QoS：輻輳回避設定ガイド

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。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system.All rights reserved.Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries.To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks.Third-party trademarks mentioned are the property of their respective owners.The use of the word partner does not imply a partnership relationship between Cisco and any other company.(1110R)

© 2017 Cisco Systems, Inc. All rights reserved.

目 次

最初にお読みください 1

輻輳回避の概要 3

機能情報の確認 3

重み付けランダム早期検出 3

ランダム早期検出について 4

機能のしくみ 4

パケットの廃棄確率 4

TCP によるトラフィック損失の対処方法 5

ルータと TCP との相互作用の方法 6

WRED について 7

WRED を使用する理由 7

機能のしくみ 7

平均キュー サイズ 9

IPv6 QoS：MQC WRED ベース ドロップ 11

機能情報の確認 11

IPv6 QoS：MQC WRED ベース ドロップに関する情報 11

QoS for IPv6 の実装方針 11

IPv6 トラフィックの輻輳回避 12

その他の参考資料 13

IPv6 QoS：MQC WRED ベース ドロップに関する機能情報 13

重み付けランダム早期検出の設定 15

機能情報の確認 15

重み付けランダム早期検出について 15

WRED の設定方法 16

WRED のイネーブル化 16

WRED パラメータの変更 17

WRED のモニタリング 17

WRED の設定例 18
WRED の設定例 18
パラメータ設定 WRED の例 19
重み付けランダム早期検出の設定に関する機能情報 19
バイトベースの重み付けランダム早期検出 21
機能情報の確認 21
バイトベースの重み付けランダム早期検出の制約事項 21
バイトベースの重み付けランダム早期検出に関する情報 22
WRED 機能の変更 22
キュー制限と WRED しきい値の変更 22
バイトベースの重み付けランダム早期検出を設定する方法 22
バイトベースの WRED の設定 22
キュー項目数と WRED しきい値の設定 24
キュー項目数と WRED しきい値の単位モードの変更 29
バイトベースの WRED の設定の確認 32
バイトベースの重み付けランダム早期検出の設定例 33
バイトベースの WRED の設定例 33
その他の参考資料 34
バイトベースの重み付けランダム早期検出の機能情報 35
WRED 明示的輻輳通知 37
機能情報の確認 37
WRED 明示的輻輳通知の前提条件 37
WRED 明示的輻輳通知に関する情報 38
WRED - 明示的輻輳通知機能の概要 38
WRED の仕組み 38
ECN による WRED 機能の拡張 38
ECN がイネーブルのときパケットはどのように処理されるか 39
WRED 明示的輻輳通知の利点 40
WRED 明示的輻輳通知の設定方法 40
明示的輻輳通知の設定 40
明示的輻輳通知の設定の確認 42
WRED 明示的輻輳通知の設定例 43

ECN の有効化の例 43
ECN 設定の確認例 44
その他の参考資料 45
WRED 明示的輻輳通知の機能情報 46
WRED およびキュー制限に関する QoS 時間ベースしきい値 49
機能情報の確認 49
WRED およびキュー制限に関する QoS 時間ベースしきい値の前提条件 50
WRED およびキュー制限に関する QoS 時間ベースしきい値の制約事項 50
WRED およびキュー制限に関する QoS 時間ベースしきい値に関する情報 50
WRED およびキュー制限に関する QoS 時間ベースしきい値の利点 50
WRED を使用したしきい値の設定 51
queue-limit コマンドを使用したしきい値の設定 51
ミリ秒キーワードを使用した random-detect コマンド 51
しきい値の測定単位の混合 52
WRED およびキュー制限に関する QoS 時間ベースしきい値を設定する方法 53
WRED の有効化と WRED を使用したしきい値の指定 53
queue-limit コマンドを使用したしきい値の指定 55
WRED 設定の QoS 時間ベースしきい値におけるインターフェイスへのポリシーマップのアタッチ 57
WRED およびキュー制限に関する QoS 時間ベースしきい値設定の確認 59
トラブルシューティングのヒント 60
WRED およびキュー制限に関する QoS 時間ベースしきい値の設定例 60
WRED を使用したしきい値の設定例 60
queue-limit コマンドを使用したしきい値の設定例 61
設定の確認例 61
WRED しきい値設定サンプル出力の例 62
queue-limit コマンドしきい値設定サンプル出力例 63
その他の参考資料 64
WRED およびキュー制限に関する QoS 時間ベースしきい値の機能情報 65
DiffServ 準拠の WRED 67
機能情報の確認 67
DiffServ 準拠の WRED に関する情報 68

WRED の差別化サービス 68
DiffServ 準拠の WRED に関する使用上のガイドライン 68
DiffServ 準拠の WRED を設定する方法 68
DiffServ 準拠の WRED の設定 68
DiffServ 準拠の WRED の設定例 72
例：DiffServ 準拠の WRED 72
その他の参考資料 72
DiffServ 準拠の WRED の機能情報 74
ダイヤラ インターフェイス上のシェーピング 75
機能情報の確認 75
ダイヤラ インターフェイス上のシェーピングに関する制約事項 76
ダイヤラ インターフェイス上のシェーピングに関する情報 76
ダイヤラ インターフェイス上の PPP セッションでの QoS 76
QoS のダイヤラ インターフェイス トポロジ 77
ダイヤラ インターフェイス上のシェーピングを設定する方法 77
ダイヤラ インターフェイスの出力キューイング ポリシーの設定 77
ダイヤラ インターフェイスの PPPoEoA 用の QoS の設定 82
ダイヤラ インターフェイスの PPPoE 用の QoS の設定 86
ダイヤラ インターフェイスの PPPoA 用の QoS の設定 89
ダイヤラ インターフェイス上の複数セッション用の QoS の設定 92
ダイヤラ インターフェイスへの CoS 値の適用 95
ダイヤラ インターフェイス上のシェーピングの設定例 98
例：ダイヤラ インターフェイス用の出力キューイング ポリシーの設定 98
例：ダイヤラ インターフェイスの PPPoEoA 用の QoS の設定 98
例：ダイヤラ インターフェイス上の PPPoE 用の QoS の設定 99
例：ダイヤラ インターフェイス上の PPPoA 用の QoS の設定 99
例：ダイヤラ インターフェイス上の複数セッション用の QoS の設定 99
例：ダイヤラ インターフェイスへの CoS 値の適用 100
ダイヤラ インターフェイス上のシェーピングに関する追加の参考資料 100
ダイヤラ インターフェイス上のシェーピングに関する機能情報 101

第 1 章

最初にお読みください

Cisco IOS XE 16 に関する重要な情報

Cisco IOS XE リリース 3.7.0E（Catalyst スイッチング用）および Cisco IOS XE リリース 3.17S（アクセスおよびエッジルーティング用）の 2 つのリリースは、コンバージドリリースの 1 つのバージョン - Cisco IOS XE 16 - に統合されました。この 1 つのリリースでスイッチングおよびルーティング ポートフォリオのアクセスおよびエッジ製品を幅広くカバーしています。

（注） 技術構成ガイドの機能情報の表に、機能の導入時期を記載しています。他のプラットフォームがその機能をサポートした時期については、記載があるものも、ないものもあります。特定の機能が、使用しているプラットフォームでサポートされているかどうかを判断するには、製品のランディング ページに掲載された技術構成ガイドを参照してください。技術構成ガイドが製品のランディング ページに表示されると、その機能が該当のプラットフォームでサポートされているかどうかが示されます。

