



モジュラ Quality of Service コマンドライン インターフェイスの QoS

このモジュールでは、モジュラ Quality of Service (QoS) コマンドラインインターフェイス (CLI) (MQC) を使用した QoS 機能の適用に関する概念と、MQC の設定タスクについて説明します。MQC を使用すると、トラフィック クラスの定義、トラフィック ポリシー (ポリシー マップ) の作成、およびインターフェイスへのトラフィック ポリシーの適用が可能になります。トラフィック ポリシーには、トラフィック クラスに適用する QoS 機能が含まれます。

- [機能情報の確認, 1 ページ](#)
- [Cisco cBR シリーズ ルータに関するハードウェア互換性マトリクス, 2 ページ](#)
- [MQC を使用した QoS 機能の適用に対する制約事項, 3 ページ](#)
- [概要, 3 ページ](#)
- [MQC を使用した QoS 機能の適用方法, 11 ページ](#)
- [MQC を使用した QoS 機能の適用の設定例, 16 ページ](#)
- [port-channel インターフェイスの入力 MQC の設定方法, 20 ページ](#)
- [例 : port-channel インターフェイスの入力 MQC の設定, 22 ページ](#)
- [その他の参考資料, 23 ページ](#)
- [モジュラ Quality of Service コマンドライン インターフェイスの QoS に関する機能情報, 24 ページ](#)

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、「[Bug Search Tool](#)」およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。この

モジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

Cisco cBR シリーズ ルータに関するハードウェア互換性マトリクス



(注) Cisco IOS-XE の特定のリリースで追加されたハードウェア コンポーネントは、特に明記しない限り、以降のすべてのリリースでもサポートされます。

表 1: Cisco cBR シリーズ ルータに関するハードウェア互換性マトリクス

Cisco CMTS プラットフォーム	プロセッサ エンジン	インターフェイス カード
Cisco cBR-8 コンバージドブロードバンドルータ	<p>Cisco IOS-XE リリース 16.5.1 以降のリリース</p> <p>Cisco cBR-8 スーパーバイザ :</p> <ul style="list-style-type: none"> • PID : CBR-CCAP-SUP-160G • PID : CBR-CCAP-SUP-60G • PID : CBR-SUP-8X10G-PIC 	<p>Cisco IOS-XE リリース 16.5.1 以降のリリース</p> <p>Cisco cBR-8 CCAP ラインカード :</p> <ul style="list-style-type: none"> • PID : CBR-LC-8D30-16U30 • PID : CBR-LC-8D31-16U30 • PID : CBR-RF-PIC • PID : CBR-RF-PROT-PIC • PID : CBR-CCAP-LC-40G-R <p>Cisco cBR-8 ダウンストリーム PHY モジュール :</p> <ul style="list-style-type: none"> • PID : CBR-D30-DS-MOD • PID : CBR-D31-DS-MOD <p>Cisco cBR-8 アップストリーム PHY モジュール :</p> <ul style="list-style-type: none"> • PID : CBR-D30-US-MOD • PID : CBR-D31-US-MOD

MQC を使用した QoS 機能の適用に対する制約事項

MQC ベースの QoS は、Internetwork Packet Exchange (IPX)、DECnet、AppleTalk などとは異なり、レガシーレイヤ2プロトコルパケットの分類をサポートしません。このタイプのパケットが一般的なレイヤ2 トンネリングメカニズムによって転送されると、パケットはMQCにより処理されますが、プロトコル分類はされません。その結果、レイヤ2 トンネルのレガシープロトコルトラフィックは、「match any」クラスまたは class-default によってのみ照合されます。

サポートされる QoS ポリシーマップとクラスマップの数は、プラットフォームとリリースにより異なります。



(注) ポリシーマップの制限は、適用されるポリシーマップのインスタンスの数ではなく、ポリシーマップの定義を参照します。

概要

MQC 構造

MQC (モジュラ QoS コマンドラインインターフェイス (CLI)) では、QoS グループ値に基づいてパケット分類とマーキングを設定できます。MQC CLI では、トラフィッククラスおよびポリシーを作成し、QoS機能 (パケット分類など) をイネーブルにして、それらのポリシーをインターフェイスに適用することができます。

MQC 構造では、エンティティ (トラフィッククラス、ポリシーマップ、サービスポリシー) を開発する必要があります。

トラフィッククラスの要素

トラフィッククラスに含まれる3つの主要な要素は、トラフィッククラス名、一連の **match** コマンド、およびトラフィッククラスで複数の **match** コマンドが使用される場合にそれらの **match** コマンドを評価する方法です。

match コマンドは、パケットを分類するために使用します。パケットがチェックされ、**match** コマンドで指定された条件を満たすかどうか判断されます。パケットが指定された条件を満たしている場合、パケットはそのクラスのメンバーと見なされます。一致条件を満たしていないパケットは、デフォルトトラフィッククラスのメンバーとして分類されます。

