポリシー マップを入力または出力インターフェイスに適用します。このポリシー マップは、そのインターフェイスのサービス ポリシーとして使用されます。
		この例では、トラフィック ポリシーでそのインターフェイスから送信されるすべてのトラフィックを評価します。

ステップ 4	**end**
	例：
	`Device(config-if)# end`
	`Device#`
		設定変更を保存します。

ステップ 5	**show policy map** 例： `Device# show policy map`	（任意）指定されたインターフェイスのポリシーの統計情報を表示します。

次のタスク

他のトラフィック ポリシーをインターフェイスに付加し、ポリシーを適用する方向を指定します。

ポリシーマップによる物理ポートのトラフィックの分類、ポリシング、およびマーキング

実行対象となるトラフィック クラスを指定する非階層型ポリシー マップを、物理ポート上に設定できます。サポートされるアクションは再マーキングとポリシングです。

始める前に

この手順を開始する前に、ネットワーク トラフィックの分類、ポリシング、およびマーキングについて、あらかじめポリシー マップによって決定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**class-map** {*class-map name* | **match-any** | **match-all**}
	例：
	`Device(config)# class-map ipclass1`
	`Device(config-cmap)# exit`
	`Device(config)#`
		クラスマップ コンフィギュレーション モードを開始します。
		• 名前を指定したクラスとパケットとの照合に使用されるクラスマップを作成します。
		• **match-any** を指定すると、トラフィック クラスで受信したトラフィックの場合、一致基準の 1 つに必ず一致し、そのトラフィック クラスの一部と分類されます。これはデフォルトです。
		• **match-all** を指定すると、トラフィック クラスで受信したトラフィックがトラフィック クラスの一部と分類されるには、すべての一致基準を満たす必要があります。

	コマンドまたはアクション	目的
		（注） これはデフォルトです。**match-any** または **match-all** が明示的に定義されていない場合、デフォルトで **match-all** が選択されます。

ステップ 3	**match access-group** { *access list index* | *access list name* }
	例：
	`Device(config-cmap)# match access-group 1000`
	`Device(config-cmap)# exit`
	`Device(config)#`
		このコマンドでは次のパラメータを使用できます。
		• access-group
		• cos
		• dscp
		• group-object
		• ip
		• mpls
		• non-client-nrt
		• precedence
		• protocol
		• qos-group
		• vlan
		• wlan
		（任意）この例では、アクセス グループ ID を入力します。
		• アクセス リスト インデックス（1 ～ 2799 の値）
		• 名前付きアクセス リスト

ステップ 4	**policy-map** *policy-map-name*
	例：
	`Device(config)# policy-map flowit`
	`Device(config-pmap)#`
		ポリシー マップ名を入力することによってポリシーマップを作成し、ポリシーマップ コンフィギュレーション モードを開始します。
		デフォルトでは、ポリシーマップは定義されていません。

	コマンドまたはアクション	目的

ステップ 5	**class** {*class-map-name* | **class-default**}
	例：
	`Device(config-pmap)# class ipclass1`
	`Device(config-pmap-c)#`
		トラフィックの分類を定義し、ポリシーマップ クラス コンフィギュレーション モードを開始します。
		デフォルトでは、ポリシーマップ クラス マップは定義されていません。
		すでに **class-map** グローバル コンフィギュレーションコマンドを使用してトラフィッククラスが定義されている場合は、このコマンドで *class-map-name* にその名前を指定します。
		class-default トラフィッククラスは定義済みで、どのポリシーにも追加できます。このトラフィック クラスは、常にポリシー マップの最後に配置されます。暗黙の **match any** が **class-default** クラスに含まれている場合、他のトラフィッククラスと一致しないパケットはすべて **class-default** と一致します。

ステップ 6	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan user-priority**}
	例：
	`Device(config-pmap-c)# set dscp 45`
	`Device(config-pmap-c)#`
		（任意）QoS 値を設定します。使用可能な QoS 設定値は次のとおりです。
		• **cos**：IEEE 802.1Q/ISL サービス クラスまたはユーザー プライオリティを設定します。
		• **dscp**：IP（v4）および IPv6 パケットの DSCP を設定します。
		• **ip**：IP 固有の値を設定します。
		• **precedence**：IP（v4）および IPv6 パケットの precedence を設定します。
		• **qos-group**：QoS グループを設定します。
		この例では、 **set dscp** コマンドが、パケットでの新しい DSCP 値を設定して IP トラフィックを分類します。

ステップ 7	**police** {*target_bit_rate* | **cir** | **rate** } 例：	（任意）ポリサーを設定します。

	コマンドまたはアクション	目的
	`Device(config-pmap-c)# police 100000 conform-action transmit exceed-action drop` `Device(config-pmap-c)#`	• *target_bit_rate*：ビット レート/秒を指定し、8000 ～ 10000000000 の値を入力します。 • **cir**：認定情報レート。 • **rate**：階層型ポリシーのポリシング レート PCR を指定します。 この例では、**police** コマンドが 100000 セットのターゲットビットレートを超えるトラフィックがドロップされるクラスにポリサーを追加します。
ステップ 8	**exit** 例： `Device(config-pmap-c)# exit`	ポリシーマップ コンフィギュレーション モードに戻ります。
ステップ 9	**exit** 例： `Device(config-pmap)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 10	**interface** *interface-id* 例： `Device(config)# interface HundredGigabitEthernet 1/0/2`	ポリシーマップを適用するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。 有効なインターフェイスには、物理ポートが含まれます。
ステップ 11	**service-policy input** *policy-map-name* 例： `Device(config-if)# service-policy input flowit`	ポリシーマップ名を指定し、入力ポートに適用します。サポートされるポリシー マップは、入力ポートに 1 つだけです。
ステップ 12	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	（任意）入力を確認します。

	コマンドまたはアクション	目的
	例： `Device# show policy-map`	
ステップ 14	**copy running-config startup-config** 例： `Device# copy-running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

次のタスク

必要に応じて QoS 設定は、ポリシー マップを使用して、SVI のトラフィックの分類、ポリシング、およびマーキングを設定します。

ポリシーマップによるトラフィックの分類およびマーキング

始める前に

この手順を開始する前に、ポリシー マップを使用して、ネットワーク トラフィックの分類、ポリシング、およびマーキングについて決定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**class-map** {*class-map name* | **match-any** | **match-all**} 例： `Device(config)# class-map class_vlan100`	クラスマップ コンフィギュレーション モードを開始します。 • 名前を指定したクラスとパケットとの照合に使用されるクラスマップを作成します。 • **match-any** を指定すると、トラフィック クラスで受信したトラフィックの場合、一致基準の 1 つに必ず一致し、そのトラフィック クラスの一部と分類されます。

	コマンドまたはアクション	目的
		• **match-all** を指定すると、トラフィック クラスで受信したトラフィックがトラフィック クラスの一部と分類されるには、すべての一致基準を満たす必要があります。 （注） これはデフォルトです。**match-any** または **match-all** が明示的に定義されていない場合、デフォルトで **match-all** が選択されます。

ステップ 3	**match vlan** *vlan number*
	例：
	`Device(config-cmap)# match vlan 100`
	`Device(config-cmap)# exit`
	`Device(config)#`
		VLAN をクラス マップに一致するように指定します。

ステップ 4	**policy-map** *policy-map-name*
	例：
	`Device(config)# policy-map policy_vlan100`
	`Device(config-pmap)#`
		ポリシー マップ名を入力することによってポリシーマップを作成し、ポリシーマップ コンフィギュレーション モードを開始します。
		デフォルトでは、ポリシーマップは定義されていません。

ステップ 5	**description** 説明 例： `Device(config-pmap)# description vlan 100`	（任意）ポリシーマップの説明を入力します。

ステップ 6	**class** {*class-map-name* | **class-default**}
	例：
	`Device(config-pmap)# class class_vlan100`
	`Device(config-pmap-c)#`
		トラフィック分類を定義し、ポリシーマップ クラス コンフィギュレーション モードを開始します。
		デフォルトでは、ポリシーマップ クラス マップは定義されていません。
		すでに **class-map** グローバル コンフィギュレーション コマンドを使用してトラフィック クラスが定義されている場

	コマンドまたはアクション	目的
		合は、このコマンドで *class-map-name* にその名前を指定します。 **class-default** トラフィッククラスは定義済みで、どのポリシーにも追加できます。このトラフィック クラスは、常にポリシー マップの最後に配置されます。暗黙の **match any** が **class-default** クラスに含まれている場合、他のトラフィッククラスと一致しないパケットはすべて **class-default** と一致します。

ステップ 7	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan user-priority**}
	
	例：
	
	`Device(config-pmap-c)# set dscp af23`
	`Device(config-pmap-c)#`
		（任意）QoS 値を設定します。使用可能な QoS 設定値は次のとおりです。
		
		• **cos**：IEEE 802.1Q/ISL サービス クラスまたはユーザー プライオリティを設定します。
		
		• **dscp**：IP（v4）および IPv6 パケットの DSCP を設定します。
		
		• **ip**：IP 固有の値を設定します。
		
		• **precedence**：IP（v4）および IPv6 パケットの precedence を設定します。
		
		• **qos-group**：QoS グループを設定します。
		
		この例では、**set dscp** コマンドが AF23（010010）の DSCP 値にパケットを照合することによって、IP トラフィックを分類します。

ステップ 8	**exit** 例： `Device(config-pmap-c)# exit`	ポリシーマップ コンフィギュレーション モードに戻ります。
ステップ 9	**exit** 例： `Device(config-pmap)# exit`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/3`	ポリシーマップを適用するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。 有効なインターフェイスには、物理ポートが含まれます。
ステップ 11	**service-policy input** *policy-map-name* 例： `Device(config-if)# service-policy input policy_vlan100`	ポリシーマップ名を指定し、入力ポートに適用します。サポートされるポリシー マップは、入力ポートに 1 つだけです。
ステップ 12	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] 例： `Device# show policy-map`	（任意）入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy-running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

テーブル マップの設定

テーブルマップはマーキングの形式であり、テーブルを使用してフィールド間のマッピングと変換を可能にすることもできます。たとえば、テーブル マップはレイヤ 2 の CoS 設定をレイヤ 3 の precedence 値にマッピングして変換するために使用できます。

（注）
- テーブル マップは、複数のポリシーで、または同じポリシー内で複数回参照できます。
- デフォルトのクラスマップでカスタム出力ポリシー用に設定されたテーブルマップは、トラフィックが分類されるクラスマップに関係なく、すべてのDSCPトラフィックに影響します。回避策は、テーブルマップを削除し、デフォルトクラスで **set dscp** コマンドを設定して、分類されたトラフィックのDSCPマーキングを変更することです。ユーザー定義クラスに非キューイングアクション（ポリサーまたはマーキング）がある場合、パケットはそのユーザー定義クラス自体の値またはコメントを保持します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**table-map** *name* {**default** {*default value* | **copy** | **ignore**} | **exit** | **map** { **from** *from value* **to** *to value* } | **no**}
	例：
	`Device(config)# table-map table01`
	`Device(config-tablemap)#`
		テーブル マップを作成し、テーブル マップ コンフィギュレーション モードを開始します。テーブル マップ コンフィギュレーション モードでは、次のタスクを実行できます。
		• **default**：テーブルマップのデフォルト値を設定するか、テーブル マップ内にない値についてのデフォルトの動作（コピーまたは無視）を設定します。
		• **exit**：テーブル マップ コンフィギュレーション モードを終了します。
		• **map**：テーブルマップで *from* 値を *to* 値にマッピングします。
		• **no**：コマンドのデフォルト値を無効にするか、設定します。