第 2 章

輻輳回避の概要

一般的なネットワークのボトルネックでの輻輳を予測および回避するために、輻輳回避技術によってネットワーク トラフィック負荷を監視します。輻輳の回避は、パケットのドロップにより行われます。より一般的に使用されている輻輳回避メカニズムはランダム早期検出（RED）で、これは高速中継ネットワークに最適です。Cisco IOS XE ソフトウェアには、RED アルゴリズムの機能を IP Precedence 機能と組み合わせた、Weighted RED（WRED）と呼ばれる RED の実装が含まれています。WRED は設定されると、ルータがパケットをドロップするタイミングを制御します。

- 機能情報の確認，3 ページ
- 重み付けランダム早期検出，3 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。

重み付けランダム早期検出

WRED は、起こりうるグローバリゼーションの問題の回避に役立ちます。グローバル同期は、輻輳の波の頂点として発生し、後には伝送リンクの一部だけが活用されている状態の谷が続きます。TCP ホストのグローバル同期は、たとえば、パケットがいっせいにドロップされるために発生することがあります。グローバル同期は、複数の TCP ホストが伝送レートを、パケットのドロップに応じて低下させる場合に現れます。輻輳が減少すると、再び伝送レートを増加させます。

ランダム早期検出について

REDメカニズムは、対処方式ではなく応答方式でネットワーク輻輳に対応することを目的にSally FloydおよびVan Jacobsonによって1990年代初めに提唱されました。REDメカニズムでは、損失の影響を受けやすく、一部のトラフィックがドロップされると一時的に速度が低下するデータ転送の実装環境でほとんどのトラフィックが動作しているということが基本的な前提になります。TCPは、たとえ堅牢でも、そのトラフィック伝送を低下させることによってトラフィックのドロップに適切に対応し、REDのトラフィックドロップ動作を輻輳回避シグナリングメカニズムとして効果的に動作させることができます。

TCPは、最も頻繁に使用されるネットワーク トランスポートを構成します。TCPの遍在的存在を前提として、REDは広範囲の有効な輻輳回避機能を提供します。

TCPなどの堅牢なトランスポートがパーベイシブである場合にREDの有用性を検討する際は、かなりの割合のトラフィックがパケット損失に対して堅牢でない場合にREDを採用することの著しいマイナス要素を検討することも重要です。Novell NetWareもAppleTalkもパケット損失に対して十分に堅牢ではないので、それらにREDを使用すべきではありません。

機能のしくみ

DiffServ準拠のWRED機能で、パケットの廃棄確率を計算する場合にWREDでDSCP値を使用できるようにします。DSCP値は、IPタイプ オブ サービス（ToS）バイトの最初の6ビットです。

この機能で、**random-detect dscp**および**dscp**の2つの新しいコマンドが追加されます。また、*dscp-based*および*prec-based*の2つの新しい引数を、既存の**random-detect**（インターフェイス）コマンドおよび**random-detect-group**コマンドの2つのWRED関連コマンドに追加します。

*dscp-based*引数は、パケットの廃棄確率を計算する場合にWREDでパケットのDSCP値を使用できるようにします。*prec-based*引数は、パケットの廃棄確率を計算する場合にWREDでパケットのIP Precedence値を使用できるようにします。

これらの引数は、任意です（このコマンドを使用するために、これらの引数のいずれかを必ずしも使用しなくてもかまいません）が、二者択一です。つまり、*dscp-based*引数を使用すると、同じコマンドを使用して*prec-based*引数を使用することはできません。

WREDでDSCP値を使用できるようにしたら、新しい**random-detect dscp**コマンドを使用して、DSCP値の最低および最高パケットしきい値を変更できます。

これらの引数を使用する3つのシナリオが提供されています。

パケットの廃棄確率

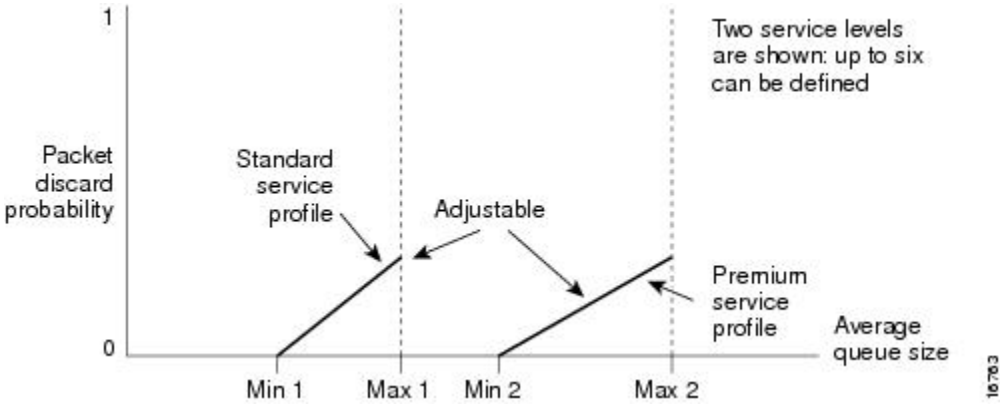
パケットの廃棄確率は、最低しきい値、最高しきい値、およびマーク確率値に基づきます。

平均のキュー深度が最低しきい値より大きい場合、REDはパケットのドロップを開始します。パケットのドロップ率は、平均キューサイズの増加に対して、平均キューサイズが最高しきい値に達するまで直線的に増加します。

マーク確率値は、平均キュー深度が最高しきい値の場合のドロップしたパケットの比です。たとえば、分母が 512 の場合、512 パケットごとに 1 つのパケットが、平均キューが最大しきい値にあるときにドロップします。

平均キュー サイズが最大しきい値を超えた場合、すべてのパケットがドロップします。次の図では、パケットの廃棄確率について要約しています。

図 *1*：*RED* パケットの廃棄確率



最低しきい値は、リンクを最大限活用するのに十分な高さに設定してください。最低しきい値が低すぎると、パケットが不必要にドロップされ、伝送リンクが最大限に使用されません。

最大しきい値と最低しきい値の差は、TCP ホストのグローバル同期を避けるのに十分な大きさにしてください（TCP ホストのグローバル同期は、複数の TCP ホストの伝送レートが低下して発生することがあります）。最大および最低しきい値の差が小さすぎると、一度に多くのパケットがドロップし、グローバル同期につながることがあります。

TCP によるトラフィック損失の対処方法

（注） セクションTCP によるトラフィック損失の対処方法，（5 ページ）およびルータと TCP との相互作用の方法，（6 ページ）に、WRED を使用するためや、RED の一般的な機能を備えるためには読まなくてもよい詳細情報が記載されています。輻輳の発生に応じてグローバル同期の問題が起こる理由と、RED のこの問題への対処方法を知りたい場合は、これらのセクションを参照してください。

TCP トラフィックの受け手（受信者と呼びます）がデータ セグメントを受信すると、データ セグメントが順番に受信されたことを示す、受信者が予想している番号に対し、セグメントの 4 オクテット（32 ビット）のシーケンス番号がチェックされます。番号が一致すると、受信者は、ターゲット アプリケーションに保持するデータすべてを提供し、次の番号を順番に反映するためにシーケンス番号を更新し、最後に、送信者に確認応答（ACK）パケットを送信するか、または短い遅延の後に ACK を送信者に送信するようにスケジューリングします。ACK は送信者に、受信者が、到着したすべてのデータ セグメントを受信したが、新しいシーケンス番号でマークされたセグメントは含まれていないことを通知します。

通常、受信者は、受信する代替データ セグメントに応じて ACK の送信を試行します。受信者は多数のアプリケーションという理由のため ACK を送信し、受信者が短い遅延を待機する場合は、それに送信者に対する通常応答の返信確認応答を効率的に含めることができます。しかし、受信者がデータ セグメントをばらばらの順番で受信した場合、ACK をただちに返信し、送信者に、損失したデータ セグメントを再送するよう指示します。