利用可能な **match** コマンド

次の表に、MQCで使用できる **match** コマンドの一部を示します。使用可能な **match** コマンドは、Cisco IOS XE のリリースによって異なります。コマンドおよびコマンドシンタックスの詳細については、『Cisco IOS Quality of Service Solutions Command Reference』を参照してください。

表 2: MQC で使用可能な **match** コマンド

コマンド	目的
matchaccess-group	指定したアクセスコントロールリスト (ACL) に基づいて、クラスマップの一致基準を設定します。
matchany	すべてのパケットに対して適切に一致する基準となる、クラス マップの一致基準を設定します。
matchcos	レイヤ 2 サービス クラス (CoS) マーキングに基づいてパケットを照合します。
matchdestination-addressmac	宛先 MAC アドレスを一致条件として使用します。
matchdiscard-class	特定の廃棄クラスの packets を照合します。
match [ip] dscp	特定の IP Diffserv コードポイント (DSCP) 値を一致条件として識別します。1 つの match 文に最大 8 つの DSCP 値を含めることができます。
matchinput-interface	指定された入力インターフェイスを一致基準として使用するクラス マップを設定します。
matchiprtp	Real-Time Transport Protocol (RTP) ポートを一致条件として使用するようクラスマップを設定します。
matchmplsexperimental	指定されたマルチプロトコル ラベル スウィッチング (MPLS) の EXP フィールドの値を一致基準として使用するクラスマップを設定します。
matchmplsexperimentaltopmost	最上位ラベル内の MPLSEXP 値を照合します。

コマンド	目的
matchnot	<p>成功しない一致基準として使用する、1 つの一致基準値を指定します。</p> <p>(注) matchnot コマンドは、一致基準として使用する特定の match パラメータを指定する代わりに、パケットがクラスのメンバーとして分類されるのを防ぐ一致基準を指定するために使用します。たとえば、トラフィック クラスの設定中に matchnotqos-group6 コマンドを発行すると、QoS グループ 6 だけが、成功する一致基準として考慮されない QoS グループ値となります。他の QoS グループ値は成功する一致基準となります。</p>
matchpacketlength	IP ヘッダー中のレイヤ 3 パケット長をクラスマップ中の一致条件として指定します。
matchport-type	トラフィックを、クラス マップのポートタイプに基づいて照合します。
match [ip] precedence	IP precedence 値を一致基準として識別します。
matchprotocol	<p>指定されたプロトコルに基づいて、クラスマップの一致基準を設定します。</p> <p>(注) 個別の matchprotocol (NBAR) コマンドを使用すると、Network-Based Application Recognition (NBAR) が認識しているプロトコルタイプでトラフィックを照合するように NBAR を設定できます。</p>
matchprotocolfasttrack	FastTrack ピアツーピア トラフィックを照合するように NBAR を設定します。
matchprotocolgnutella	Gnutella ピアツーピア トラフィックを照合するように NBAR を設定します。
matchprotocolhttp	URL、ホスト、多目的インターネットメール拡張 (MIME) タイプ、HTTP パケットヘッダー内のフィールドによってハイパーテキスト転送プロトコル (HTTP) トラフィックを照合するように NBAR を設定します。

コマンド	目的
<code>matchprotocolrtp</code>	RTP トラフィックを照合するように NBAR を設定します。
<code>matchqos-group</code>	特定の QoS グループ値を一致基準として識別します。
<code>matchsource-addressmac</code>	送信元 MAC アドレスを一致基準として使用します。

1つのトラフィック クラスでの複数の `match` コマンド

トラフィック クラスに複数の `match` コマンドが含まれている場合、それらの `match` コマンドの評価方法を指定する必要があります。指定するには、`class-map` コマンドの `match-any` キーワードまたは `match-all` キーワードを使用します。`match-any` キーワードと `match-all` キーワードについては、次の点に注意してください。

- `match-any` キーワードを指定した場合、トラフィック クラスによって評価されるトラフィックは、指定した基準の 1つに一致する必要があります。
- `match-all` キーワードを指定した場合、トラフィック クラスによって評価されるトラフィックは、指定したすべての基準に一致する必要があります。
- どちらのキーワードも指定しない場合、トラフィック クラスによって評価されるトラフィックは、指定したすべての基準に一致する必要があります（つまり、`match-all` キーワードの動作が適用されます）。

トラフィック ポリシーの要素

トラフィック ポリシーには、トラフィック ポリシー名、トラフィック クラス (`class` コマンドで指定)、QoS 機能をイネーブルにするために使用するコマンドの 3つの要素が含まれています。

ポリシー マップをインターフェイスに適用すると (`service-policy` コマンドを使用)、トラフィック ポリシー (ポリシー マップ) は、イネーブルにした QoS 機能をトラフィック クラスに適用します。



(注)

パケットは、トラフィック ポリシー内のいずれかのトラフィック クラスだけに一致します。パケットがトラフィック ポリシー内の複数のトラフィック クラスに一致する場合、ポリシーで定義されている最初のトラフィック クラスが使用されます。

QoS 機能をイネーブルにするために使用するコマンド

QoS 機能をイネーブルにするために使用するコマンドは、Cisco IOS XE リリースごとに異なります。以下の表に、使用可能なコマンドとそれによってイネーブルになる QoS 機能の一部を示します。コマンド構文の詳細については、『Cisco IOS QoS Command Reference』を参照してください。