ステップ 3	**map from** *value* **to** *value* 例： `Device(config-tablemap)# map from 0 to 2`	この手順では、DSCP 値が 0 のパケットを CoS 値 2 に、DSCP 値が 1 のパケットを CoS 値 4 に、DSCP 値が 24 のパケットを CoS 値 3 に、DSCP 値が 40 のパケットを CoS 値 6 に、およびそれ

	コマンドまたはアクション	目的
	Device(config-tablemap)# **map from 1 to 4** Device(config-tablemap)# **map from 24 to 3** Device(config-tablemap)# **map from 40 to 6** Device(config-tablemap)# **default 0** Device(config-tablemap)#	以外のすべてのパケットを CoS 値 0 にマークします。 （注） この例の CoS 値から DSCP 値へのマッピングは、後で説明するように、**set** ポリシー マップ クラス コンフィギュレーションコマンドを使用して設定します。
ステップ 4	**exit** 例： Device(config-tablemap)# **exit** Device(config)#	グローバル コンフィギュレーション モードに戻ります。
ステップ 5	**exit** 例： Device(config) **exit** Device#	特権 EXEC モードに戻ります。
ステップ 6	**show table-map** 例： Device# **show table-map** Table Map table01 from 0 to 2 from 1 to 4 from 24 to 3 from 40 to 6 default 0	テーブル マップ設定を表示します。
ステップ 7	**configure terminal** 例： Device# **configure terminal** Device(config)#	グローバル コンフィギュレーション モードを開始します。
ステップ 8	**policy-map** 例： Device(config)# **policy-map table-policy** Device(config-pmap)#	テーブルマップのポリシーマップを設定します。

	コマンドまたはアクション	目的

ステップ 9	**class class-default**
	例：
	`Device(config-pmap)# class class-default`
	`Device(config-pmap-c)#`
		クラスをシステムデフォルトに一致させます。

ステップ 10	**set cos dscp table** *table map name*
	例：
	`Device(config-pmap-c)# set cos dscp table table01`
	`Device(config-pmap-c)#`
		このポリシーが入力ポートに適用された場合、そのポートでは trust dscp がイネーブルになり、テーブルマップに応じてマーキングが行われます。

ステップ 11	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		特権 EXEC モードに戻ります。

次のタスク

ネットワークの QoS 用の追加のポリシーマップを設定します。ポリシーマップを作成したら、**service-policy** コマンドを使用してトラフィックポリシーをインターフェイスに付加します。

有線ターゲットの QoS に関する制約事項

ターゲットとは、ポリシーが適用されるエンティティです。有線ターゲットには、ポートまたは VLAN を指定できます。

次に、QoS 機能を有線ターゲットのデバイスに適用する場合の制限事項を示します。

- 有線ターゲットのデバイスポートでは、最大 8 つのキューイングクラスがサポートされます。
- 有線ターゲットの有線ポートでは、入力または出力方向でポリシーごとに最大 63 のポリサーがサポートされます。
- Cisco IOS XE リリース 16.x.x 以降のリリースでは、ダウンリンクポートのサイズは 10 GB ですが、デフォルトでは、すべてのダウンリンクポートに 1 GB のポートバッファが割り当てられます。この変更の前は、すべての 1 GB ダウンリンク ポートには 1 GB バッファが、10 GB ダウンリンク ポートには 10 GB バッファが割り当てられていました。
- 最大 1599 のポリシーマップを作成できます。
- QoS 階層でサポートされるのは最大 2 レベルです。

- 階層型ポリシーでは、子ポリシーの親およびキュー機能のポリシーにポートシェーパーがある場合を除き、親子間のオーバーラップは許可されていません。
- QoS ポリシーは、EtherChannel インターフェイスに付加できません。
- 親と子の両方のポリシングは、QoS 階層ではサポートされていません。
- 親と子の両方のマーキングは、QoS 階層ではサポートされていません。
- シェーピングでは、ハードウェア内部に占める 20 バイトの IPG オーバーヘッドがすべてのパケットにあります。シェーピングの精度はこれにより向上し、とくに小さいサイズのパケットに対して効果があります。
- 空のクラスはサポートされません。
- 空のアクションによるクラス マップはサポートされません。クラス マップの順序が同じポリシーが 2 つあり、どちらかのポリシーにアクションが含まれていないクラスマップがある場合、トラフィックのドロップが起こる可能性があります。回避策として、PRIORITY_QUEUE 内のすべてのクラスに最小帯域幅を割り当てます。
- 有線ターゲットの有線ポートでは、ポリシーごとに最大 256 のクラスがサポートされます。
- Cisco UADP アーキテクチャに基づき、トラフィックは QoS ルックアップの対象となり、対応する設定済みアクションに従います。このトラフィックがたとえ Egress Global Resolution ブロックに後でドロップされて、実際のインターフェイスから送信されない場合も同様です。
- ポリシー マップ内のポリサーのアクションには、次の制限事項があります。
  - 適合アクションは送信する必要があります。
  - マークダウン タイプの超過/違反アクションは、cos2cos、prec2prec、dscp2dscp だけです。
  - マークダウン タイプはポリシー内で同じである必要があります。
- SVI では、マーキングポリシーのみがサポートされます。
- ポート レベルの入力マーキング ポリシーは SVI ポリシーより優先されますが、ポート ポリシーが設定されていない場合は、SVI ポリシーが優先されます。優先するポート ポリシーに対し、ポート レベルのポリシーを定義します。SVI ポリシーが上書きされるようにするためです。
- 分類カウンタには、次の制限事項があります。
  - 分類カウンタは、バイトの代わりにパケットをカウントします。
  - フィルタ ベースの分類カウンタはサポートされません。
  - マーキングまたはポリシングによる QoS 設定だけが、分類カウンタをトリガーします。

- 分類カウンタはポートベースではありません。これは、分類カウンタが、異なるインターフェイスに接続し、同じポリシーの同じクラスに属するすべてのパケットを集約することを意味します。
- ポリシー内にポリシングまたはマーキングアクションがある限り、クラスは分類カウンタを保持します。
- 分類カウンタは、どのクラスマップ下の完全なキューイングポリシーでもサポートされません。
- クラスに複数の match ステートメントがある場合、トラフィックカウンタはクラスのすべての match ステートメントで累積されます。
- 分類カウンタ（クラスマップ）は、帯域幅、WRED、キューバッファ、シェーピングなどのアクションを含むキューイングポリシーでは使用できません。**show policy-map interface** コマンド出力には、リマーキングまたはポリサーアクションのいずれかを持つポリシーの分類カウンタ（クラスマップ）のみが表示されます。

- デバイスは、ポリサー超過マークダウンでは合計 8 つのテーブルマップ、ポリサー違反マークダウンでは 8 つのテーブルマップをサポートします。
- 階層型ポリシーは次の機能で必要になります。
  - ポート シェーパー
  - 集約ポリシング機能
  - PV ポリシー
  - 親シェーピングおよび子マーキング/ポリシング
- 親シェーピングと、プライオリティ レベル キューイングおよびプライオリティ レベル ポリシングが設定された子ポリシーを含む HQoS ポリシーでは、ポリシングの統計情報は更新されません。QoS シェイパーの統計情報のみが更新されます。QoS シェイパーの統計情報を表示するには、グローバル コンフィギュレーション モードで **show policy-map interface** コマンドを使用します。
- 有線ターゲットを含むポートでは、次の階層型ポリシーだけがサポートされています。
  - 同じポリシー内でのポリシングの連結はサポートされていません。
  - 同じポリシー内で階層型キューイングはサポートされていません（ポート シェーパーは例外）。
  - 親クラスでは、すべてのフィルタが同じタイプでなければなりません。子フィルタ タイプは次の例外を除き、親フィルタのタイプと一致している必要があります。
    - IP に一致するように親クラスが設定されている場合、ACL に一致するように子クラスを設定できます。
    - CoS に一致するように親クラスが設定されている場合、ACL に一致するように子クラスを設定できます。

- インターフェイス コンフィギュレーション モードで使用可能な **trust device** *device_type* コマンドは、デバイスでのスタンドアロンコマンドです。このコマンドを AutoQoS 設定で使用するときに、接続されているピア デバイスが対応デバイス（信頼ポリシーに一致するデバイスとして定義されているデバイス）ではない場合、CoS 値と DSCP 値の両方が「0」に設定され、いずれの入力ポリシーも有効になりません。接続されているピア デバイスが対応するデバイスである場合は、入力ポリシーが有効になります。

次に、VLAN の QoS 機能を有線ターゲットに適用する場合の制限事項を示します。

- フラットつまり非階層型ポリシーでは、マーキングまたはテーブルマップのみサポートされます。

次に、EtherChannel とチャネル メンバー インターフェイスで QoS 機能を適用するための制限事項と考慮事項を示します。

- QoS は、EtherChannel インターフェイスではサポートされません。
- QoS は、入力および出力方向の EtherChannel メンバーインターフェイスでサポートされます。すべての EtherChannel メンバーが同じ QoS ポリシーを適用する必要があります。QoS ポリシーが同じでない場合、異なるリンクの個々のポリシーは独立して機能します。
- チャネル メンバーへサービス ポリシーを付加すると、EtherChannel 内のすべてのポートに同じポリシーが接続されていることを確認するようユーザーに知らせる、次の警告メッセージが表示されます。「Warning: add service policy will cause inconsistency with port xxx in ether channel xxx. 」
- 自動 QoS は EtherChannel メンバーではサポートされません。

（注） EtherChannel へサービス ポリシーを付加すると、次のメッセージがコンソールに表示されます。「Warning: add service policy will cause inconsistency with port xxx in ether channel xxx. 」。この警告メッセージは予期されるメッセージです。この警告メッセージは、同じ EtherChannel 内の他のポートに同じポリシーを付加するように促すものです。同じメッセージがブートアップ中にも表示されます。このメッセージは、EtherChannel メンバー ポート間に不一致があることを意味するものではありません。

QoS の特性と機能の設定

帯域幅の設定

この手順は、デバイスで帯域幅を設定する方法を説明しています。

始める前に

この手順を開始する前に、帯域幅のクラス マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_bandwidth01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class_bandwidth01`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ コンフィギュレーション モードには、次のコマンドオプションが含まれます。
		• *word*：クラスマップ名。
		• **class-default**：未分類のパケットを照合するシステムデフォルトクラス。