送信者が ACK を受信すると、未処理のデータがあるかどうかが調査されます。未処理のデータがない場合、送信者は、ACK がキープアライブであると決定します。つまり、回線をアクティブのままにし、何もしません。未処理のデータがある場合、送信者は、受信者がデータの一部を受け取ったか、まったく受け取っていないかが ACK に示されているかを確認します。ACK に、送信データの一部を受領していることが示されている場合、送信者は、さらにデータを送信できる新しいクレジットが付与されているかどうかを確認します。ACK に、送信したデータがまったく受領されていないことが示されている場合で、未処理のデータがある場合は、送信者はこの ACK を、再送信が繰り返されている ACK であると解釈します。この状態は、一部のデータがばらばらの順番で受信され、受信者が最初の ACK を強制的に送信し、2 番めのデータ セグメントが、ばらばらの順番で受信され、受信者が別の ACK を強制的に送信したことを示しています。多くの場合、データ セグメントのうち 1 つがドロップしたため、受信者は 2 つのセグメントをばらばらの順番で受け取ります。

TCP の送信者がドロップされたデータ セグメントを検出すると、送信者はセグメントを再送信します。その後、転送レートを、ドロップが検出される前の半分に調整します。これが、TCP のバックオフまたは低速化動作です。この動作は輻輳に適切に対応しますが、複数の TCP セッションが同じルータで同時に伝送され、すべての TCP 送信者がパケットの伝送速度を同時に低下させると、問題が発生する場合があります。

ルータと TCP との相互作用の方法

（注） セクションTCP によるトラフィック損失の対処方法, （5 ページ）およびルータと TCP との相互作用の方法, （6 ページ）に、WRED を使用するためや、RED の一般的な機能を備えるためには読まなくてもよい詳細情報が記載されています。輻輳の発生に応じてグローバル同期の問題が起こる理由と、RED のこの問題への対処方法を知りたい場合は、これらのセクションを参照してください。

ルータがどのように TCP と相互作用するかを、例で確認します。この例では、平均して、ルータは、MAE-EAST または FIX-WEST のインターフェイスの 1 つおき、10 個おき、100 個または 200 個おきのメッセージごとに 1 つの特定の TCP ストリームからトラフィックを受信します。ルータは、複数の TCP セッションを同時に処理できます。ネットワーク フローが加法型なため、トラフィックが送信キュー制限（TQL）を超えた場合はすべて、制限を大きく超える可能性が非常に高くなります。ただし、超過トラフィックの深度が一時的で、トラフィック フローが統合するポイントやエッジ ルータを除き、トラフィックが過度に深いままにはならない可能性も大いにあります。

ルータが TQL を超えるトラフィックをすべてドロップした場合、多くの TCP セッションで同時にスロー スタートが開始されます。したがって、トラフィックは一時的に極端に低速になり、す

べてのフローが再度スロースタートで処理されます。この動作により、グローバル同期の状態が作り出されます。

しかし、均等化キューイングまたはプライオリティキューイング（PQ）などのキューイング機能が使用されている場合と同様に、ルータがトラフィックをドロップしない場合、データはメインメモリに格納されることが多く、ルータの性能が著しく低下します。

一度に 1 つの TCP セッションの速度低下を指示することで、RED は上記の問題を解決し、トラフィックの使用が山と谷として現れずに、帯域幅を完全に使用することができます。

WRED について

WRED は、RED アルゴリズムの機能を IP Precedence 機能と統合して、優先順位の高いパケットの優先的なトラフィック処理を実現します。WRED では、インターフェイスで輻輳が発生し始めると、優先順位が低いトラフィックを選択的に廃棄し、異なるサービス クラスに対して差別化したパフォーマンス特性を提供できます。

重み付けされていない RED の動作が行われるように、ドロップの決定を行う際に IP precedence を無視するよう WRED を設定できます。

リソース予約プロトコル（RSVP）機能を使用するよう設定されたインターフェイスでは、WRED は、RSVP フロー以外の他のフローから、ドロップするパケットを選択します。また、IP Precedence で、どのパケットをドロップするかを制御します。優先順位の低いトラフィックは、ドロップ率が高く、抑制されやすくなります。

WRED は、キューイング戦略のような他の輻輳回避技術とは異なります。発生した輻輳をコントロールするのではなく、輻輳を予測し回避しようとするためです。

WRED を使用する理由

WRED は、輻輳の可能性を早期検出し、複数のトラフィック クラスを実現します。また、グローバル同期に対して保護されます。そのため、WRED は輻輳が発生すると予測されるいずれの出力インターフェイスでも有用です。

ただし、WRED は通常は、ネットワークのエッジではなくネットワークのコア ルータで使用されます。エッジ ルータは、ネットワークに送信されてくるパケットに IP Precedence を割り当てます WRED は、これらの優先順位を使用して、異なる種類のトラフィックの処理方法を定義します。

WRED は個別のしきい値と重み付けを、それぞれの IP Precedence に使用し、異なるトラフィックの種類でのパケットのドロップに対し、異なるサービス品質を実現できます。標準的なトラフィックは、輻輳時には、優先度の高いトラフィックよりも頻繁にドロップされる可能性があります。

また WRED は、RSVP を認識し、統合サービスの負荷制御型 QoS サービスも提供します。

機能のしくみ

高輻輳期間の前にランダムにパケットをドロップすることにより、WRED はパケットの送信元に、その伝送レートを低下させるよう指示します。そのパケットの送信元が TCP を使用している場

合、すべてのパケットがその宛先に到達する、つまり輻輳がなくなったことが示されるまで伝送レートを下げます。

WRED では一般的に、IP Precedence に基づいてパケットを選択的にドロップします。より高い IP Precedence のパケットは、低い IP Precedence のパケットよりもドロップしにくくなります。したがって、パケットの優先順位が高いほど、パケットが送信される確率が高くなります。

WRED は、出力インターフェイスが輻輳の兆候を示し始めると、選択的にパケットをドロップします。キューがいっぱいになるのを待たずに、早期にパケットの一部をドロップすることで、WRED は一度に大量のパケットがドロップしないようにし、グローバル同期の確率を最低限にします。このように、WRED は、いつでも伝送ラインをいっぱいに使用することができるようにします。

さらに、WREDは統計的に、小さなユーザよりも大きなユーザから、より大きなパケットをドロップします。したがって、トラフィックの大部分を生成しているトラフィック発信元は、生成しているトラフィックが少しのトラフィック発信元よりも速度が低下しやすくなります。