イネーブルにする特定の QoS 機能の詳細については、『Cisco IOS XE Quality of Service Solutions Configuration Guide』の適切なモジュールを参照してください。

表 3: QoS 機能をイネーブルにするために使用するコマンド

コマンド	目的
bandwidth	クラスの最小帯域幅保証を設定します。
bandwidthremaining	クラスの過剰重量を設定します。
fair-queue	トラフィック クラス内のフローベースのキューイング機能をイネーブルにします。
fair-queue pre-classify	qos pre-classify コマンドを設定し、それを均等化キューに使用できるかどうかを確認します。トンネルインターフェイスで qos pre-classify コマンドが有効にされた後、 fair-queue pre-classify コマンドがポリシー マップ用に有効にされると、ポリシー マップはトンネル インターフェイスまたは物理インターフェイスのいずれかに関連付けられます。均等化キューのハッシュアルゴリズムには、トンネルの内部 IP ヘッダーが使用されます。
drop	指定したトラフィック クラスのパケットを廃棄します。
police	トラフィック ポリシングを設定します。
police(percent)	インターフェイスで利用可能な帯域幅の割合に基づいてトラフィック ポリシングを設定します。
police(tworates)	認定情報レート (CIR) と最大情報レート (PIR) の 2 つのレートを使用したトラフィック ポリシングを設定します。
priority	ポリシーマップに属するトラフィックのクラスにプライオリティを与えます。

コマンド	目的
queue-limit	ポリシーマップで設定されているクラスに対してキューが保持できるパケットの最大数を指定または変更します。
random-detect	重み付けランダム早期検出 (WRED) をイネーブルにします。
random-detectdiscard-class	ポリシー マップ内のクラスの discard-class 値に対し、WRED パラメータを設定します。
random-detectdiscard-class-based	パケットの廃棄クラス値に基づく WRED を設定します。
random-detectexponential-weighting-constant	クラス用に予約されたキューの平均キューサイズ計算用の指数加重係数を設定します。
random-detectprecedence	ポリシーマップ内のクラスポリシーに対する、特定の IP precedence の WRED パラメータを設定します。
service-policy	一致基準として使用するトラフィック ポリシーの名前を指定します (トラフィック ポリシーを互いにネストさせるため (階層型トラフィックポリシー))。
setatm-clp	ポリシーマップを設定するときのセル損失率優先度 (CLP) ビットを設定します。
setcos	送信パケットのレイヤ 2 サービス クラス (CoS) 値を設定します。
setdiscard-class	discard-class 値でパケットをマークします。
set[ip] dscp	タイプオブサービス (ToS) バイト内の DiffServ コード ポイント (DSCP) 値を設定することでパケットをマークします。
setfr-de	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に対し、フレームリレーフレームのアドレス フィールドの廃棄適性 (DE) ビット設定を 1 に変更します。

コマンド	目的
setmplsexperimental	パケットが指定したポリシーマップに一致する場合に MPLS ビットを設定する値を指定します。
setprecedence	パケット ヘッダーに precedence 値を設定します。
setqos-group	後でパケットを分類するために使用できる QoS グループ ID を設定します。
shape	指定したアルゴリズムに従って、指示されたビットレートまでトラフィックをシェーピングします。

ネストしたトラフィック クラス

MQC では、必ずしも 1 つのトラフィック クラスだけを 1 つのトラフィック ポリシーに関連付ける必要はありません。

パケットが複数の一致基準を満たしているシナリオでは、MQC により複数のトラフィック クラスを 1 つのトラフィック ポリシー（ネストしたトラフィック クラスとも呼ぶ）に関連付けることができます。これを行うには、**matchclass-map** コマンドを使用します（これらをネストしたクラス マップまたは MQC 階層型クラス マップと呼びます）。このコマンドで、1 つのトラフィック クラス内で **match-any** 特性と **match-all** 特性を組み合わせる唯一の方法が提供されます。これを実行することにより、1 つの一致基準評価命令（**match-any** と **match-all** のどちらか）を使用してトラフィック クラスを作成して、そのトラフィック クラスを別の一致基準タイプを使用するトラフィック クラス内に一致基準として使用できます。たとえば、**match-any** 命令を使用して作成したトラフィック クラスは **match-all** 命令を一致基準として使用して設定したクラスを使用する必要があります、その逆の場合も同様です。

考えられるシナリオは次のとおりです。A、B、C、および D が、すべて異なる一致基準であるとし、A、B、または C かつ D（A または B または（C かつ D））に一致するトラフィックをトラフィック クラスに属するものとして分類するとします。ネストしたトラフィック クラスがない場合、トラフィックがトラフィック クラスの一部であると見なされるためには、4 つの一致基準すべてに一致するか（A かつ B かつ C かつ D）、いずれかの一致基準に一致する必要があります（A または B または C または D）。「かつ」（**match-all**）文と「または」（**match-any**）文をトラフィック クラス内で組み合わせることはできないため、目的の設定を実現できません。