ステップ 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio* }}
	例：
	`Device(config-pmap-c)# bandwidth 200000`
	`Device(config-pmap-c)#`
		ポリシーマップの帯域幅を設定します。パラメータは次のとおりです。
		• *Kb/s*：特定の値を kbps で設定します（100 ～ 100000000）。
		• **percent**：割合に基づいて、特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。100 % 未満の場合、帯域幅の残りは、すべての帯域幅キュー上に均等に分割されます。
		• **remaining**：特定のクラスに最小帯域幅を割り当てます。キューは、他

	コマンドまたはアクション	目的
		のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。このコマンドは、ポリシー内の特定のキューに対して **priority** コマンドが使用されている場合に使用します。各キューには、割合ではなく比率を割り当てることもできます。キューにはそれらの比率に従って、特定の重みが割り当てられます。比率は 0 ～ 100 の範囲で指定できます。この場合のポリシーの全帯域幅での比率の割り当ては、100 を超えることができます。 （注） ポリシー マップで帯域幅タイプを混在させることはできません。たとえば、1 つのポリシー マップで帯域幅の割合と kbps の両方を使用して、帯域幅を設定することはできません。

ステップ 5	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。

ステップ 6	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

ネットワークの QoS 用の追加のポリシーマップを設定します。ポリシーマップを作成したら、**service-policy** コマンドを使用して、インターフェイスにトラフィックポリシーを付加します。

ポリシングの設定

この手順は、デバイスでポリシングを設定する方法を説明しています。

始める前に

この手順を開始する前に、ポリシングのクラス マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_police01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class_police01`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ コンフィギュレーション モードには、次のコマンドオプションが含まれます。
		• *word*：クラス マップ名。
		• **class-default**：未分類のパケットを照合するシステム デフォルト クラス。

ステップ 4	**police** {*target_bit_rate* [*burst bytes* | **bc** | **conform-action** | **pir**] | **cir** {*target_bit_rate* | **percent** *percentage*} | **rate** {*target_bit_rate* | **percent** *percentage*} **conform-action transmit exceed-action** {**drop** [**violate action**] | **set-cos-transmit** | **set-dscp-transmit** | **set-prec-transmit** | **transmit** [**violate action**] }}
	例：
	`Device(config-pmap-c)# police 8000 conform-action transmit exceed-action drop`
	`Device(config-pmap-c)#`
		次の **police** サブコマンドオプションを使用できます。
		• *target_bit_rate*：ビット/秒（8000 ～ 10000000000）。
		• *burst bytes*：1000 ～ 512000000 の値を入力します。
		• **bc**：適合バースト。
		• **conform-action**：レートが適合バーストより小さくなる場合に実行されるアクション。
		• **pir**：最大情報レート。
		• **cir**：認定情報レート。

	コマンドまたはアクション	目的
		• *target_bit_rate*：ターゲット ビット レート（8000 ～ 10000000000）。 • **percent**：CIR のインターフェイス帯域幅の割合。 • **rate**：ポリシング レート、階層型ポリシーの PCR、またはシングルレベルの ATM 4.0 ポリサー ポリシーの SCR を指定します。 • *target_bit_rate*：ターゲット ビット レート（8000 ～ 10000000000）。 • **percent**：レートのインターフェイス帯域幅の割合。 次の **police conform-action transmit exceed-action** サブコマンドオプションを使用できます。 • **drop**：パケットをドロップします。 • **set-cos-transmit**：CoS 値を設定して送信します。 • **set-dscp-transmit**：DSCP 値を設定して送信します。 • **set-prec-transmit**：パケットの precedence を書き換えて送信します。 • **transmit**：パケットを送信します。 （注） ポリサー ベースのマークダウン アクションは、テーブル マップを使用する場合のみサポートされます。デバイスの各マーキングフィールドで許可されているマークダウンテーブルマップは 1 つだけです。

	コマンドまたはアクション	目的

ステップ 5	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。

ステップ 6	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。 （注） **show policy-map** コマンドの出力では、適合バイトおよび超過バイトのカウンタを表示しません。

次のタスク

ネットワークのQoS用の追加のポリシーマップを設定します。ポリシーマップを作成したら、**service-policy** コマンドを使用してトラフィックポリシーをインターフェイスに付加します。

プライオリティの設定

この手順は、デバイスでプライオリティを設定する方法を説明しています。

（注） デバイスでは、指定されたキューにプライオリティを指定できます。使用可能な2つのプライオリティ レベルがあります（1 および 2）。音声とビデオに対応するキューには、プライオリティ レベル 1 を割り当てます。

始める前に

この手順を開始する前に、プライオリティのクラス マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_priority01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class_priority01`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ コンフィギュレーション モードには、次のコマンドオプションが含まれます。
		• *word*：クラス マップ名。
		• **class-default**：未分類のパケットを照合するシステム デフォルト クラス。

ステップ 4	**priority** [*Kb/s* [*burst_in_bytes*] | **level** *level_value* [*Kb/s* [*burst_in_bytes*] | **percent** *percentage* [*burst_in_bytes*]] | **percent** *percentage* [*burst_in_bytes*]]
	例：
	`Device(config-pmap-c)# priority level 1`
	`Device(config-pmap-c)#`
		（任意）**priority** コマンドは、クラスに完全スケジューリング プライオリティを割り当てます。
		コマンドオプションは次のとおりです。
		• *Kb/s*：kbps を指定します（1 ～ 2000000）。
		• *burst_in_bytes*：バイトでバーストを指定します（32 ～ 2000000）。
		• **level** *level_value*：マルチレベル（1 ～ 2）のプライオリティ キューを指定します。
		• *Kb/s*：kbps を指定します（1 ～ 2000000）。
		• *burst_in_bytes*：バイトでバーストを指定します（32 ～ 2000000）。
		• **percent**：総帯域幅の割合。

	コマンドまたはアクション	目的
		• *burst_in_bytes*：バイトでバーストを指定します（32 ～ 2000000）。 • **percent**：総帯域幅の割合。 • *burst_in_bytes*：バイトでバーストを指定します（32 ～ 2000000）。 （注） プライオリティ レベル 1 はプライオリティ レベル 2 より重要です。プライオリティ レベル 1 は、QoS に最初に処理される帯域幅を予約するため、遅延は非常に低くなります。プライオリティ レベル 1 と 2 はどちらも帯域幅を予約します。

ステップ 5	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。

ステップ 6	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

ネットワークのQoS用の追加のポリシーマップを設定します。ポリシーマップを作成したら、**service-policy** コマンドを使用してトラフィックポリシーをインターフェイスに付加します。

SGT ベースの QoS の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**class-map** *class-map-name* {**match-any** | **match-all** } 例： `Device(config)# class-map c1`	クラスマップを指定し、クラスマップ コンフィギュレーション モードを開始します。
ステップ 3	**match security-group source tag** *sgt-number* 例： `Device(config-cmap)# match security-group source tag 1000`	security-group source security tag の値を設定します。
ステップ 4	**match security-group destination tag** *dgt-number* 例： `Device(config-cmap)# match security-group destination tag 2000`	security-group destination security tag の値を設定します。

ステップ 5	**exit**
	例：
	`Device(config-cmap)# exit`
	`Device#`
		ルート マップ インターフェイス コンフィギュレーション モードを終了して、グローバル コンフィギュレーション モードに戻ります。

ステップ 6	**policy-map** *policy-map-name*
	例：
	`Device(config)# policy-map pin`
	`Device(config-pmap)#`
		ポリシー マップを指定し、ポリシー マップ コンフィギュレーション モードを開始します。
		policy-map-name は子ポリシーマップの名前です。名前には最大 40 文字までの英数字を指定できます。

ステップ 7	**class** *class-name* 例： `Device(config-pmap)# class c1`	ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラスマップ コンフィギュレーション モードに

	コマンドまたはアクション	目的
	`Device(config-pmap-c)#`	は、次のコマンドオプションが含まれます。 • *word*：クラス マップ名。 • **class-default**：未分類のパケットを照合するシステムデフォルトクラス。
ステップ 8	**set dscp** *dscp-value* 例： `Device(config-pmap-c)# set dscp af11`	DiffServ コードポイント（DSCP）値を設定します。

ステップ 9	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。クラスマップコンフィギュレーションモードを終了し、グローバルコンフィギュレーション モードを開始します。

ステップ 10	**interface** *interface-num* 例： `Device(config)# interface GigabitEthernet1/0/24`	インターフェイスを指定し、インターフェイス コンフィギュレーションモードを開始します。
ステップ 11	**service-policy** { **input** | **output** } *policy-map-name* 例： `Device(config-if)#service-policy input pin`	インターフェイスの入力にポリシーマップを割り当てます。

ステップ 12	**end**
	例：
	`Device(config-if)# end`
	`Device#`
		設定変更を保存します。インターフェイス コンフィギュレーションモードを終了し、グローバルコンフィギュレーション モードに入ります。

SGT ベースの QoS 分類の設定例

次に、インターフェイスでの SGT ベースの QoS の設定例を示します。

```
ip access-list role-based sgt_acl
 10 permit ip
cts role-based sgt-map 24.0.0.0/8 sgt 24
cts role-based enforcement
cts role-based permissions from 24 to 24 sgt_acl
```

```
class-map match-all c1
 match protocol attribute business-relevance business-relevant
 match protocol attribute traffic-class ops-admin-mgmt
 match security-group destination tag 24
 match security-group source tag 24

policy-map pin
 class c1
  set dscp af11
 class class-default
  set dscp af12

interface GigabitEthernet1/0/24
 no switchport
 ip address 24.1.1.2 255.255.255.0
 service-policy input pin
 ip nbar protocol-discovery
```

キューとシェーピングの設定

出力キューの特性の設定

ネットワークおよびQoSソリューションの複雑さによっては、この項の手順をすべて実行する必要があります。次の特性を決定する必要があります。

- DSCP、CoS、またはQoSグループ値によって各キューおよびしきい値IDにマッピングされるパケット
- キューに適用されるドロップ割合のしきい値と、トラフィック タイプで必要な予約メモリと最大メモリ
- キューに割り当てる固定バッファ スペース
- ポートの帯域幅に関するレート制限の必要性
- 出力キューの処理頻度、および使用する技術（シェーピング、共有、または両方）

（注） 出力キューはデバイスでのみ設定できます。

キュー バッファの設定

デバイスでは、キューにバッファを割り当てることができます。バッファが割り当てられていない場合は、すべてのキューに対して均等に分割されます。queue-buffer ratioを使用して、特定の比率で分割できます。デフォルトでDTS（Dynamic Threshold and Scaling）はすべてのキューでアクティブになるため、これらはソフト バッファになります。

（注） queue-buffer ratio は queue-limit とともに設定することはできません。

始める前に

この手順の前提条件を次に示します。

- この手順を開始する前に、キュー バッファのクラス マップを作成する必要があります。
- キュー バッファを設定する前に、ポリシー マップの帯域幅、シェーピング、またはプライオリティを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_queuebuffer01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class_queuebuffer01`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ コンフィギュレーション モードには、次のコマンドオプションが含まれます。
		• *word*：クラス マップ名。
		• **class-default**：未分類のパケットを照合するシステム デフォルト クラス。