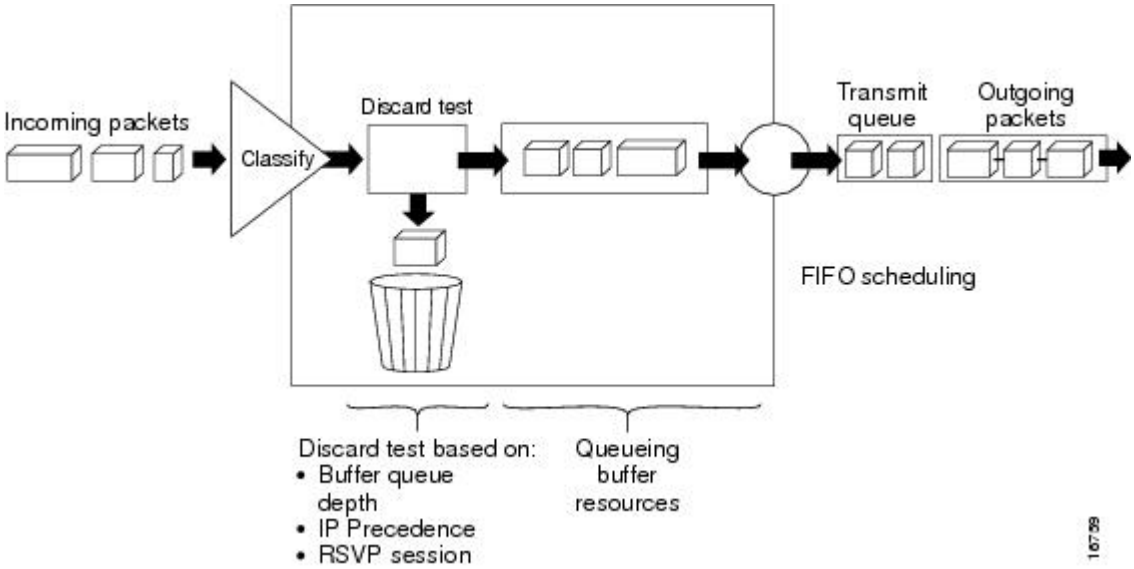
WREDは、グローバリゼーションの問題の回避に役立ちます。グローバル同期は、複数の TCP ホストが伝送レートを、パケットのドロップに応じて低下させる場合に現れます。輻輳が減少すると、再び伝送レートを増加させます。

WRED は、ほとんどのトラフィックが TCP/IP トラフィックである場合にだけ有用です。TCP では、破棄されたパケットは輻輳を示しているため、パケットの発信元は伝送レートを低くします。他のプロトコルでは、パケットの発信元は応答しないか、または破棄されたパケットを同じレートで再送信します。このため、パケットを破棄しても輻輳は削減されません。

WRED は非 IP トラフィックを優先度 0 の最低優先順位として扱います。したがって、非 IP トラフィックは通常、IP トラフィックよりもドロップしやすくなります。

次の図は、WRED がどのように機能するかを示しています。

図 *2*：重み付けランダム早期検出



平均キュー サイズ

平均キュー サイズは、キューの前回の平均と現在のサイズを基にしています。式は次のようになります。

```
average = (old_average * (1-1/2^n)) + (current_queue_size * 1/2^n)
```

ここで、*n* は指数加重係数で、ユーザが設定可能な値です。

n を高い値にすると、前回の平均キュー サイズがより重要視されます。係数を大きくすると、キューの長さの最大値と最小値が滑らかになります。平均キュー サイズは、素早い変化はしにくく、サイズの急激な増減を回避します。WRED 処理で、パケットのドロップの開始が遅くなりますが、実際のキュー サイズが最低しきい値を下回った時点でも、パケットのドロップが続く場合があります。平均がゆっくり変化するため、トラフィックの一時的なバーストに対応します。

（注） *n* の値が高すぎる場合、WRED は輻輳に反応しません。パケットは、WRED が無効のときのように送信またはドロップします。

n の値が低い場合、平均キュー サイズは現在のキュー サイズ付近を追跡します。結果、平均はトラフィック レベルの変化とともに上下します。この場合、WRED 処理は、長いキューに素早く応答します。キューが最低しきい値を下回ると、パケットのドロップ処理が停止します。

n の値が低すぎると、WRED は一時的なトラフィック バーストに過剰に反応し、不必要にトラフィックをドロップします。

第 3 章

IPv6 QoS：MQC WRED ベース ドロップ

WRED は、CBWFQ の制限を超える可能性のあるパケットに対して RED ベースのドロップ ポリシーを実装します。

- 機能情報の確認，11 ページ
- IPv6 QoS：MQC WRED ベース ドロップに関する情報，11 ページ
- その他の参考資料，13 ページ
- IPv6 QoS：MQC WRED ベース ドロップに関する機能情報，13 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。

IPv6 QoS：MQC WRED ベース ドロップに関する情報

QoS for IPv6 の実装方針

IPv6 パケットは、IPv4 パケットとは別のパスで転送されます。IPv6 環境でサポートされている QoS 機能には、パケット分類、キューイング、トラフィック シェーピング、重み付けランダム早期検出（WRED）、クラスベース パケット マーキング、および IPv6 パケットのポリシングが含

まれます。これらの機能は、IPv6 のプロセス スイッチング パスとシスコ エクスプレス フォワーディング スイッチング パスのどちらでも使用できます。

IPv6 環境で使用可能な QoS 機能はすべて、モジュラ QoS コマンドライン インターフェイス（MQC）から管理します。MQC を使用すると、トラフィック クラスを定義し、トラフィック ポリシー（ポリシーマップ）を作成および設定してから、それらのトラフィック ポリシーをインターフェイスに適用することができます。

IPv6 が稼働しているネットワークに QoS を実装するには、IPv4 だけが稼働しているネットワークに QoS を実装するときの手順に従ってください。高度なレベルで QoS を実装するための基本手順は、次のとおりです。

- QoS を必要とするネットワーク内のアプリケーションを特定します。
- どの QoS 機能が適切であるかを判断するために、アプリケーションの特性を理解します。
- 変更と転送がリンク層ヘッダー サイズに及ぼす影響を理解するために、ネットワーク トポロジについて理解します。
- ネットワークに規定する基準に基づいて、クラスを作成します。具体的には、同じネットワークで IPv6 トラフィックだけでなく IPv4 トラフィックも伝送されている場合、IPv6 トラフィックと IPv4 トラフィックを同様に処理するか、それとも別個の方法で処理し、それぞれに応じた一致基準を指定するかを決定します。両者を同様に処理する場合は、**matchprecedence**、**matchdscp**、**setprecedence**、**setdscp** などの match 文を使用します。両者を別の方法で処理する場合は、match-all クラス マップ内に **matchprotocolip** や **matchprotocolipv6** などの一致基準を追加します。
- 各クラスにマーキングするためのポリシーを作成します。
- QoS 機能を適用する際は、エッジからコアに向かって作業します。
- トラフィックを処理するためのポリシーを構築します。
- ポリシーを適用します。

IPv6 トラフィックの輻輳回避

WRED は、クラスベース均等化キューイング（CBWFQ）の制限を超える可能性のあるパケットに対して RED ベースのドロップ ポリシーを実装します。WRED では、（DSCP または precedence の値を使用する）クラスベース キューイングとフローベース キューイングをサポートしています。

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 Command List, All Releases』
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★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

IPv6 QoS：MQC WRED ベース ドロップに関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *1*：*IPv6 QoS*：*MQC WRED* ベース ドロップに関する機能情報

機能名	リリース	機能情報
IPv6 QoS：MQC WRED ベース ドロップ	Cisco IOS XE Release 2.1	WRED は、CBWFQ の制限を超える可能性のあるパケットに対して RED ベースのドロップ ポリシーを実装します。

第 4 章

重み付けランダム早期検出の設定

このモジュールでは、ルータで重み付けランダム早期検出（WRED）を設定するためのタスクについて説明します。

- 機能情報の確認，15 ページ
- 重み付けランダム早期検出について，15 ページ
- WRED の設定方法，16 ページ
- WRED の設定例，18 ページ
- 重み付けランダム早期検出の設定に関する機能情報，19 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。

重み付けランダム早期検出について

ランダム早期検出（RED）は、TCP の輻輳制御メカニズムを利用する輻輳回避メカニズムです。高輻輳期間の前にランダムにパケットをドロップすることにより、RED はパケットの送信元に、その伝送レートを低下させるよう指示します。WRED では、IP Precedence に基づいてパケットを選択的にドロップします。エッジルータは、ネットワークに送信されてくるパケットに IP Precedence を割り当てます（WRED は、輻輳が予想される任意の出力インターフェイスに有効です。ただし、