解決策：C と D に対して **match-all** を使用する 1 つのトラフィック クラスを作成し（これを条件 E と呼びます）、A、B、E を使用して新しい **match-any** トラフィック クラスを作成します。新しいトラフィック クラスの評価順序は正しくなります（A または B または E、つまり A または B または（C かつ D））。

class-map コマンドの match-all キーワードと match-any キーワード

トラフィッククラスを作成するときに使用するコマンドの1つが **class-map** コマンドです。 **class-map** コマンドのコマンドシンタックスには、2つのキーワード **match-all** と **match-any** が含まれています。 **match-all** キーワードと **match-any** キーワードの指定が必要になるのは、トラフィッククラスで複数の一致条件を設定する場合だけです。これらのキーワードについて、次の点に注意してください。

- 指定したトラフィック クラスにパケットを分類するために、トラフィック クラス内のすべての一致基準に一致する必要がある場合、 **match-all** キーワードを使用します。
- 指定したトラフィック クラスにパケットを分類するために、トラフィック クラス内の1つの一致基準だけに一致する必要がある場合に、 **match-any** キーワードを使用します。
- **match-all** キーワードも **match-any** キーワードも指定しない場合、トラフィック クラスの動作は、 **match-all** キーワードを指定した場合と同じになります。

service-policy コマンドの input および output キーワード

一般的な規則として、トラフィック ポリシーで設定する QoS 機能は、インターフェイスで受信されるパケットか、インターフェイスで送信されるパケットに適用できます。そのため、 **service-policy** コマンドを使用する場合は、 **input** キーワードまたは **output** キーワードを使用してトラフィック ポリシーの方向を指定する必要があります。

たとえば、 **service-policy output policy-map1** コマンドは、トラフィック ポリシーの QoS 機能を出力方向のインターフェイスに適用します。インターフェイス（出力）から送信されるすべてのパケットが、 **policy-map1** という名前のトラフィック ポリシーで指定された基準に従って評価されます。



- (注) Cisco のリリースでは、キューイング メカニズムは入力方向ではサポートされていません。非キューイング メカニズム（トラフィック ポリシングやトラフィック マーキングなど）は、入力方向でサポートされています。また、送信元 MAC アドレスに基づくトラフィックの分類（ **match source-address mac** コマンドを使用）は、入力方向でのみサポートされています。

MQC を使用して QoS 機能を適用することの利点

MQC 構造では、一度トラフィック ポリシー（ポリシー マップ）を作成すると、必要な数のトラフィッククラスに適用できます。また、トラフィック ポリシーを必要な数のインターフェイスに適用できます。

MQC を使用した QoS 機能の適用方法

トラフィック クラスの作成

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 クラス名を指定します。次に、1つ以上の **match** コマンドを使用して、適切な一致基準を指定します。指定した基準に一致するパケットがトラフィック クラスに分類されます。**class-map** コマンドの **match-all** キーワードと **match-any** キーワードの詳細については、「class-map コマンドの match-all キーワードと match-any キーワード」の項を参照してください。



(注) **matchcos** コマンドはステップ 4 に記載されています。**matchcos** コマンドは、使用できる **match** コマンドの 1 つに過ぎません。他の利用可能な **match** コマンドについては、「class-map コマンドの match-all キーワードと match-any キーワード」の項を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Router# 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 3	class-map [match-all match-any] class-map-name 例： Router(config)# class-map match-any class1	クラス マップで使用するクラスを作成し、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。 • クラスマップは、パケットを指定したクラスに照合するために使用します。 • クラス名を入力します。

	コマンドまたはアクション	目的
		(注) match-all キーワードは、すべての一致基準が満たされることが必要であることを指定します。 match-any キーワードは、いずれかの一致条件が満たされることが必要であることを指定します。これらのキーワードは、複数の match コマンドを指定する場合にだけ使用します。
ステップ 4	matchcos cos-number 例 : <pre>Router(config-cmap) # match cos 2</pre>	レイヤ 2 サービスクラス (CoS) 番号に基づいてパケットを照合します。 <ul style="list-style-type: none"> • CoS 番号を入力します。 (注) matchcos コマンドは、使用できる match コマンドの一例です。他の利用可能な match コマンドについては、「class-map コマンドの match-all キーワードと match-any キーワード」の項を参照してください。
ステップ 5	必要に応じて追加の match コマンドを入力します。追加のコマンドが不要な場合はステップ 6 に進みます。	--
ステップ 6	end 例 : <pre>Router(config-cmap) # end</pre>	(任意) QoS class-map コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