ステップ 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value* }} 例：	ポリシーマップの帯域幅を設定します。コマンドパラメータは次のとおりです。

	コマンドまたはアクション	目的
	`Device(config-pmap-c)# bandwidth percent 80` `Device(config-pmap-c)#`	• *Kb/s*：特定の値を設定するには、このコマンドを使用します。指定できる範囲は 20000 ～ 100000000 です。 • **percent**：割合を使用して特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。100 % 未満の場合、帯域幅の残りは、すべての帯域幅キュー上に均等に分割されます。 • **remaining**：特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。このコマンドは、ポリシー内の特定のキューに対して **priority** コマンドが使用されている場合に使用します。各キューには、割合ではなく比率を割り当てることもできます。キューにはそれらの比率に従って、特定の重みが割り当てられます。比率は 0 ～ 100 の範囲で指定できます。この場合のポリシーの全帯域幅での比率の割り当ては、100 を超えることができます。 （注） ポリシー マップで帯域幅タイプを混在させることはできません。
ステップ 5	**queue-buffers** { **ratio** *ratio value*} 例：	キューの相対的なバッファ サイズを設定します。

	コマンドまたはアクション	目的
	Device(config-pmap-c)# **queue-buffers ratio 10** Device(config-pmap-c)#	（注） ポリシーに設定されているすべてのバッファの合計が 100 % 以下である必要があります。未割り当てバッファは、残りのキューに均等に分散されます。プライオリティ キューを含むすべてのキューに十分なバッファが割り当てられるようにします。 （注） スパニングツリーや LACP などのネットワーク制御プロトコルのプロトコル データ ユニット（PDU）は、プライオリティ キューまたはキュー 0（プライオリティ キューが設定されていない場合）を使用します。プロトコルが機能するには、これらのキューに十分なバッファが割り当てられるようにします。
ステップ 6	**end** 例： Device(config-pmap-c)# **end** Device#	設定変更を保存します。
ステップ 7	**show policy-map** 例： Device# **show policy-map**	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

ネットワークのQoS用の追加のポリシーマップを設定します。ポリシーマップを作成したら、**service-policy** コマンドを使用してトラフィックポリシーをインターフェイスに付加します。

キュー制限の設定

重み付けテールドロップ（WTD）を設定するためにキュー制限を使用します。WTDを使用すると、キューごとに複数のしきい値を設定できます。各サービスクラスが異なるしきい値でドロップされてQoS差別化が実現されます。デバイスによって、3つの明示的にプログラム可能なしきい値クラスとして各キューに0、1、2を指定できます。したがって、キューごとに各パケットのキューイング/ドロップの決定は、フレーム ヘッダーのDSCP、CoS、またはQoS グループ フィールドに指定されたパケットのしきい値クラスの割り当てによって決定されます。

WTD では柔軟な制限が使用されるため、最大 400 %（共通プールで予約されるバッファの最大4倍）のキュー制限を設定できます。この柔軟な制限は、他の機能に影響することなく、共通プールのオーバーランを防止します。

（注） キュー制限は、有線ポートのデバイスの出力キューでのみ設定できます。

始める前に

この手順の前提条件を次に示します。

- この手順を開始する前に、キュー制限を使用するクラス マップを作成する必要があります。
- キュー制限を設定する前に、ポリシーマップの帯域幅、シェーピング、またはプライオリティを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_queuelimit01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。

ステップ3	**class** *class name* 例： `Device(config-pmap)# class`	ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ

	コマンドまたはアクション	目的
	class_queuelimit01 Device(config-pmap-c)#	コンフィギュレーション モードには、次のコマンドオプションが含まれます。 • *word*：クラス マップ名。 • **class-default**：未分類のパケットを照合するシステム デフォルト クラス。
ステップ 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value* }} 例： Device(config-pmap-c)# **bandwidth 500000** Device(config-pmap-c)#	ポリシーマップの帯域幅を設定します。パラメータは次のとおりです。 • *Kb/s*：特定の値を設定するには、このコマンドを使用します。指定できる範囲は 20000 ～ 100000000 です。 • **percent**：特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。100 % 未満の場合、帯域幅の残りは、すべての帯域幅キュー上に均等に分割されます。 • **remaining**：特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。このコマンドは、ポリシー内の特定のキューに対して **priority** コマンドが使用されている場合に使用します。各キューには、割合ではなく比率を割り当てることもできます。キューにはそれらの比率に従って、特定の重みが割り当てられます。比率は 0 ～ 100 の範囲で指定できます。この場合のポリシーの全帯域幅での比率の割り当ては、100 を超えることができます。

	コマンドまたはアクション	目的
		（注） ポリシー マップで帯域幅タイプを混在させることはできません。

ステップ 5	**queue-limit** {*packets* **packets** | **cos** {*cos value* { *maximum threshold value* | **percent** *percentage* } | **values** {*cos value* | **percent** *percentage* } } | **dscp** {*dscp value* {*maximum threshold value* | **percent** *percentage*} | *match packet* {*maximum threshold value* | **percent** *percentage*} | **default** {*maximum threshold value* | **percent** *percentage*} | **ef** {*maximum threshold value* | **percent** *percentage*} | **dscp values** *dscp value*} | **percent** *percentage* }}
	例：
	`Device(config-pmap-c)# queue-limit dscp 3 percent 20`
	`Device(config-pmap-c)# queue-limit dscp 4 percent 30`
	`Device(config-pmap-c)# queue-limit dscp 5 percent 40`
		キュー制限のしきい値の割合を設定します。
		すべてのキューで、3つのしきい値（0、1、2）があり、それぞれのしきい値についてデフォルト値があります。デフォルトまたはその他のキュー制限しきい値設定を変更するには、このコマンドを使用します。たとえば、DSCP 3、4、および5のパケットが設定した特定のキューに送信される場合、このコマンドは、この3つの DSCP 値のしきい値パーセンテージを設定できます。キュー制限しきい値に関する詳細については、重み付けテール ドロップ （64 ページ）を参照してください。
		（注） デバイスは絶対キュー制限の割合をサポートしません。デバイスは、DSCP または CoS キュー制限の割合だけをサポートします。

ステップ 6	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。

ステップ 7	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

ネットワークの QoS 用の追加ポリシー マップを設定します。ポリシーマップを作成したら、**service-policy** コマンドを使用して、トラフィックポリシーをインターフェイスに付加します。

シェーピングの設定

特定のクラスのシェーピング（最大帯域幅）を設定するには、**shape** コマンドを使用します。ポートに残っている追加帯域幅があっても、キューの帯域幅はこの値に制限されます。シェーピングは平均の割合で、または bps のシェーピングの平均値で設定できます。

始める前に

この手順を開始する前に、シェーピングのクラス マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_shaping01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class_shaping01`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ コンフィギュレーション モードには、次のコマンドオプションが含まれます。
		• *word*：クラス マップ名。
		• **class-default**：未分類のパケットを照合するシステム デフォルト クラス。

ステップ 4	**shape average** {*target bit rate* | **percent** *percentage*}
	例：
	`Device(config-pmap-c)# shape average percent 50`
	`Device(config-pmap-c)#`
		平均シェーピング レートを設定します。平均シェーピング レートを、ターゲット ビット レート（bps）または認定情報レート（CIR）のインターフェイス帯域幅の割合で設定できます。

	コマンドまたはアクション	目的

ステップ 5	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。

ステップ 6	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

ネットワークの QoS 用の追加のポリシー マップを設定します。ポリシー マップを作成したら、**service-policy** コマンドを使用してトラフィック ポリシーをインターフェイスに付加します。

シャープ プロファイル キューイングの設定

この手順は、スイッチでシャープ プロファイル キューイングを設定する方法を説明しています。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**policy-map** *policy name*
	例：
	`Device(config)# policy-map policy_shaping01`
	`Device(config-pmap)#`
		ポリシー マップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシー マップを作成または修正し、サービス ポリシーを指定します。
		policy-map-name は子ポリシーマップの名前です。名前には最大 40 文字までの英数字を指定できます。

	コマンドまたはアクション	目的

ステップ 3	**class** *class name*
	例：
	`Device(config-pmap)# class class_shaping01`
	`Device(config-pmap-c)#`
		ポリシー クラス マップ コンフィギュレーション モードを開始します。ポリシーを作成または変更するクラスの名前を指定します。ポリシー クラス マップ コンフィギュレーション モードには、次のコマンドオプションが含まれます。
		• *word*：クラス マップ名。
		• **class-default**：未分類のパケットを照合するシステム デフォルト クラス。

ステップ 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value*}}
	例：
	`Device(config-pmap-c)# bandwidth 200000`
	`Device(config-pmap-c)#`
		ポリシーマップの帯域幅を設定します。パラメータは次のとおりです。
		• *Kb/s*：特定の値を kbps で設定します（100 ～ 100000000）。
		• **percent-**：割合に基づいて、特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。100 % 未満の場合、帯域幅の残りは、すべての帯域幅キュー上に均等に分割されます。
		• **remaining**：特定のクラスに最小帯域幅を割り当てます。キューは、他のキューが全体のポート帯域幅を使用しない場合は、帯域幅をオーバーサブスクライブすることができます。合計が 100 % を超えることはできません。このコマンドは、ポリシー内の特定のキューに対して **priority** コマンドが使用されている場合に使用します。各キューには、割合ではなく比率を割り当てることもできます。キューにはそれらの比率に従って、特定の重みが割り当てられます。比率の範囲は 1 ー 65536 です。この場合のポリシーの全帯域

	コマンドまたはアクション	目的
		幅での比率の割り当ては、100 を超えることができます。 （注） ポリシー マップで帯域幅タイプを混在させることはできません。

ステップ 5	**shape average** {*target bit rate* | **percent** *percentage*}
	例：
	`Device(config-pmap-c)# shape average percent 50`
	`Device(config-pmap-c)#`
		平均シェーピング レートを設定します。平均シェーピング レートを、ターゲット ビット レート（bps）または認定情報レート（CIR）のインターフェイス帯域幅の割合で設定できます。

ステップ 6	**end**
	例：
	`Device(config-pmap-c)# end`
	`Device#`
		設定変更を保存します。

シャープ プロファイル キューイングの設定

次に、シャープキューイングの例を示します。

```
Policy Map test
    Class test1
      bandwidth 20 (%)
      Average Rate Traffic Shaping
      cir 40%
    Class test3
      Average Rate Traffic Shaping
      cir 50%
    Class test2
      Average Rate Traffic Shaping
      cir 50%
    Class test4
      bandwidth 20 (%)
    Class test5
      Average Rate Traffic Shaping
      cir 70%
    Class test6
      Average Rate Traffic Shaping
      cir 60%
```