WREDは通常は、エッジではなくネットワークのコアルータで使用されます。）WREDは、これらの優先順位を使用して、異なるタイプのトラフィックの処理方法を定義します。

パケットが着信すると、次のイベントが発生します。

1. 平均キュー サイズが計算されます。
2. 平均が最低キューしきい値未満の場合は、到着したパケットはキューイングされます。
3. 平均値がそのトラフィック タイプの最小キューしきい値と、インターフェイスの最大しきい値の間の場合、そのトラフィック タイプのパケット ドロップ確率に応じて、パケットはドロップされるか、キューに保存されます。
4. 平均キュー サイズが最大しきい値を超える場合、パケットはドロップします。

（注） WRED は TCP/IP などの適応型トラフィックに有効です。TCP では、破棄されたパケットは輻輳を示しているため、パケットの発信元は伝送レートを低くします。他のプロトコルでは、パケットの発信元は応答しないか、または破棄されたパケットを同じレートで再送信します。このため、パケットを破棄しても輻輳は削減されません。WRED は非 IP トラフィックを優先度 0 の最低優先順位として扱います。そのため、非 IP トラフィックは IP トラフィックよりもドロップされる可能性が高くなります。

random-detect インターフェイス コンフィギュレーション コマンドで WRED をイネーブルにすると、パラメータはデフォルト値に設定されます。加重係数は 9 です。すべての precedence について、マーク確率値は10で、最大しきい値は、インターフェイスの出力バッファリング容量と送信速度に基づいています。

デフォルトの最小しきい値は precedence によって変わります。IP precedence 0 の最小しきい値は、最大しきい値の半分に相当します。残りの precedence の値は、均等な間隔で、最大しきい値の半分と最大しきい値の間に含まれます。

（注） デフォルトの WRED パラメータ値は、最高の使用可能データに基づきます。値を変更することでアプリケーションに利点があると判断した場合を除き、パラメータのデフォルト値を変更しないことを推奨します。

WRED の設定方法

WRED のイネーブル化

コマンド	目的
`Router(config-if)#` **random-detect**	WRED をイネーブルにします。

WRED パラメータの変更

コマンド	目的
`Router(config-if)#` **random-detect exponential-weighting-constant** *exponent*	平均キュー長の計算に使用する加重係数を設定します。
`Router(config-if)#` **random-detect precedence** *precedence min-threshold max-threshold mark-prob-denominator*	特定の IP precedence を持つパケットのパラメータを設定します。IP precedence 0 の最小しきい値は、インターフェイスの最大しきい値の半分に相当します。precedence ごとにこのコマンドを繰り返します。WRED ではなく RED を設定するには、各 precedence に同じパラメータを使用します。

WRED のモニタリング

コマンド	目的
`Router#` **show queue** *interface-type interface-number*	キュー内のパケットのヘッダー情報を表示します。
`Router#` **show queueing interface** *interface-number* [**vc** [[*vpi*/] *vci*]]	インターフェイス上の特定の仮想回線（VC）の WRED 統計情報を表示します。
`Router#` **show queueing random-detect**	WRED のキューイング設定を表示します。
`Router#` **show interfaces** [*type slot* | *port-adapter* | *port*]	インターフェイス上の WRED 設定を表示します。

WRED の設定例

WRED の設定例

次に、デフォルトのパラメータ値で WRED をイネーブルにする例を示します。

```
interface Serial5/0
 description to qos1-75a
 ip address 200.200.14.250 255.255.255.252
 random-detect
```

show interfaces コマンドの出力を使用して、設定を確認します。「Queueing strategy」レポートには、「ランダム早期検（RED）」がリストされます。

```
Router# show interfaces serial 5/0
Serial5/0 is up, line protocol is up
  Hardware is M4T
  Description: to qos1-75a
  Internet address is 200.200.14.250/30
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 237/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  Last input 00:00:15, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:05:08
  Input queue: 0/75/0 (size/max/drops); Total output drops: 1036
  Queueing strategy: random early detection(RED)
  5 minutes input rate 0 bits/sec, 2 packets/sec
  5 minutes output rate 119000 bits/sec, 126 packets/sec
     594 packets input, 37115 bytes, 0 no buffer
     Received 5 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     37525 packets output, 4428684 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

show queue コマンドの出力を使用して、インターフェイス キューの現在の内容を表示します。ドロップが発生した後にすべての IP precedence のパケットを配置するキューは 1 つだけです。5 パケットのうち 3 パケットのみを示すために、出力を省略しました。

```
Router# show queue serial 5/0

Output queue for Serial5/0 is 5/0
Packet 1, linktype: ip, length: 118, flags: 0x288
  source: 190.1.3.4, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 128 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765
Packet 2, linktype: ip, length: 118, flags: 0x288
  source: 190.1.3.5, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 160 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765
Packet 3, linktype: ip, length: 118, flags: 0x280
  source: 190.1.3.6, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 192 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765
```

show queueing コマンドの出力を使用して、各 precedence の現在の設定を表示します。また、デフォルトの最小しきい値は、最大しきい値の半分と最大しきい値の間で均等の間隔です。しきい値はパケット カウントを単位として指定します。

```
Router# show queueing
Current random-detect configuration:
     Serial5/0
       Queueing strategy:random early detection (WRED)
       Exp-weight-constant:9 (1/512)
       Mean queue depth:28

       Class   Random       Tail    Minimum    Maximum     Mark
                 drop       drop  threshold  threshold  probability
         0        330          0         20         40     1/10
         1        267          0         22         40     1/10
         2        217          0         24         40     1/10
         3        156          0         26         40     1/10
         4         61          0         28         40     1/10
         5          6          0         31         40     1/10
         6          0          0         33         40     1/10
         7          0          0         35         40     1/10
        rsvp        0          0         37         40     1/10
```

パラメータ設定 WRED の例

次に、インターフェイスで WRED をイネーブルにし、異なる IP precedence にパラメータを指定する例を示します。