トラフィック ポリシーの作成



(注) **bandwidth** コマンドはステップ 5 に記載されています。**bandwidth** コマンドは、QoS 機能を有効にするためにポリシー マップで使用できるコマンドの一例です (ここでは、クラスベース均等化キューイング (CBWFQ))。利用可能なその他のコマンドの詳細については、「トラフィック ポリシーの要素」セクションを参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Router# 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 3	policy-map <i>policy-map-name</i> 例： Router(config)# policy-map policy1	トラフィック ポリシーの名前を作成または指定し、QoS 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。 • ポリシー マップ名を入力します。
ステップ 4	class {<i>class-name</i> class-default} 例： Router(config-pmap)# class class1	トラフィック クラスの名前を指定し、QoS 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。 (注) この手順により、トラフィック クラスがトラフィック ポリシーに関連付けられません。
ステップ 5	bandwidth {<i>bandwidth-kbps</i> percent <i>percent</i>} 例： Router(config-pmap-c)# bandwidth 3000	(任意) トラフィック クラスに対する輻輳期間中の最小帯域幅保証を指定します。 • 最小帯域幅保証は、kb/s 単位か、使用可能な全帯域幅のパーセンテージで指定します。 (注) bandwidth コマンドを使用すると CBWFQ がイネーブルになります。 bandwidth コマンドは、QoS 機能をイネーブルにするためにポリシーマップで使用できるコマンドの一例です。利用可能なその他のコマンドの詳細については、「トラフィック ポリシーの要素」セクションを参照してください。
ステップ 6	イネーブルにする追加の QoS 機能に対するコマンドを入力します。他に QoS 機能が必要な場合は、ステップ 7 に進みます。	--

	コマンドまたはアクション	目的
ステップ 7	end 例： <pre>Router(config-pmap-c)# end</pre>	(任意) QoS ポリシーマップ クラス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

MQC を使用したインターフェイスへのトラフィック ポリシーの適用

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： <pre>Router> enable</pre>	特権 EXEC モードをイネーブルにします。 <ul style="list-style-type: none"> パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： <pre>Router# configure terminal</pre>	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 3	interface type number 例： <pre>Router(config)# interface TenGigabitEthernet 4/1/0</pre>	インターフェイス タイプを設定し、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。 <ul style="list-style-type: none"> インターフェイス タイプとインターフェイス番号を入力します。
ステップ 4	service-policy {input output} policy-map-name 例： <pre>Router(config-if)# service-policy input policy1</pre>	ポリシー マップをインターフェイスに付加します。 <ul style="list-style-type: none"> input キーワードまたは output キーワードとポリシー マップ名を入力します。
ステップ 5	end 例： <pre>Router(config-if)# end</pre>	(任意) 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、特権 EXEC モードに戻ります。

トラフィック クラスとトラフィック ポリシー情報の確認

ここで取り上げる show コマンドはいずれも任意で使います。また、入力順も任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	showclass-map 例： Router# show class-map	（任意）すべてのクラス マップとその一致条件を表示します。
ステップ 3	showpolicy-map <i>policy-map-name</i> class <i>class-name</i> 例： Router# show policy-map policy1 class class1	（任意）指定したポリシー マップの指定したクラスの設定を表示します。 • ポリシーマップ名とクラス名を入力します。
ステップ 4	showpolicy-map 例： Router# show policy-map	（任意）存在するすべてのポリシー マップのすべてのクラスの設定を表示します。
ステップ 5	showpolicy-mapinterface <i>type</i> <i>number</i> 例： Router# show policy-map interface TengigabitEthernet 4/1/0	（任意）インターフェイスに適用された入力ポリシーと出力ポリシーの統計情報と設定を表示します。 • インターフェイスタイプと番号を入力します。
ステップ 6	exit 例： Router# exit	（任意）特権 EXEC モードを終了します。

MQC を使用した QoS 機能の適用の設定例

トラフィック クラスの作成

次の例では、トラフィッククラスを作成し、一致基準を定義します。最初のトラフィッククラス (`class1`) では、アクセスコントロールリスト (ACL) 101 を使用し、2つめのトラフィッククラス (`class2`) では ACL 102 を使用します。パケットとこれらの ACL のコンテンツを照合し、そのクラスに属しているかどうかを判断します。

```
class-map class1
  match access-group 101
  exit
class-map class2
  match access-group 102
  end
```

ポリシー マップの作成

次の例では、`class1` と `class2` の2つのクラスに適用する QoS 機能を含むトラフィック ポリシー (`policy1`) を定義します。これらのクラスの一貫基準は、[トラフィック クラスの作成](#)、(16 ページ) ですすでに定義されています。

`class1` では、帯域幅割り当て要求と、そのクラス用に予約されるキューの最大パケット数の制限がポリシーに含まれています。`class2` に対しては、帯域幅割り当て要求だけがポリシーで指定されています。

```
policy-map policy1
  class class1
    bandwidth 3000
    queue-limit 30
  exit
  class class2
    bandwidth 2000
  end
```

例：トラフィック ポリシーのインターフェイスへの適用

次に、既存のトラフィック ポリシーをインターフェイスに適用する例を示します。`policy-map` コマンドを使用してトラフィック ポリシーを定義した後、`service-policy` コマンドをインターフェイス コンフィギュレーション モードで使用して、1つ以上のインターフェイスに適用できます。同じトラフィック ポリシーを複数のインターフェイスに割り当てることができますが、各インター