QoS のモニタリング

デバイスでの QoS のモニタリングには、次のコマンドを使用できます。

表 *11 : QoS* のモニタリング

コマンド	説明
show class-map [*class_map_name*]	設定されているすべてのクラスマップのリストを表示します。
show policy-map [*policy_map_name*]	設定されているすべてのポリシーマップのリストを表示します。コマンドパラメータは次のとおりです。 • policy map name • interface • session
show policy-map session [**input** | **output** | **uid** *UUID*]	セッションの QoS ポリシーを表示します。コマンドパラメータは次のとおりです。 • input：入力ポリシー • output：出力ポリシー • uid：SSS 固有の ID に基づくポリシー
show table-map	すべてのテーブルマップと設定を表示します。

QoS の設定例

例：TCP プロトコル分類

TCP パケットは、ポート番号に基づいて分類できます。TCP プロトコルの設定は次のとおりです。

```
Device#show ip acce tcp
Extended IP access list tcp
    10 permit tcp any any eq 80
Device #
```

```
Device #show run class-map tcp

Current configuration : 63 bytes
!
class-map match-all tcp
 match access-group name tcp
!
end
Device #
Device #show run policy-map tcp

Current configuration : 56 bytes
!
policy-map tcp
 class tcp
  police 1000000000
!
end
Device #

Device #show run int tw 1/0/1

Current configuration : 93 bytes
!
interface TwentyFiveGigE1/0/1
 no ip address
 no keepalive
 service-policy output tcp
end

Device #
```

例：UDP プロトコル分類

UDP パケットは、ポート番号に基づいて分類できます。UDP プロトコルの設定例は次のとおりです。

```
Device#show ip acce udp
Extended IP access list udp
    10 permit udp any any eq ntp
Device #

Device #show run class-map udp
Building configuration...

Current configuration : 63 bytes
!
class-map match-all udp
 match access-group name udp
!
end

Device #
Device #show run policy-map udp
Building configuration...

Current configuration : 56 bytes
!
policy-map udp
 class udp
  police 1000000000
!
```

```
end
Device #
Device #show run int tw 1/0/1

Current configuration : 93 bytes
!
interface TwentyFiveGigE1/0/1
 no ip address
 no keepalive
 service-policy output udp
end

Device #
```

例：RTP プロトコル分類

RTPパケットは、ポート番号に基づいて分類できます。RTPプロトコルの設定例は次のとおりです。

```
Device# show ip access-list rtp
Extended IP access list rtp
    10 permit udp any any eq 554
    11 permit tcp any any eq 554
Device #

Device #show run class-map rtp

Current configuration : 63 bytes
!
class-map match-all rtp
 match access-group name rtp
!
end

Device #
Device #show run policy-map rtp

Current configuration : 56 bytes
!
policy-map rtp
 class rtp
  police 1000000000
!
end

Device #
Device #show run int tw 1/0/1

Current configuration : 93 bytes
!
interface TwentyFiveGigE1/0/1
 no ip address
 no keepalive
 service-policy output rtp
end

Device #
```

例：アクセス コントロール リストによる分類

この例は、アクセス コントロール リスト（ACL）を使用して QoS のパケットを分類する方法を示しています。

```
Device# configure terminal
Device(config)# access-list 101 permit ip host 12.4.1.1 host 15.2.1.1
Device(config)# class-map acl-101
Device(config-cmap)# description match on access-list 101
Device(config-cmap)# match access-group 101
Device(config-cmap)#
```

ACL を使用してクラスマップを作成した後で、クラスのポリシー マップを作成し、ポリシー マップを QoS のインターフェイスに適用します。

例：サービス クラス レイヤ 2 の分類

この例は、サービス クラス レイヤ 2 の分類を使用して QoS に対してパケットを分類する方法を示しています。

```
Device# configure terminal
Device(config)# class-map cos
Device(config-cmap)# match cos ?
  <0-7>  Enter up to 4 class-of-service values separated by white-spaces
Device(config-cmap)# match cos 3 4 5
Device(config-cmap)#
```

CoS レイヤ 2 の分類を使用してクラス マップを作成したら、そのクラスのポリシー マップを作成し、QoS のインターフェイスにポリシー マップを適用します。

例：サービス クラス DSCP の分類

この例は、サービス クラス DSCP の分類を使用して、QoS に対してパケットを分類する方法を示しています。

```
Device# configure terminal
Device(config)# class-map dscp
Device(config-cmap)# match dscp af21 af22 af23
Device(config-cmap)#
```

DSCP 分類を使用してクラス マップを作成したら、クラスのポリシー マップを作成し、QoS のインターフェイスにポリシー マップを適用します。

例：VLAN ID レイヤ 2 の分類

この例は、VLAN ID レイヤ 2 の分類を使用して QoS に分類する方法を示しています。

```
Device# configure terminal
```

```
Device(config)# class-map vlan-120
Device(config-cmap)# match vlan ?
  <1-4095>  VLAN id
Device(config-cmap)# match vlan 120
Device(config-cmap)#
```

VLAN レイヤ 2 の分類を使用してクラス マップを作成したら、クラスのポリシー マップを作成し、QoS のインターフェイスにポリシー マップを適用します。

例：DSCP 値または precedence 値による分類

この例は、DSCP 値または precedence 値を使用してパケットを分類する方法を示しています。

```
Device# configure terminal
Device(config)# class-map prec2
Device(config-cmap)# description matching precedence 2 packets
Device(config-cmap)# match ip precedence 2
Device(config-cmap)# exit
Device(config)# class-map ef
Device(config-cmap)# description EF traffic
Device(config-cmap)# match ip dscp ef
Device(config-cmap)#
```

DSCP 値または precedence 値を使用してクラス マップを作成したら、クラスのポリシー マップを作成し、QoS のインターフェイスにポリシー マップを適用します。

例：階層型分類

次の例は、child という名前の別のクラスに一致する parent という名前のクラスが作成される、階層型分類を示しています。child という名前のクラスは、2 に設定された IP precedence に基づいて照合されます。

```
Device# configure terminal
Device(config)# class-map child
Device(config-cmap)# match ip precedence 2
Device(config-cmap)# exit
Device(config)# class-map parent
Device(config-cmap)# match class child
Device(config-cmap)#
```

親クラス マップを作成したら、クラスのポリシー マップを作成し、QoS のインターフェイスにポリシー マップを適用します。

例：階層型ポリシーの設定

次の例は、階層型ポリシーを使用した設定を示しています。

```
Device# configure terminal
Device(config)# class-map c1
Device(config-cmap)# match dscp 30
```

```
Device(config-cmap)# exit

Device(config)# class-map c2
Device(config-cmap)# match precedence 4
Device(config-cmap)# exit

Device(config)# class-map c3
Device(config-cmap)# exit

Device(config)# policy-map child
Device(config-pmap)# class c1
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# police rate percent 20 conform-action transmit exceed action drop

Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit

Device(config-pmap)# class c2
Device(config-pmap-c)# bandwidth 20000
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# bandwidth 20000
Device(config-pmap-c)# exit
Device(config-pmap)# exit

Device(config)# policy-map parent
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 1000000
Device(config-pmap-c)# service-policy child
Device(config-pmap-c)# end
```

次の例は、テーブル マップを使用した階層型ポリシーを示しています。

```
Device(config)# table-map dscp2dscp
Device(config-tablemap)# default copy
Device(config)# policy-map ssid_child_policy
Device(config-pmap)# class voice
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# police 15000000
Device(config-pmap)# class video
Device(config-pmap-c)# priority level 2
Device(config-pmap-c)# police 10000000
Device(config)# policy-map ssid_policy
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 30000000
Device(config-pmap-c)# queue-buffer ratio 0
Device(config-pmap-c)# set dscp dscp table dscp2dscp
Device(config-pmap-c)# service-policy ssid_child_policy
```

例：音声およびビデオの分類

この例は、デバイス固有の情報を使用して、音声とビデオのパケットストリームを分類する方法を示しています。

この例では、音声とビデオがエンドポイント A からデバイスの GigabitEthernet1/0/1 に送信され、それぞれ precedence 値 5 と 6 を持ちます。また、音声とビデオは、エンドポイント B からデバイスの FortyGigabitEthernet1/0/2 にそれぞれ DSCP 値 EF と AF11 で送信されます。

両方のインターフェイスからのすべてのパケットがアップリンクインターフェイスに送信されます。その場合、音声は 100 Mbps にポリシングし、ビデオは 150 Mbps にポリシングする必要があります。

上記の要件ごとに分類するために、GigabitEthernet1/0/1 で送信される音声パケットに一致するクラスが作成されます。これには、precedence 5 に一致する voice-interface-1 という名前が付けられます。同様に、GigabitEthernet1/0/2 の音声パケットに一致する、voice-interface-2 という名前の音声用の別のクラスが作成されます。これらのクラスは、GigabitEthernet1/0/1 に接続される input-interface-1 と、GigabitEthernet1/0/2 に接続される input-interface-2 という 2 つの別個のポリシーに関連付けられます。このクラスのアクションは、qos-group に 10 とマーキングすることです。出力インターフェイスで QoS-group 10 のパケットを照合するために、QoS-group 10 で一致する voice という名前のクラスが作成されます。これは、output-interface という名前の別のポリシーに関連付けられ、アップリンクインターフェイスに関連付けられます。ビデオも同じ方法で処理されますが、QoS-group 20 で一致します。

次の例は、上記のデバイス固有の情報を使用して分類する方法を示しています。

```
Device(config)#
Device(config)# class-map voice-interface-1
Device(config-cmap)# match ip precedence 5
Device(config-cmap)# exit

Device(config)# class-map video-interface-1
Device(config-cmap)# match ip precedence 6
Device(config-cmap)# exit

Device(config)# class-map voice-interface-2
Device(config-cmap)# match ip dscp ef
Device(config-cmap)# exit

Device(config)# class-map video-interface-2
Device(config-cmap)# match ip dscp af11
Device(config-cmap)# exit

Device(config)# policy-map input-interface-1
Device(config-pmap)# class voice-interface-1
Device(config-pmap-c)# set qos-group 10
Device(config-pmap-c)# exit

Device(config-pmap)# class video-interface-1
Device(config-pmap-c)# set qos-group 20

Device(config-pmap-c)# policy-map input-interface-2
Device(config-pmap)# class voice-interface-2
Device(config-pmap-c)# set qos-group 10
Device(config-pmap-c)# class video-interface-2
Device(config-pmap-c)# set qos-group 20
Device(config-pmap-c)# exit
Device(config-pmap)# exit

Device(config)# class-map voice
Device(config-cmap)# match qos-group 10
Device(config-cmap)# exit

Device(config)# class-map video
Device(config-cmap)# match qos-group 20
Device(config)# policy-map output-interface
```

```
Device(config-pmap)# class voice
Device(config-pmap-c)# police 256000 conform-action transmit exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit

Device(config-pmap)# class video
Device(config-pmap-c)# police 1024000 conform-action transmit exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit
```