```
interface Hssi0/0/0
 description 45Mbps to R1
 ip address 10.200.14.250 255.255.255.252
 random-detect
 random-detect precedence 0 32 256 100
 random-detect precedence 1 64 256 100
 random-detect precedence 2 96 256 100
 random-detect precedence 3 120 256 100
 random-detect precedence 4 140 256 100
 random-detect precedence 5 170 256 100
 random-detect precedence 6 290 256 100
 random-detect precedence 7 210 256 100
 random-detect precedence rsvp 230 256 100
```

重み付けランダム早期検出の設定に関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *2*：重み付けランダム早期検出の設定に関する機能情報

機能名	リリース	機能情報
クラスベース重み付け均等化キューイング（CBWFQ）および重み付けランダム早期検出（WRED）	Cisco IOS XE Release 2.1	この機能は、Cisco ASR 1000 シリーズ ルータに追加されました。 （注） CBWFQについては、「Configuring Weighted Fair Queueing」モジュールを参照してください。
ランダム早期検出（RED）	Cisco IOS XE Release 2.1	この機能は、Cisco ASR 1000 シリーズ ルータに追加されました。
重み付けランダム早期検出	Cisco IOS XE Release 2.1	この機能は、Cisco ASR 1000 シリーズ ルータに追加されました。
WRED	Cisco IOS XE Release 2.1	この機能は、Cisco ASR 1000 シリーズ ルータに追加されました。

第 5 章

バイトベースの重み付けランダム早期検出

このモジュールでは、バイトベースの重み付けランダム早期検出（WRED）をイネーブルにして、バイトベースのキュー制限と WRED しきい値を設定する方法について説明します。

- 機能情報の確認，21 ページ
- バイトベースの重み付けランダム早期検出の制約事項，21 ページ
- バイトベースの重み付けランダム早期検出に関する情報，22 ページ
- バイトベースの重み付けランダム早期検出を設定する方法，22 ページ
- バイトベースの重み付けランダム早期検出の設定例，33 ページ
- その他の参考資料，34 ページ
- バイトベースの重み付けランダム早期検出の機能情報，35 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。

バイトベースの重み付けランダム早期検出の制約事項

- WRED は、ほとんどのトラフィックが TCP/IP トラフィックである場合にだけ有用です。TCP では、破棄されたパケットは輻輳を示しているため、パケットの発信元は伝送レートを低く

します。他のプロトコルでは、パケットの発信元は応答しないか、または破棄されたパケットを同じレートで再送信します。このため、パケットを破棄しても輻輳は削減されません。

- queue-limit がミリ秒単位またはパケット単位で設定されているクラスでは、バイトベースの WRED を設定できません。

バイトベースの重み付けランダム早期検出に関する情報

WRED 機能の変更

この機能により、WRED の機能が拡張されます。以前のリリースでは、パケット数に基づいて WRED のアクションを指定していました。バイトベースの WRED では、バイト数に基づいて WRED のアクションを指定できます。

キュー制限と WRED しきい値の変更

Cisco IOS XE Release 2.4 の Cisco ASR 1000 シリーズのアグリゲーション サービス ルータでは、キュー制限と WRED しきい値の両方の設定単位としてバイト単位の追加がサポートされています。したがって、このリリースでは、いくつかの制限がありますが、パケットベースとバイトベースの両方の制限を設定できます。

バイトベースの重み付けランダム早期検出を設定する方法

バイトベースの WRED の設定

手順の概要

1. **enable**
2. **configureterminal**
3. **class-map***class-map-name*
4. **matchipprecedence**ip-precedence-value
5. exit
6. **policy-map***policy-name*
7. **class***class-name*
8. **random-detect**
9. **random-detectprecedence***precedencemin-threshold***bytes***max-threshold***bytes***mark-prob-denominator*

手順の詳細

	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	**configureterminal** 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***class-map-name* 例： `Router(config)# class-map c1`	トラフィック クラスのユーザ定義名を指定します。
ステップ 4	**matchipprecedence**ip-precedence-value 例： `Router(config-cmap)# match ip precedence 1`	一致基準として使用する IP precedence 値を最大 8 個指定します。
ステップ 5	exit 例： `Router(config-cmap)# exit`	class-map 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	**policy-map***policy-name* 例： `Router(config)# policy-map p1`	設定するトラフィック ポリシーの名前を指定します。
ステップ 7	**class***class-name* 例： `Router(config-pmap)# class c1`	定義済みのトラフィック クラスの名前を指定します。この名前は、トラフィックをトラフィック ポリ

	コマンドまたはアクション	目的
		シーに分類するために使用する **class-map** コマンドを使用して設定されています。
ステップ 8	**random-detect** 例： `Router(config-pmap-c)# random-detect`	WRED をイネーブルにします。
ステップ 9	**random-detectprecedence***precedencemin-threshold***bytes***max-threshold***bytes***mark-prob-denominator* 例： 例： `Router(config-pmap-c)# random-detect precedence 1 2000 bytes 3000 bytes 200`	特定の IP precedence を持つバイトのパラメータを設定します。

キュー項目数と WRED しきい値の設定

はじめる前に

キュー項目数と WRED しきい値を設定する場合は、次の条件を満たしている必要があります。

- バイトベース モードを設定する場合は、キュー制限を設定してから WRED しきい値を設定し、サービス ポリシーを適用する必要があります。
- 拡張 QoS ポリシー集約設定でキュー項目数と WRED しきい値を設定する場合は、サブインターフェイス ポリシー マップのデフォルト クラスと、メイン インターフェイス ポリシー マップの任意のクラスでだけ制限がサポートされます。

（注） キュー項目数と WRED しきい値を設定する場合は、次の制約事項を考慮してください。

- トラフィック クラスのキューイング機能を設定する前に、キュー制限単位を設定しないでください。
- キュー制限を設定しない場合、デフォルト モードはパケット単位になります。
- WRED しきい値を設定する場合は、次の制約事項が適用されます。
  - WRED しきい値では、キュー制限と同じ単位を使用する必要があります。たとえば、キュー制限がパケット単位の場合、WRED しきい値もパケット単位にする必要があります。
  - キュー制限をバイト単位に設定しない場合、デフォルト モードはパケット単位になるため、WRED しきい値もパケット単位に設定する必要があります。
  - キュー制限のサイズは、WRED しきい値よりも大きくする必要があります。
- サービス ポリシーを適用した後は、キュー制限または WRED しきい値のいずれの単位モードも、動的に変更できなくなります。

>

手順の概要

1. **enable**
2. **configureterminal**
3. **policy-map***policy-map-name*
4. **class***class-name*
5. *qos-queueing-feature*
6. **queue-limit***queue-limit-size* [**bytes** | **packets**]
7. **random-detect** [**dscp-based** | **prec-based**]
8. 次のいずれかを実行します。
   - **random-detectdscp** dscp-value {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} [*max-probability-denominator*]
   -
   -
   - **random-detectprecedence** precedence {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} max-probability-denominator

手順の詳細

	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	**configureterminal** 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***policy-map-name* 例： `Router(config)# policy-map main-interface`	設定または変更するトラフィック ポリシーの名前を指定し、policy-map コンフィギュレーションモードを開始します。
ステップ 4	**class***class-name* 例： `Router(config-pmap)# class AF1`	トラフィック クラスの名前を指定し、policy-map class コンフィギュレーション モードを開始します。
ステップ 5	*qos-queueing-feature* 例： `Router(config-pmap-c)# bandwidth remaining ratio 90`	QoS コンフィギュレーション コマンドを入力します。現在サポートされているキューイング機能には、**bandwidth**、**priority**、**shape** などがあります。 （注） このステップでは、複数の QoS キューイング コマンドを入力できます。ただし、キュー制限と WRED しきい値の依存関係のため、キュー制限を設定した後に WRED を設定する必要があります。
ステップ 6	**queue-limit***queue-limit-size* [**bytes** | **packets**] 例： `Router(config-pmap-c)# queue-limit 547500 bytes`	キューがこのクラス用に保持できるバイトまたはパケットの最大数を指定します（1 ～ 8192000）。
ステップ 7	**random-detect** [**dscp-based** | **prec-based**] 例： `Router(config-pmap-c)# random-detect dscp-based`	DSCP ベース モードまたは precedence ベース モードのいずれかで WRED をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 8	次のいずれかを実行します。 • **random-detectdscp** dscp-value {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} [*max-probability-denominator*] • • • **random-detectprecedence** precedence {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} max-probability-denominator 例： `Router(config-pmap-c)# random-detect precedence 8 750000 bytes 750000 bytes`	特定の DSCP 値または IP precedence の WRED パラメータを設定します。 （注） **queue-limit** 単位もパケットである場合（デフォルト）、**bytes** キーワードを指定せずに *min-thresholdmax-threshold* 引数を使用して、パケットベースのしきい値を設定します。または、**queue-limit** 単位がバイト単位に設定されている場合は、**bytes** キーワードを指定してこれらの引数を使用します。

例

次の例は、一部の制約事項を示すために正しい設定と無効な設定の両方を使用しています。

正しい設定

次の例は正しい設定方法を示しており、トラフィック クラスに対して **bandwidthremainingratio** キューイング機能を設定した後に、キュー制限をバイト モードで設定しています。