フェイスには、入力方向と出力方向に対して、それぞれトラフィック ポリシーを1つだけ割り当てることができます。

```
Router(config)# interface TengigabitEthernet 4/1/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
Router(config)# interface TengigabitEthernet 4/1/0
Router(config-if)# service-policy output policy1
Router(config-if)# end
```

match not コマンドの使用

matchnot コマンドを使用して、一致基準として使用していない QoS ポリシー値を指定します。その QoS ポリシーの他のすべての値が成功する一致基準となります。たとえば、QoS クラスマップ コンフィギュレーション モードで **matchnotqos-group4** コマンドを発行すると、指定したクラスは、成功する一致基準として 4 を除くすべての QoS グループ値を受け入れます。

次のトラフィック クラスでは、IP 以外のすべてのプロトコルが成功する一致条件と見なされます。

```
class-map noip
  match not protocol ip
end
```

デフォルト トラフィック クラスの設定

トラフィック クラスで指定された一致条件を満たさないトラフィック（つまり、未分類のトラフィック）は、デフォルト トラフィック クラスに属するものとして扱われます。

デフォルト クラスを設定しない場合でも、パケットはそのクラスのメンバーとして扱われます。デフォルト クラスにはイネーブルになっている QoS 機能がないため、このクラスに属するパケットには QoS 機能がありません。そのようなパケットをテールドロップが管理する先入先出 (FIFO) キューに配置します。テールドロップは、すべてのトラフィックを等しく扱ってサービスクラス間での区別はせずに輻輳を回避する手段です。輻輳期間中はキューが一杯になります。出力キューが一杯でテールドロップがアクティブな場合、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。

次の例では、次の特性を持つデフォルト クラス（常に **class-default** という名前になります）のポリシー マップ (**policy1**) を設定します。ポリシーがクラス ポリシー **policy1** で定義されている他のクラスの一致基準を満たさないトラフィック用に 10 個のキューがあり、キューあたり 20 個のパケットを超えると、追加でキューに格納されるパケットを処理するためにテールドロップが実施されます。

次の例では、次の特性を持つデフォルト クラス（常に **class-default** という名前になる）のポリシー マップ (**policy1**) を設定します。ポリシーがトラフィック ポリシー **policy1** で定義されている他のクラスの一致基準を満たさないトラフィック用に 10 個のキューがあります。

```
policy-map policy1
  class class-default
    shape average 100m
```

「class-map match-any」コマンドと「class-map match-all」コマンドの違い

次に、複数の一致条件がある場合にパケットを評価する例を示します。この例で、**class-map match-any** コマンドと **class-map match-all** コマンドの違いを示します。トラフィック クラスのメンバーと見なされるためには、パケットが一致基準のすべて (**match-all**) を満たすか、または一致基準のいずれか1つ (**match-any**) を満たす必要があります。

次に、**class-map match-all** コマンドを使用してトラフィック クラスを設定する例を示します。

```
class-map match-all cisco1
  match qos-group 4
  match access-group 101
```

インターフェイス上に設定されているトラフィック クラス **cisco1** を持つルータにパケットが到着した場合、そのパケットが IP プロトコル、QoS グループ 4、およびアクセス グループ 101 に一致するかどうかの評価されます。これらの一致基準がすべて満たされると、パケットはトラフィック クラス **cisco1** のメンバーとして分類されます (論理 AND 演算子: IP プロトコル AND QoS グループ 4 AND アクセス グループ 101)。

```
class-map match-all vlan
  match vlan 1
  match vlan inner 1
```

次に、**class-map match-any** コマンドを使用する例を示します。パケットがトラフィック クラスのメンバーとして分類されるには、一致基準の1つのみを満たす必要があります (つまり、論理 OR 演算子: プロトコル IP OR QoS グループ 4 OR アクセス グループ 101)。

```
class-map match-any cisco2
  match protocol ip
  match qos-group 4
  match access-group 101
```

トラフィック クラス **cisco2** では、成功する一致が見つかるまで連続的に一致基準が評価されます。パケットは、まず IP プロトコルを一致基準として使用できるかどうかを判断するために評価されます。使用できる場合、パケットはトラフィック クラス **cisco2** に一致します。使用できない場合、QoS グループ 4 は一致基準として評価され、以降も同様に評価されます。パケットが指定したどの条件にも一致しない場合、パケットはデフォルトトラフィック クラス (**class default-class**) のメンバーとして分類されます。

一致基準としてのトラフィック クラス (ネストしたトラフィック クラス) の確立

matchclass-map コマンドを使用する理由は 2 つあります。1 つの理由はメンテナンスです。現在大規模なトラフィック クラスが存在している場合、トラフィック クラス一致基準を使用するほうが、同じトラフィック クラス設定を再入力するよりも簡単です。2 つめのより一般的な理由としては、1 つのポリシー内で **match-all** と **match-any** の特性を組み合わせるためです。これにより、1

つの一致基準評価命令（`match-any` と `match-all` のどちらか）を使用したトラフィック クラスを作成し、このトラフィック クラスを、異なるタイプの一致基準を使用するトラフィック クラスで一致条件として使用します。