例：平均レート シェーピングの設定

次の例は、平均レート シェーピングを設定する方法を示しています。

```
Device# configure terminal
Device(config)# class-map prec1
Device(config-cmap)# description matching precedence 1 packets
Device(config-cmap)# match ip precedence 1
Device(config-cmap)# end

Device# configure terminal
Device(config)# class-map prec2
Device(config-cmap)# description matching precedence 2 packets
Device(config-cmap)# match ip precedence 2
Device(config-cmap)# exit

Device(config)# policy-map shaper
Device(config-pmap)# class prec1
Device(config-pmap-c)# shape average 512000
Device(config-pmap-c)# exit

Device(config-pmap)# policy-map shaper
Device(config-pmap)# class prec2
Device(config-pmap-c)# shape average 512000
Device(config-pmap-c)# exit

Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 1024000
```

クラス マップ、ポリシー マップ、シェーピング平均を設定したら、QoS のインターフェイスにポリシー マップを適用します。

例：キュー制限の設定

次の例は、DSCP 値および割合に基づいて、キュー制限ポリシーを設定する方法を示しています。

```
Device# configure terminal
Device#(config)# policy-map port-queue
Device#(config-pmap)# class dscp-1-2-3
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 1 percent 80
Device#(config-pmap-c)# queue-limit dscp 2 percent 90
Device#(config-pmap-c)# queue-limit dscp 3 percent 100
Device#(config-pmap-c)# exit
```

```
Device#(config-pmap)# class dscp-4-5-6
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 4 percent 20
Device#(config-pmap-c)# queue-limit dscp 5 percent 30
Device#(config-pmap-c)# queue-limit dscp 6 percent 20
Device#(config-pmap-c)# exit

Device#(config-pmap)# class dscp-7-8-9
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 7 percent 20
Device#(config-pmap-c)# queue-limit dscp 8 percent 30
Device#(config-pmap-c)# queue-limit dscp 9 percent 20
Device#(config-pmap-c)# exit

Device#(config-pmap)# class dscp-10-11-12
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 10 percent 20
Device#(config-pmap-c)# queue-limit dscp 11 percent 30
Device#(config-pmap-c)# queue-limit dscp 12 percent 20
Device#(config-pmap-c)# exit

Device#(config-pmap)# class dscp-13-14-15
Device#(config-pmap-c)# bandwidth percent 10
Device#(config-pmap-c)# queue-limit dscp 13 percent 20
Device#(config-pmap-c)# queue-limit dscp 14 percent 30
Device#(config-pmap-c)# queue-limit dscp 15 percent 20
Device#(config-pmap-c)# end
Device#
```

上記のポリシーマップのキュー制限の設定が終了すると、QoS のインターフェイスにポリシーマップを適用することができます。

例：キュー バッファの設定

次の例は、キュー バッファ ポリシーを設定して QoS のインターフェイスに適用する方法を示しています。

```
Device# configure terminal
Device(config)# policy-map policy1001
Device(config-pmap)# class class1001
Device(config-pmap-c)# bandwidth remaining ratio 10
Device(config-pmap-c)# queue-buffer ratio ?
  <0-100>  Queue-buffers ratio limit
Device(config-pmap-c)# queue-buffer ratio 20
Device(config-pmap-c)# end

Device# configure terminal
Device(config)# interface HundredGigabitE1/0/3
Device(config-if)# service-policy output policy1001
Device(config-if)# end
```

例：ポリシング アクションの設定

次の例は、ポリサーに関連付けることができるさまざまなポリシングアクションを示しています。これらのアクションは、パケット設定の適合、超過、または違反によって実現されます。トラフィック プロファイルを超過または違反したパケットをドロップ、マーク付け、または送信することができます。

たとえば、1つの一般的な導入シナリオでは、エンタープライズ顧客ポリシートラフィックがネットワークからサービスプロバイダに送信され、DSCP値が異なる、適合、超過、および違反パケットをマーキングします。サービスプロバイダは、輻輳があるとDSCP値の超過および違反としてマーキングされたパケットをドロップすることができますが、使用可能な帯域幅がある場合は送信することも可能です。

（注） Layer 2 フィールドには CoS フィールドが含まれるようにマーキングでき、Layer 3 フィールドには precedence および DSCP フィールドが含まれるようにマーキングできます。

1 つの便利な機能として、複数のアクションとイベントを関連付ける機能があります。たとえば、すべての適合パケットについて、precedence ビットと CoS を設定できます。アクションを設定するサブモードは、ポリシング機能によって配信できます。

これは、ポリシング アクションの設定例を示しています。

```
Device# configure terminal
Device(config)# policy-map police
Device(config-pmap)# class class-default
Device(config-pmap-c)# police cir 1000000 pir 2000000
Device(config-pmap-c-police)# conform-action transmit
Device(config-pmap-c-police)# exceed-action set-dscp-transmit dscp table
exceed-markdown-table
Device(config-pmap-c-police)# violate-action set-dscp-transmit dscp table
violate-markdown-table
Device(config-pmap-c-police)# end
```

この例では、exceed-markdown-table と violate-mark-down-table がテーブル マップです。

（注） ポリサー ベースのマークダウン アクションは、テーブル マップを使用する場合のみサポートされます。デバイスの各マーキングフィールドで許可されているマークダウンテーブルマップは 1 つだけです。

例：ポリサーの VLAN 設定

次の例では、VLANのポリサー設定を表示します。この設定の最後に、QoSのインターフェイスに VLAN ポリシー マップを適用します。

```
Device# configure terminal
```

```
Device(config)# class-map vlan100
Device(config-cmap)# match vlan 100
Device(config-cmap)# exit
Device(config)# policy-map vlan100
Device(config-pmap)# policy-map class vlan100
Device(config-pmap-c)# police 100000 bc conform-action transmit exceed-action drop
Device(config-pmap-c-police)# end
Device# configure terminal
Device(config)# interface HundredGigabitE1/0/5
Device(config-if)#  service-policy input vlan100
```

例：ポリシングの単位

ポリシングの単位は、トークンバケットが機能する基礎となります。CIR および PIR はビット/秒で指定します。バーストパラメータはバイト単位で指定します。これはデフォルトのモードであり、単位が指定されていない場合に使用される単位です。CIR および PIR は、パーセントでも設定できます。その場合バースト パラメータをミリ秒単位で設定する必要があります。

次の例は、ビット/秒のポリサー設定を示しています。この設定では、測定単位がビットであるデュアルレートの 3 カラーポリサーが設定されます。バーストおよびピークバーストはすべてビットに指定されます。

```
Device(config)# policy-map bps-policer
Device(config-pmap)# class class-default
Device(config-pmap-c)# police rate 100000 peak-rate 1000000
conform-action transmit exceed-action set-dscp-transmit dscp table
DSCP_EXCE violate-action drop
```

例：シングルレート 2 カラー ポリシング設定

次の例は、シングルレート 2 カラー ポリサーを設定する方法を示しています。

```
Device(config)# class-map match-any prec1
Device(config-cmap)# match ip precedence 1
Device(config-cmap)# exit
Device(config)# policy-map policer
Device(config-pmap)# class prec1
Device(config-pmap-c)# police cir 256000 conform-action transmit exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)#
```

例：デュアルレート 3 カラー ポリシング設定

次の例は、デュアルレート 3 カラー ポリサーを設定する方法を示しています。

```
Device# configure terminal
Device(config)# policy-Map dual-rate-3color-policer
Device(config-pmap)# class class-default
Device(config-pmap-c)# police cir 64000 bc 2000 pir 128000 be 2000
Device(config-pmap-c-police)# conform-action transmit
```

```
Device(config-pmap-c-police)# exceed-action set-dscp-transmit dscp table
exceed-markdown-table
Device(config-pmap-c-police)# violate-action set-dscp-transmit dscp table
violate-markdown-table
Device(config-pmap-c-police)# exit
Device(config-pmap-c)#
```

この例では、exceed-markdown-table と violate-mark-down-table がテーブル マップです。

（注） ポリサー ベースのマークダウン アクションは、テーブル マップを使用する場合のみサポートされます。デバイスの各マーキングフィールドで許可されているマークダウンテーブルマップは 1 つだけです。

例：テーブル マップのマーキング設定

次のステップと例は、QoS設定でテーブルマップマーキングを使用する方法を示しています。

1. テーブル マップを定義します。

   table-map コマンドを使用してテーブルマップを定義し、値のマッピングを示します。このテーブルでは、テーブルが使用されるポリシーまたはクラスを認識しません。テーブルマップのデフォルトのコマンドは、一致する「from」フィールドがない場合に、「to」フィールドにコピーされる値を示します。この例では、table-map1 というテーブル マップが作成されます。定義されたマッピングでは、値 0 が 1 に、2 が 3 に変換され、デフォルト値は 4 に設定されます。

   ```
   Device(config)# table-map table-map1
   Device(config-tablemap)# map from 0 to 1
   Device(config-tablemap)# map from 2 to 3
   Device(config-tablemap)# default 4
   Device(config-tablemap)# exit
   ```

2. テーブル マップが使用されるポリシー マップを定義します。

   この例では、着信 CoS が table-map1 テーブルで指定されたマッピングに基づいて、DSCP にマッピングされます。この例では、着信パケットの DSCP が 0 である場合、パケット内の CoS は 1 に設定されます。テーブル マップ名が指定されていない場合、このコマンドではデフォルトの動作が実行され、値が「from」フィールド（この場合は DSCP）から「to」フィールド（この場合は CoS）にコピーされます。ただし、CoS が 3 ビット フィールドであっても DSCP は 6 ビット フィールドです。これは、DSCP 内の最初の 3 ビットに CoS がコピーされることを意味します。

   ```
   Device(config)# policy map policy1
   Device(config-pmap)# class class-default
   Device(config-pmap-c)# set cos dscp table table-map1
   Device(config-pmap-c)# exit
   ```

3. ポリシーをインターフェイスに関連付けます。

```
Device(config)# interface HundredGigabitE1/0/2
Device(config-if)# service-policy output policy1
Device(config-if)# exit
```

例：CoS マーキングを保持するテーブル マップの設定

次の例は、テーブル マップを使用して、QoS 設定のインターフェイスで CoS マーキングを保持する方法を示しています。

（例で設定されている）cos-trust-policy ポリシーは入力方向でイネーブルになり、インターフェイスに着信する CoS マーキングが保持されます。ポリシーがイネーブルになっていない場合は、デフォルトで DSCP だけが信頼されます。純粋なレイヤ 2 パケットがインターフェイスに着信すると、CoS の入力ポートに一致するポリシーがない場合は、CoS 値が 0 に書き換えられます。

```
Device# configure terminal
Device(config)# table-map cos2cos
Device(config-tablemap)# default copy
Device(config-tablemap)# exit


Device(config)# policy map cos-trust-policy
Device(config-pmap)# class class-default
Device(config-pmap-c)# set cos cos table cos2cos
Device(config-pmap-c)# exit

Device(config)# interface HundredGigabitE1/0/2
Device(config-if)# service-policy input cos-trust-policy
Device(config-if)# exit
```

次の作業

QoS 設定でこれらの自動機能を使用できるかどうかについては、自動 QoS のマニュアルを参照してください。

QoS に関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Command Reference* 『Cisco IOS Quality