```
class AF1
 bandwidth remaining ratio 90
  queue-limit 750000 bytes
```

無効な設定

次の例は無効な設定方法を示しており、トラフィック クラスに対して **bandwidthremainingratio** キューイング機能を設定する前に、キュー制限をバイト モードで設定しています。

```
class AF1
 queue-limit 750000 bytes
  bandwidth remaining ratio 90
```

正しい設定

次の例は正しい設定方法を示しており、トラフィック クラスに対して **bandwidthremainingratio** キューイング機能を設定し、キュー制限をバイトモードで設定した後に、WRED しきい値を互換性のあるバイト モードで設定しています。

```
class AF1
 bandwidth remaining ratio 90
  queue-limit 750000 bytes
   random-detect dscp-based
    random-detect dscp 8 750000 bytes 750000 bytes
```

無効な設定

次の例は無効な設定方法を示しており、キュー制限を設定せずに（このため、デフォルトでパケットベースのキュー項目数になります）WRED しきい値をバイト単位で設定しています。したがって、WRED しきい値もパケット単位にする必要があります。

```
class AF1
 bandwidth remaining ratio 90
 random-detect dscp-based
 random-detect dscp 8 750000 bytes 750000 bytes
```

キュー項目数と WRED しきい値の単位モードの変更

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber*
4. **noservice-policyoutput***policy-map-name*
5. **exit**
6. **policy-map***policy-map-name*
7. **class***class-name*
8. **queue-limit***queue-limit-size* [**bytes** | **packets**]
9. 次のいずれかを実行します。
   - **norandom-detectdscp***dscp-value* {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} [*max-probability-denominator*]
   -
   -
   - **norandom-detectprecedence***precedence* {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} *max-probability-denominator*
10. 次のいずれかを実行します。
    - **random-detectdscp***dscp-value* {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} [*max-probability-denominator*]
    -
    -
    - **random-detectprecedence***precedence* {*min-thresholdmax-threshold*| *min-threshold***bytes***max-threshold***bytes**} *max-probability-denominator*

手順の詳細

	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）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 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	**interface***typenumber* 例： `Router(config)# policy-map main-interface`	サービス ポリシーを削除する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 4	**noservice-policyoutput***policy-map-name* 例： `Router(config-if)# no service-policy output main-interface-policy`	指定したインターフェイスに適用されているサービス ポリシーを削除します。
ステップ 5	**exit** 例： `Router(config-if)# exit`	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 モードに戻ります。
ステップ 6	**policy-map***policy-map-name* 例： `Router(config)# policy-map main-interface-policy`	変更するトラフィック ポリシーの名前を指定し、policy-map コンフィギュレーション モードを開始します。
ステップ 7	**class***class-name* 例： `Router(config-pmap)# class AF1`	トラフィック クラスの名前を指定し、policy-map class 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	**queue-limit***queue-limit-size* [**bytes** | **packets**] 例： `Router(config-pmap-c)# queue-limit 5000 packets`	キューがこのクラス用に保持できるバイトまたはパケットの最大数を指定します（1 ～ 8192000）。
ステップ 9	次のいずれかを実行します。 • **norandom-detectdscp***dscp-value* {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} [*max-probability-denominator*]	前に設定した特定の DSCP 値または IP precedence の WRED パラメータを削除します。

	コマンドまたはアクション	目的
	• • • **norandom-detectprecedence***precedence* {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} *max-probability-denominator* 例： `Router(config-pmap-c)# no random-detect dscp 8 750000 bytes 750000 bytes`	
ステップ 10	次のいずれかを実行します。 • **random-detectdscp***dscp-value* {*min-thresholdmax-threshold* | *min-threshold***bytes***max-threshold***bytes**} [*max-probability-denominator*] • • • **random-detectprecedence***precedence* {*min-thresholdmax-threshold*| *min-threshold***bytes***max-threshold***bytes**} *max-probability-denominator* 例： `Router(config-pmap-c)# random-detect dscp 8 4000 4000`	特定の DSCP 値または IP precedence の WRED パラメータを設定します。 （注） queue-limit 単位もパケットである場合（デフォルト）、**bytes** キーワードを指定せずに *min-thresholdmax-threshold* 引数を使用して、パケットベースのしきい値を設定します。または、queue-limit 単位がバイト単位に設定されている場合は、**bytes** キーワードを指定してこれらの引数を使用します。

例

次の例は、サービスポリシーをインターフェイスに適用した後に、キュー項目数と WRED しきい値をパケットベースの値に変更する方法を示しています。

```
interface GigabitEthernet1/2/0
no service-policy output main-interface-policy
end
policy-map main-interface-policy
class AF1
 queue-limit 5000 packets
 no random-detect dscp 8 750000 bytes 750000 bytes
 random-detect dscp 8 4000 4000
```

バイトベースの WRED の設定の確認

手順の概要

1. **showpolicy-map**
2. **show policy-map interface**

手順の詳細

ステップ 1 **showpolicy-map**
show policy-map

例：

```
Router# show policy-map
Policy Map pol1
  Class class c1
Bandwidth 10 (%)
exponential weight 9
        class   min-threshold(bytes)   max-threshold(bytes)  mark-probability
        ----------------------------------------------------------------------

        0       -                       -                     1/10
        1       20000                   30000                 1/10
        2       -                       -                     1/10
        3       -                       -                     1/10
        4       -                       -                     1/10
        5       -                       -                     1/10
        6       -                       -                     1/10
        7       -                       -                     1/10
        rsvp    -                       -                     1/10
```

ステップ 2 **show policy-map interface**

例：

```
Router# show policy-map interface
serial3/1
Service-policy output: pol
Class-map: silver (match-all)
366 packets, 87840 bytes
30 second offered rate 15000 bps, drop rate 300 bps
Match: ip precedence 1
Queueing
Output Queue: Conversation 266
Bandwidth 10 (%)
(pkts matched/bytes matched) 363/87120
depth/total drops/no-buffer drops) 147/38/0
exponential weight: 9
mean queue depth: 25920
class     Transmitted       Random drop      Tail drop     Minimum Maximum Mark
          pkts/bytes        pkts/bytes       pkts/bytes    thresh  thresh  prob
                                                           (bytes)  (bytes)
0             0/0               0/0              0/0      20000    40000  1/10
1           328/78720         38/9120            0/0      22000    40000  1/10
2             0/0               0/0              0/0      24000    40000  1/10
```

```
3            0/0              0/0              0/0      26000    40000  1/10
4            0/0              0/0              0/0      28000    40000  1/10
```

バイトベースの重み付けランダム早期検出の設定例

バイトベースの WRED の設定例

次の例は、wred-policy というサービス ポリシーを示しています。このポリシーは、prec2 というクラスとデフォルトクラスに対してバイトベースのWREDを設定します。その後、このポリシーはファスト イーサネット インターフェイス 0/0/1 に適用されます。

```
policy wred-policy
 class prec2
   bandwidth 1000
   random-detect
   random-detect precedence 2 100 bytes 200 bytes 10
class class-default
   random-detect
   random-detect precedence 4 150 bytes 300 bytes 15
   random-detect precedence 6 200 bytes 400 bytes 5
interface fastethernet0/0/1
  service-policy output wred-policy
```

次の例は、イーサネット インターフェイス 0/0/1 に適用されているサービス ポリシーに関するバイトベースの WRED の結果を示しています。