考えられるシナリオは次のとおりです。A、B、C、および D が、すべて異なる一致基準であるとし、A、B、または C かつ D（A または B または（C かつ D））に一致するトラフィックをトラフィック クラスに属するものとして分類するとします。ネストしたトラフィック クラスがない場合、トラフィックがトラフィック クラスの一部であると見なされるためには、4 つの一致基準すべてに一致するか（A かつ B かつ C かつ D）、いずれかの一致基準に一致する必要があります（A または B または C または D）。「かつ」（`match-all`）文と「または」（`match-any`）文をトラフィック クラス内で組み合わせることはできないため、目的の設定を実現できません。

解決策：C と D に対して `match-all` を使用する 1 つのトラフィック クラスを作成し（これを条件 E と呼びます）、A、B、E を使用して新しい `match-any` トラフィック クラスを作成します。新しいトラフィック クラスの評価順序は正しくなります（A または B または E、つまり A または B または（C かつ D））。

例：メンテナンスのためにネストされたトラフィック クラス

次の例で、トラフィック クラス `class1` の特性は、トラフィック クラス `class2` の特性とほぼ同じですが、トラフィック クラス `class1` では、一致条件として宛先アドレスが追加されています。トラフィック クラス `class1` をゼロから設定する代わりに、`matchclass-map class2` コマンドを入力できます。このコマンドを使用すると、トラフィック クラス `class2` のすべての特性をトラフィック クラス `class1` に取り込み、トラフィック クラスを再設定することなく、新しい宛先アドレスの一致基準を追加できます。

```
Router(config)# class-map match-any class2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# class-map match-all class1
Router(config-cmap)# match class-map class2
Router(config-cmap)# match destination-address mac 0000.0000.0000
Router(config-cmap)# exit
```

例：match-any 特性と match-all 特性を 1 つのトラフィック クラスで組み合わせるためのネストしたトラフィック クラス

1 つのトラフィック クラスで `match-any` 特性と `match-all` 特性を使用するための唯一の方法は、`matchclass-map` コマンドを使用することです。1 つのクラスに `match-any` と `match-all` の特性を組み合わせるには、一致基準に `match-all` 命令を使用して設定されたクラスを使用するトラフィック クラスを作成する `match-any` 命令を使用します（`matchclass-map` コマンドを使用）。

次に、2 つのトラフィック クラスの特性を組み合わせる例を示します。1 つは `match-any` 特性を使用し、1 つは `match-all` 特性を使用しています。これを、`matchclass-map` コマンドで 1 つのトラフィック クラスとして設定します。その結果、パケットがトラフィック クラス `class4` のメンバーと見なされるためには、次の 3 つの一致基準のいずれかを満たしている必要があります。IP プロトコルかつ QoS グループ 4、宛先 MAC アドレス 00.00.00.00.00.00、またはアクセス グループ 2。

この例で、トラフィック クラス class4 だけがトラフィック ポリシー policy1 と共に使用されています。

```
Router(config)# class-map match-all class3
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit
Router(config)# class-map match-any class4
Router(config-cmap)# match class-map class3
Router(config-cmap)# match destination-address mac 00.00.00.00.00.00
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class4
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# end
```

例：QoS ポリシーとしてのトラフィック ポリシー（階層型トラフィック ポリシー）

QoS ポリシー マップ クラス コンフィギュレーション モードで **service-policy** コマンドを使用すると、トラフィック ポリシーを QoS ポリシー内に含めることができます。トラフィック ポリシーを含むトラフィック ポリシーは、階層型トラフィック ポリシーと呼ばれます。

階層型トラフィック ポリシーには、1つの子ポリシーと1つの親ポリシーが含まれています。子ポリシーは、以前に定義したトラフィック ポリシーであり、**service-policy** コマンドを使用して新しいトラフィック ポリシーに関連付けられます。既存のトラフィック ポリシーを使用する新しいトラフィック ポリシーが親ポリシーです。ここに示す例では、トラフィック ポリシー child が子ポリシーであり、トラフィック ポリシー parent が親ポリシーです。

階層型トラフィック ポリシーはサブインターフェイスに追加できます。階層型トラフィック ポリシーを使用すると、1つのトラフィック ポリシー（1つの子ポリシー1つの親のポリシーを持つ）を使用して、サブインターフェイスのシェーピングと優先順位付けを行うことができます。

```
Router(config)# policy-map child
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# service-policy child
```

shape コマンドで使用する値は、サービスプロバイダーから通知された認定情報レート（CIR）値からプロビジョニングします。

port-channel インターフェイスの入力 MQC の設定方法

トラフィック フローを区別し、対応する「qos-group」機能を設定するために port-channel インターフェイス上で入力 MQC を設定するには、次に説明する手順に従います。



制約事項

- ポリシング、シェーピング、WRED、キューイングなどの QoS アクションはサポートされていません。
- ケーブルの物理インターフェイスで入力 MQC を設定することはできません。