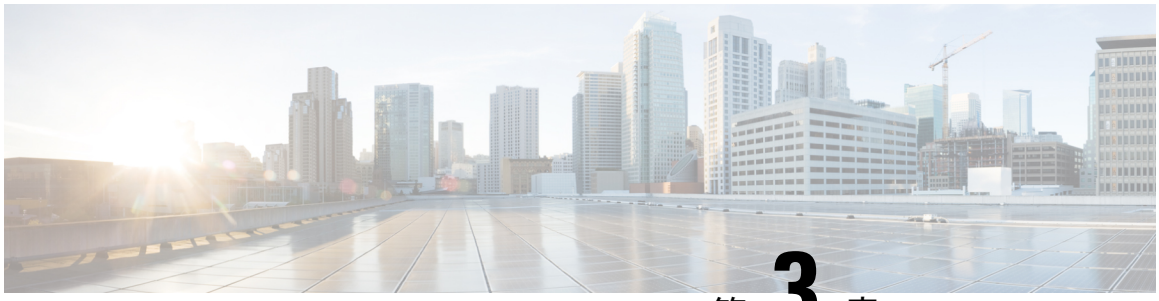
QoS の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	QoS の機能	QoS により、他のトラフィックタイプの代わりに特定のトラフィックタイプを優先的に処理できます。QoS を設定しない場合、デバイスはパケットの内容やサイズに関係なく、各パケットにベスト エフォート型のサービスを提供します。 （注） このリリースでは、コンバージドアクセスはサポートされません。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 3 章

重み付けランダム早期検出の設定

- ネットワーク輻輳の回避 （137 ページ）
- テール ドロップ （137 ページ）
- 重み付けランダム早期検出 （138 ページ）
- WRED 設定の制限 （139 ページ）
- WRED 使用上の注意事項 （139 ページ）
- WRED の設定 （140 ページ）
- WRED の設定例 （144 ページ）
- 階層化 QoS を使用した WRED のサポート （144 ページ）
- WRED 設定の確認 （145 ページ）
- WRED 設定のベスト プラクティス （146 ページ）
- 重み付けランダム早期検出の機能履歴 （148 ページ）

ネットワーク輻輳の回避

異種ネットワークには、アプリケーションが使用する異なるプロトコルが含まれており、これにより、ファイル転送などの時間依存が比較的少ないアプリケーションのニーズに対処しながら、タイムクリティカルなアプリケーションに応えるためにトラフィックの優先順位を付ける必要が生じています。ネットワーク内のデバイス間で単一のデータ パスを共有するさまざまなタイプのトラフィックをサポートするようにネットワークが設定されている場合、輻輳回避メカニズムを実装することにより、さまざまなタイプのトラフィックを公平に処理し、共通のネットワーク ボトルネックでの輻輳を回避できます。輻輳回避メカニズムは、パケットのドロップにより実現します。

ランダム早期検出（RED）は、ネットワークで一般的に使用される輻輳回避メカニズムです。

テール ドロップ

テール ドロップでは、すべてのトラフィックを平等に扱い、サービス クラス内では差別化しません。出力キューが一杯でテール ドロップが有効な場合、輻輳が解消されてキューが一杯でなくなるまでパケットはドロップされます。

重み付けランダム早期検出

RED メカニズムは、TCP の輻輳制御メカニズムを利用します。輻輳が頻繁に発生する前にパケットがランダムにドロップされます。パケット送信元が TCP を使用する場合、送信元はすべてのパケットが宛先に届くようになるまで送信速度を下げます。これは輻輳が解消されたことを示します。RED を、TCP のパケットの転送速度を下げる方法として使用できます。TCP は停止するだけでなく、素早く再起動して、ネットワークがサポート可能なレートに伝送レートを対応させます。

WRED は、シスコが実装している RED です。RED アルゴリズムの機能と、IP プレシデンス、DiffServ コード ポイント（DSCP）、またはサービス クラス（CoS）の値を組み合わせています。

WRED の仕組み

WREDは、出力インターフェイスにネットワーク混雑の兆候が表れた際に、選択的にパケットをドロップしてテール ドロップの確率を減らします。WRED は、キューが一杯になるまで待機するのではなく、一部のパケットを早期にドロップします。そのため、一度に大量のパケットをドロップすることを防ぎ、TCP グローバル同期の可能性を最小限に抑えます。

Approximate Fair Drop（AFD）は、パケットのドロップ確率を決定するアクティブキュー管理（AQM）アルゴリズムです。パケットをドロップする確率は、入力時のフローの着信レート計算と現在のキュー長によって異なります。

AFD ベースの WRED は、有線ネットワークポートに実装されます。

AFD ベースの WRED は、WRED の優先的なドロップ動作をエミュレートします。この優先的なドロップ動作は、WRED の対応するドロップしきい値に基づいて AFD サブクラスの重みを変更することで実現します。物理キュー内では、重みが大きいトラフィクのドロップ確率は、重みの小さいトラフィックよりも低くなります。

- 各 WRED 対応キューには、上限と下限のしきい値があります。
- 優先度の高いサブクラスには 大きな AFD の重みが設定されます。
- サブクラスは、最も低い WRED minThreshold に基づいて昇順でソートされます。

WRED 重み計算

AFD の重みは、下限と上限のしきい値を使用して計算されます。AFD は、WRED の上限と WRED の下限のしきい値の平均を表す調整されたインデックスです。

パケットがインターフェイスに着信すると、次のイベントが発生します。

1. ドロップ確率が計算されます。AFD の重みが減少するほど、ドロップ確率は高くなります。つまり、下限と上限のしきい値の平均が小さいほど、ドロップ確率は高くなります。

2. WREDは、パケットのドロップを決定する前に、パケットフローのプライオリティとしきい値を検討します。CoS、DSCP、またはIP Precedenceの値は、指定されたしきい値にマッピングされます。これらのしきい値を超えると、これらのしきい値にマッピングされた設定値を持つパケットはドロップの対象になります。高いしきい値に割り当てられたCoS、DSCP、またはIP Precedence値を持つその他のパケットは、キューに入れられます。このプロセスにより、プライオリティの高いフローがそのまま維持され、パケット伝送の遅延が最小限に抑えられます。
3. パケットがWREDを使用してドロップされない場合、テールドロップされます。

WRED 設定の制限

- デフォルトでは、重み付きテールドロップ（WTD）がすべてのキューでイネーブルになっています。
- WREDはキューごとに有効または無効にできます。WREDを無効にすると、WTDがターゲットキューに適用されます。WREDプロファイルを持つポリシーマップは出力ポリシーとして物理ポート上にのみ設定されます。
- WREDは、ネットワーク ポート キューのみでサポートされており、内部CPUキューとスタック キューではサポートされていません。
- 各WRED物理キューは、一意のWREDしきい値ペア設定を使用して3つのサブキューをサポートできます。
- WREDとともに、ポリシーマップで帯域幅または形状を設定することを確認します。
- すべてのWREDしきい値は必ずパーセンテージ モードで指定します。
- WREDしきい値ペアのマッピングは、対応する一致フィルタを使用してクラスマップフィルタをマッピングすることで行います。

  「any」一致フィルタが設定されたクラスマップをお勧めします。
- プライオリティトラフィックのWREDはサポートされていません。
- WREDとキュー制限は、同じポリシーではサポートされません。
- 有線ポートは最大で8つの物理キューをサポートします。そのうちの4つの物理キューでそれぞれが3つのしきい値ペアを持つWREDを設定できます。残りのキューは、WTDで設定されます。5つ以上のWREDキューを持つポリシーは拒否されます。

WRED 使用上の注意事項

AFDベースのWRED機能を設定するには、ポリシー マップを指定し、クラスを追加します。**random-detect** コマンドを使用し、ドロップ確率の計算にWREDが使用する方式を（dscp-based/cos-based/cos-based引数を使用して）指定します。

（注） ポリシーは作業中に変更できます。AFD の重みが自動的に再計算されます。

WRED は IPv4/IPv6、マルチキャストなどのどのような種類のトラフィックにも設定できます。WRED は、8 つのキューイングクラスすべてでサポートされます。

random-detect コマンドを使用して WRED を設定する場合は次の点を考慮してください。

- dscp-based 引数を使用する場合、WRED は DSCP 値を使用してドロップ確率を計算します。
- cos-based 引数を使用する場合、WRED は CoS 値を使用してドロップ確率を計算します。
- デフォルトでは、WRED はドロップ確率の計算に IP precedence 値を使用します。**precedence-based** 引数がデフォルトであり、CLI には表示されません。

（注） **show run policy-map** *policy-map* コマンドは、**random-detect** コマンドで precedence が設定されていても、「precedence」を表示しません。

- dscp-based 引数と precedence-based 引数は、相互に排他的です。
- 8 つの物理キューを、それぞれ異なる WRED プロファイルで設定できます。

WRED の設定

DSCP 値に基づく WRED の設定

DSCP 値に基づいて WRED プロファイルをパケット モードで設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**class-map** *match-criteria class-name* **例：** `device(config)# class-map match-any CS`	クラスマップに一致基準を設定します。 推奨する一致基準は match-any です。
ステップ 2	**match** *class-map-name* **例：** `device(config-cmap)#match dscp CS1`	クラスマップを照合します。

	コマンドまたはアクション	目的
ステップ 3	**policy-map** *name* 例： `device(config)#policy-map PWRED`	作成する WRED プロファイル ポリシーの名前を指定します。
ステップ 4	**class** *class-name* 例： `device(config-pmap)#class CS`	ポリシーに関連付けるクラスの名前を指定します。
ステップ 5	Use either **bandwidth** {*kbps*| **remaining** *percentage* | **percent** *percentage*} or **shape** { **average** | **peak** }*cir* 例： `device(config-pmap-c)#bandwidth percent 10`	ポリシーマップに属しているクラスに割り当てる帯域幅またはトラフィック シェーピングを指定します。
ステップ 6	**random-detect** *dscp-based* 例： `device(config-pmap-c)#random-detect dscp-based`	パケットのドロップ確率を計算する際には DSCP 値を使用するように WRED を設定します。
ステップ 7	**random-detect dscp** *dscp-value* **percent** *minThreshold maxThreshold* 例： `device(config-pmap-c)#random-detect dscp cs1 percent 10 20`	最小しきい値および最大しきい値をパーセンテージで指定します。
ステップ 8	**interface** *interface-name* 例： `device(config)#interface HundredGigE1/0/2`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 9	**service-policy output** ポリシーマップ 例： `device(config-if)#service-policy output pwred`	ポリシー マップを出力インターフェイスに付加します。