トラフィック クラスの作成

トラフィック クラスを定義するには、**class-map** コマンドを使用します。トラフィック クラスに含まれる 3 つの主な要素は、名前、一連の **match** コマンド、そしてトラフィック クラスに複数の **match** コマンドが存在する場合に **match** コマンドを評価する方法です。

match コマンドは、パケット分類のためのさまざまな基準を指定するために使用します。パケットが検査されて、**match** コマンドで指定された基準に一致するかどうか判別されます。指定された基準にパケットが一致する場合、そのパケットはクラスのメンバーと見なされ、トラフィックポリシーで設定された QoS 仕様に従って転送されます。一致基準を満たさないパケットは、デフォルトのトラフィック クラスのメンバーとして分類されます。

トラフィック クラスを作成して一致基準を定義するには、次の手順に従います。

```
configure terminal
class-map class
match type
```

ポリシー マップの作成

トラフィック クラスの作成後にトラフィック ポリシーを設定すると、これらのクラス内の選択したトラフィックに特定のアクションを適用するためのマーキング機能を設定できます。

トラフィック ポリシーを作成するには、**policy-map** コマンドを使用します。トラフィック ポリシーの目的は、ユーザ指定のトラフィック クラスに分類されたトラフィックに関連付ける QoS 機能を設定することです。



(注)

パケットは、トラフィック ポリシー内のいずれかのトラフィック クラスだけに一致します。パケットがトラフィック ポリシー内の複数のトラフィック クラスに一致する場合、ポリシーで定義されている最初のトラフィック クラスが使用されます。

トラフィック ポリシーを定義するには、次の手順を実行します。

```
configure terminal
policy-map policy
class class
```

ポリシー マップでの QoS アクションの定義

ポリシー マップでクラス モードからアクション コマンドを追加できます。

set アクション

set コマンドを使用してトラフィックにマークを付けると、転送パスにある他のネットワーク デバイスが、トラフィック フローに適用する適切なサービス クラスを迅速に判断できるようになります。

set アクションを定義するには、次の手順に従います。

```
configure terminal
policy-map policy
class class
set option
```

集約 port-channel インターフェイスの設定

port-channel インターフェイスを設定するには、次の手順に従います。

```
configure terminal
platform qos port-channel-aggregate port_channel_number
interface port-channel port_channel_number
ip address ip mask
interface name
channel-group number
```

トラフィック ポリシーのインターフェイスへの適用

policy-map コマンドを使用してトラフィック ポリシーを定義した後、**service-policy** コマンドをインターフェイス コンフィギュレーション モードで使用して、1つ以上のインターフェイスに適用できます。同じトラフィック ポリシーを複数のインターフェイスに割り当てることができますが、各インターフェイスには、入力方向と出力方向に対して、それぞれトラフィック ポリシーを1つだけ割り当てることができます。

トラフィック ポリシーをインターフェイスに適用するには、次の手順をすべて行います。

```
configure terminal
interface port-channel port_channel_number
service-policy input policy
```

例：port-channel インターフェイスの入力 MQC の設定

以下に、port-channel インターフェイスの入力 MQC を設定する例を示します。

```
Router# configure terminal
Router(config)# class-map class1
```

```

Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# set dscp af11
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# platform qos port-channel-aggregate 2 Router(config)# interface port-channel
2
Router(config-if)# ip address 192.168.0.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# interface tenGigabitEthernet 4/1/1
Router(config-if)# no ip address
Router(config-if)# no shut
Router(config-if)# channel-group 2
Router(config-if)# interface port-channel 2
Router(config-if)# service-policy input policy1
Device(config-if)# end

```

その他の参考資料

関連資料

関連項目	マニュアルタイトル
Cisco IOS コマンド	『 Cisco IOS Master Commands List, All Releases 』
QoS コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『 <i>Cisco IOS Quality of Service Solutions Command Reference</i> 』
パケット分類	「Classifying Network Traffic」モジュール
フレームリレーフラグメンテーション (FRF) PVC	「FRF .20 Support」モジュール
選択的パケット廃棄	「IPv6 Selective Packet Discard」モジュール
スケーリングとパフォーマンスの情報	『 Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide 』の「Broadband Scalability and Performance」モジュール

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

モジュラ Quality of Service コマンドラインインターフェイスの QoS に関する機能情報

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 を使用すると、ソフトウェア イメージがサポートする特定のソフトウェア リリース、フィーチャセット、またはプラットフォームを確認できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> からアクセスします。Cisco.com のアカウントは必要ありません。



(注) 次の表は、特定のソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースのみを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 4: モジュラ Quality of Service コマンドラインインターフェイスの QoS に関する機能情報

機能名	リリース	機能情報
モジュラ Quality of Service コマンドラインインターフェイスの QoS	Cisco IOS XE Everest 16.6.1	この機能が Cisco cBR シリーズ コンバージドブロードバンド ルータ上の Cisco IOS XE Everest 16.6.1 に統合されました。
port-channel インターフェイスでのサービス ポリシー	Cisco IOS XE Everest 16.6.1	この機能が Cisco IOS XE Everest 16.6.1 上の Cisco cBR シリーズ コンバージドブロードバンド ルータ に統合されました。