サービス クラス値に基づく WRED の設定

サービス クラス（CoS）値に基づいて WRED プロファイルをパケット モードで設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**class-map** *match-criteria class-name* 例： `device(config)# class-map match-any CS`	クラスマップに一致基準を設定します。 推奨する一致基準は match-any です。
ステップ 2	**match** *class-map-name* 例： `device(config-cmap)#match cos 3`	クラスマップを照合します。
ステップ 3	**policy-map** *name* 例： `device(config)#policy-map PWRED`	作成する WRED プロファイル ポリシーの名前を指定します。
ステップ 4	**class** *class-name* 例： `device(config-pmap)#class CS`	ポリシーに関連付けるクラスの名前を指定します。
ステップ 5	**bandwidth** {*kbps*| **remaining** *percentage* | **percent** *percentage*} 例： `device(config-pmap-c)#bandwidth percent 10`	ポリシー マップに属しているクラスに割り当てる帯域幅を指定します。
ステップ 6	**random-detect** *cos-based* 例： `device(config-pmap-c)#random-detect cos-based`	パケットのドロップ確率を計算する際には CoS 値を使用するように WRED を設定します。
ステップ 7	**random-detect cos** *cos-value* **percent** *minThreshold maxThreshold* 例： `device(config-pmap-c)#random-detect cos 3 percent 10 20`	最小しきい値および最大しきい値をパーセンテージで指定します。
ステップ 8	**interface** *interface-name* 例： `device(config)# interface HundredGigE1/0/2`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 9	**service-policy output** ポリシーマップ 例： `device(config-if)#service-policy output pwred`	ポリシー マップを出力インターフェイスに付加します。

IP プレシデンス値に基づく WRED の設定

IP プレシデンス値に基づいて WRED プロファイルをパケット モードで設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**class-map** *match-criteria class-name* 例： `device(config)# class-map match-any CS`	クラスマップに一致基準を設定します。 推奨する一致基準は match-any です。
ステップ 2	**match** *class-map-name* 例： `device(config-cmap)#match precedence 3`	クラスマップを照合します。
ステップ 3	**policy-map** *name* 例： `device(config)#policy-map pwred`	作成する WRED プロファイル ポリシーの名前を指定します。
ステップ 4	**class** *class-name* 例： `device(config-pmap)#class CS`	ポリシーに関連付けるクラスの名前を指定します。
ステップ 5	**bandwidth** {*kbps*| **remaining** *percentage* | **percent** *percentage*} 例： `device(config-pmap-c)#bandwidth percent 10`	ポリシー マップに属しているクラスに割り当てる帯域幅を指定します。
ステップ 6	**random-detect** *precedence-based* 例： `device(config-pmap-c)#random-detect precedence-based`	パケットのドロップ確率を計算する際には IP プレシデンス値を使用するように WRED を設定します。
ステップ 7	**random-detect precedence** *precedence-value* **percent** *minThreshold maxThreshold* 例： `device(config-pmap-c)#random-detect precedence 3 percent 10 20`	最小しきい値および最大しきい値をパーセンテージで指定します。
ステップ 8	**interface** *interface-name* 例：	インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`device(config)#interface HundredGigE1/0/2`	
ステップ 9	**service-policy output** ポリシーマップ 例： `device(config-if)#service-policy output pwred`	ポリシー マップを出力インターフェイスに付加します。

WRED の設定例

次に、クラス CS の DSCP プロファイルを使用するように WRED をイネーブルにする例を示します。この例では、cs1、cs2、および cs3 という 3 つのサブクラスを WRED の最小しきい値および最大しきい値で設定し、最終的にはポリシーを 100 ギガビット イーサネット インターフェイス 8 に適用します。

```
Device(config)# class-map match-any CS
Device(config-cmap)# match dscp cs1
Device(config-cmap)# match dscp cs2
Device(config-cmap)# match dscp cs3
Device(config-cmap)# policy-map PWRED
Device(config-pmap)# class CS
Device(config-pmap-c)# bandwidth percent 10
Device(config-pmap-c)# random-detect dscp-based
Device(config-pmap-c)# random-detect dscp cs1 percent 10 20
Device(config-pmap-c)# random-detect dscp cs2 percent 20 30
Device(config-pmap-c)# random-detect dscp cs3 percent 34 44
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface HundredGigE1/0/8
Device(config-if)# service-policy output PWRED
```

階層化 QoS を使用した WRED のサポート

階層型 QoS では、トラフィック管理をより細かい粒度で実行する、複数のポリシー レベルで QoS 動作を指定できます。

HQoSの場合、子ポリシーでのみWREDが許可され、親ポリシーでは許可されません。親ポリシーにシェーピングを、子ポリシーに WRED を設定できます。

次に、親ポリシー **pwred-parent** を帯域幅の 10 パーセントでシェーピングしたトラフィックで設定し、それを DSCP ベースの WRED に設定されたその子ポリシー **pwred-child** に適用する例を示します。

```
policy-map PWRED-CHILD
 class CWRED
    bandwidth percent 10
    random-detect dscp-based
    random-detect dscp 1 percent 10 20
    random-detect dscp 10 percent 20 30
```

```
policy-map PWRED-PARENT
  class class-default
  shape average percent 10
  service-policy PWRED-CHILD
```

次に、HQoS WRED 設定を確認する show コマンドを示します。

```
device# show policy-map PWRED-PARENT
  policy Map PWRED-PARENT
    class class-default
      average Rate Traffic Shaping
      cir 30%
      service-policy PWRED-CHILD
policy-map PWRED-CHILD
 class CWRED
  bandwidth percent 10
  random-detect dscp-based
  random-detect dscp 1 percent 10 20
  random-detect dscp 10 percent 20 30
policy-map PWRED-PARENT
 class class-default
   shape average percent 30
   service-policy PWRED-CHILD
```

WRED 設定の確認

次の show コマンドを使用して、WRED の設定を確認します。

手順

ステップ 1 **show policy-map** *policy-map-name*

WRED としきい値のラベルが表示されます。

例：

```
Device# show policy-map PWRED
  Policy Map PWRED
    Class CS
      bandwidth 10 (%)
       percent-based wred

      dscp    min-threshold    max-threshold
      ---------------------------------------
      cs1 (8)      10               20
      cs2 (16)     20               30
      cs3 (24)     34               44
      default (0)   -
```

ステップ 2 **show policy-map interface** *interface-name*

WRED AFD の重み、WRED Enq（パケット数およびバイト数）、WRED ドロップ（パケット数およびバイト数）、しきい値ペアに対して設定された DSCP ラベルが表示されます。

（注） トラフィックを開始した後にのみ、このコマンドを使用します。**show policy-map interface** は、トラフィックが送信された後にのみ、WRED 設定が更新されます。

例：

```
Device#show policy-map interface HundredGigE 1/0/2
 HundredGigE1/0/2

  Service-policy output: PWRED

    Class-map: CS (match-any)
      0 packets
      Match:  dscp cs1 (8)
      Match:  dscp cs2 (16)
      Match:  dscp cs3 (24)
      Queueing

      (total drops) 27374016
      (bytes output) 33459200081
      bandwidth 10% (1000000 kbps)

AFD WRED STATS BEGIN
Virtual Class   min/max        Transmit                      Random drop
 AFD Weight


   0          10 / 20        (Byte)33459183360             27374016
 12
                             (Pkts)522799759                  427716

              dscp : 8


   1          20 / 30          Byte)0                        0
 20
                               (Pkts)0                       0

              dscp : 16


   2          34 / 44        (Byte)16721                   0
 31
                               (Pkts)59                      0
              dscp : 24

     Total Drops(Bytes)   : 27374016

     Total Drops(Packets) : 427716
AFD WRED STATS END


    Class-map: class-default (match-any)
      0 packets
      Match: any


      (total drops) 0
      (bytes output) 192
```

WRED 設定のベスト プラクティス

- 3 つの WRED 設定ペアのサポート

各 WRED 物理キュー（AFD キュー）は、一意の WRED しきい値ペア設定を使用して 3 つの WRED 設定ペアをサポートできます。

```
Policy-map P1
    Class CS
        Random-detect dscp-based
        Random-detect dscp CS1 percent 10 20      // WRED pair 1
        Random-detect dscp CS2  percent 20 30      // WRED pair 2
        Random-detect dscp CS3 percent 30 40      // WRED pair 3
Class-map match-any CS
        match cs1
        match cs2
        match cs3
```

- **WRED 設定ペアの追加**

  重複するしきい値ペアを WRED 設定ペアに追加できます。

```
Policy-map P1
    Class CS
       Random-detect dscp-based
       Random-detect dscp CS1 percent 10 20       // WRED pair 1
       Random-detect dscp CS2 percent 20 30       // WRED pair 2
       Random-detect dscp CS3 percent 30 40       // WRED pair 3
       Random-detect dscp CS4 percent  30 40     ==> belongs to WRED pair 3
       Random-detect dscp CS5 percent 20 30     ==> belongs to WRED pair 2
Class-map match-any CS
       match cs1
       match cs2
       match cs3
       match cs4  >>
       match cs5  >>
```

- **デフォルトの WRED ペア**

  2 つ以下の WRED ペアが設定されている場合、WRED に参加しているどのクラスマップ フィルタも最大しきい値（100, 100）でデフォルトの 3 番目の WRED ペアに割り当てられます。

```
Policy-map P1
    Class CS
        Random-detect dscp-based
        Random-detect dscp CS1 percent 10 20   // WRED pair 1
        Random-detect dscp CS2 percent 20 30   // WRED pair 2
Class-map match-any CS
        match CS1
        match CS2
        match CS3
        match CS4
```

  この場合は、CS3 と CS4 のクラスはしきい値（100, 100）で WRED ペア 3 にマッピングされます。

- **一致しない設定の拒否**

  クラスマップ内に一致フィルタがない場合に random-detect を設定すると、ポリシーのインストールが拒否されます。

```
Class-map match-any CS
       match CS1
       match CS2
       match CS5
```

```
Policy-map P1
    Class CS
        Shape average percent 10
        Random-detect dscp-based
        Random-detect dscp CS1 percent 10 20   // WRED pair 1
        Random-detect dscp CS2 percent 20 30   // WRED pair 2
        Random-detect dscp CS3 percent 30 40   // WRED pair 3  ===> Mismatched
sub-class.
```

このポリシーを出力側のインターフェイスに適用すると、クラスマップ値が不正であるとして、インストール時にそのポリシーは失敗します。

```
device(config)# int Fo1/0/5
device(config-if)# service-policy output P1
device(config-if)#
*Feb 20 17:33:16.964: %IOSXE-5-PLATFORM: Switch 1 R0/0: fed: WRED POLICY INSTALL
FAILURE.Invalid WRED filter mark: 24 in class-map: CS
*Feb 20 17:33:16.965: %FED_QOS_ERRMSG-3-LABEL_2_QUEUE_MAPPING_HW_ERROR: Switch 1
R0/0: fed: Failed to detach queue-map for FortyGigabitEthernet1/0/5: code 2
```

重み付けランダム早期検出の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	重み付けランダム早期検出メカニズム	WRED は、ネットワーク内の輻輳を回避するメカニズムです。WREDは、出力インターフェイスにネットワーク混雑の兆候が表れた際に、選択的にパケットをドロップしてテール ドロップの確率を減らし、多数のパケットが一度にドロップされないようにします。次の値に基づいて動作するように WRED を設定できます。 • DiffServ コードポイント • IP Precedence • サービスクラス

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。