```
Router# show policy-map interface
 Ethernet0/0/1
Service-policy output: wred-policy (1177)
Class-map: prec2 (match-all) (1178/10)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip precedence 2  (1179)
Queueing
queue limit 62500 bytes
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts queued/bytes queued) 0/0
bandwidth 1000 (kbps)
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0 bytes
class     Transmitted       Random drop      Tail drop Minimum        Maximum     Mark
          pkts/bytes        pkts/bytes       pkts/bytes thresh         thresh     prob
                                                         bytes          bytes
0             0/0               0/0              0/0     15625          31250     1/10
1             0/0               0/0              0/0     17578          31250     1/10
2             0/0               0/0              0/0       100            200     1/10
3             0/0               0/0              0/0     21484          31250     1/10
4             0/0               0/0              0/0     23437          31250     1/10
5             0/0               0/0              0/0     25390          31250     1/10
6             0/0               0/0              0/0     27343          31250     1/10
7             0/0               0/0              0/0     29296          31250     1/10
Class-map: class-default (match-any) (1182/0)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any  (1183)
0 packets, 0 bytes
5 minute rate 0 bps
```

```
queue limit 562500 bytes
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts queued/bytes queued) 0/0
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0 bytes
class     Transmitted       Random drop      Tail drop Minimum        Maximum     Mark
          pkts/bytes        pkts/bytes       pkts/bytes thresh         thresh      prob
                                                         bytes          bytes
0             0/0               0/0              0/0     140625         281250     1/10
1             0/0               0/0              0/0     158203         281250     1/10
2             0/0               0/0              0/0     175781         281250     1/10
3             0/0               0/0              0/0     193359         281250     1/10
4             0/0               0/0              0/0        150            300     1/15
5             0/0               0/0              0/0     228515         281250     1/10
6             0/0               0/0              0/0        200            400     1/5
7             0/0               0/0              0/0     263671         281250     1/10
```

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	『*Cisco IOS Quality of Service Solutions Command Reference*』
Modular QoS CLI	「Modular Quality of Service Command-Line Interface」モジュール

標準

規格	タイトル
新しい規格または変更された規格はサポートされていません。また、既存の規格に対するサポートに変更はありません。	--

MIB

MIB	MIB のリンク
新しい MIB または変更された MIB はサポートされていません。また、既存の MIB に対するサポートに変更はありません。	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
新しい RFC または変更された RFC はサポートされていません。また、既存の RFC に対するサポートに変更はありません。	--

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★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

バイトベースの重み付けランダム早期検出の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *3*：バイトベースの重み付けランダム早期検出の機能情報

機能名	リリース	機能情報
バイトベースの重み付けランダム早期検出	Cisco IOS XE Release 2.4	バイトベースの重み付けランダム早期検出機能によって、WREDの機能が拡張されます。以前のリリースでは、パケット数に基づいて WRED のアクションを指定していました。バイトベースの WRED では、バイト数に基づいて WRED のアクションを指定できます。 この機能は、Cisco ASR 1000 シリーズ ルータに追加されました。 **random-detect**、**random-detectprecedence**、**showpolicy-map**、**showpolicy-mapinterface** の各コマンドが導入または変更されました。

第 6 章

WRED 明示的輻輳通知

- 機能情報の確認，37 ページ
- WRED 明示的輻輳通知の前提条件，37 ページ
- WRED 明示的輻輳通知に関する情報，38 ページ
- WRED 明示的輻輳通知の設定方法，40 ページ
- WRED 明示的輻輳通知の設定例，43 ページ
- その他の参考資料，45 ページ
- WRED 明示的輻輳通知の機能情報，46 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。

WRED 明示的輻輳通知の前提条件

ECN は、モジュラ QoS コマンドライン インターフェイス（MQC）を使用して設定する必要があります。MQC の詳細については、「Applying QoS Features Using the MQC」モジュールを参照してください。

WRED 明示的輻輳通知に関する情報

WRED - 明示的輻輳通知機能の概要

現在、Transmission Control Protocol（TCP）の輻輳管理と回避アルゴリズムは、ベストエフォート型サービス モデルを使用してデータを送信するネットワークの輻輳を適切に示すのはパケット損失であるという概念に基づいています。ネットワークがベストエフォート型サービス モデルを使用する場合、ネットワークは信頼性、遅延限界、スループットを保証せずに、可能であればデータを配信します。ただし、これらのアルゴリズムとベストエフォート型サービス モデルは、遅延やパケット損失の影響を受ける用途には適していません（例：Telnet、Webブラウジング、音声およびビデオ データなどの双方向トラフィック転送）。Weighted Random Early Detection（WRED）および明示的輻輳通知（ECN）は、この問題の解決に役立ちます。

RFC 3168「*IP* への明示的輻輳通知（*ECN*）の追加（*The Addition of Explicit Congestion Notification (ECN) to IP*）」は、インターネット インフラにアクティブ キュー管理（例：WRED）を追加すると、ルータは輻輳の兆候としてパケット損失に限定されなくなることを示しています。

WRED の仕組み

WRED は、輻輳の早期検出を可能にし、複数のトラフィック クラスを処理する手段を提供します。WRED では、ルータで輻輳が発生し始めると、よりプライオリティが低いトラフィックを選択的に廃棄し、異なるサービス クラスに対して差別化したパフォーマンス特性を提供できます。また、グローバル同期に対して保護されます。グローバル同期は輻輳の波が頂点に達すると発生し、その後は転送リンクが容量いっぱいまで使用されない時間が続きます。そのため、WRED は輻輳の発生が予測される出力インターフェイスやルータで役立ちます。

WREDはネットワークのコア ルータで実行されます。エッジルータは、パケットがネットワークに入ると IP 優先順位をパケットに割り当てます。WRED では、コア ルータはこれらの優先順位を使用して、異なる種類のトラフィックの処理方法を決定します。WRED は個別のしきい値と重み付けを、それぞれの IP 優先順位に使用し、ネットワークは異なるトラフィックの種類でのパケットのドロップに対し、異なるサービス品質を実現できます。標準的なトラフィックは、輻輳時には、優先度の高いトラフィックよりも頻繁にドロップされる可能性があります。

WRED の詳細情報については、『Congestion Avoidance Overview（輻輳回避の概要）』モジュールを参照してください。

ECN による WRED 機能の拡張

WRED は、輻輳を示す、特定のしきい値を超える平均キュー長に基づいてパケットをドロップします。ECN はWREDの拡張で、平均キュー長が特定のしきい値を超えた場合にパケットをドロップせずにマーキングします。WRED 明示的輻輳通知機能を設定すると、ルータとエンド ホストは、このマーキングをネットワークの輻輳によってパケットの送信速度が低下していることを示す警告として使用します。

RFC 3168『*TheAdditionofExplicitCongestionNotification(ECN)toIP*』に記述されているように、ECN を実装するには、ECN 専用フィールドで ECN 対応転送（ECT）ビットと CE（Congestion Experienced）ビットの 2 つのビットが IP ヘッダに含まれている必要があります。ECT ビットと CE ビットを使用して、00 から 11 の 4 つの ECN フィールドの組み合わせを作成できます。最初の数字は ECT ビットで、2 番目の数字は CE ビットです。次の表は、ECN フィールドの ECT と CE ビットの組み合わせの設定一覧と、その組み合わせの意味を示しています。

表 *4*：*ECN* ビットの設定

ECT ビット	CE ビット	組み合わせが示す内容
0	0	ECN 対応ではない
0	1	転送プロトコルのエンドポイントが ECN 対応
1	0	転送プロトコルのエンドポイントが ECN 対応
1	1	Congestion experienced

ECN のフィールドの組み合わせ 00 は、パケットが ECN を使用していないことを示します。

ECN のフィールドの組み合わせ 01 と 10（それぞれ着信側 ECT（1）と ECT（0））は、データの送信側によって設定され、転送プロトコルのエンドポイントが ECN 対応であることを示します。ルータは、これらの 2 つのフィールドの組み合わせを同様に取り扱います。データの送信元は、これらの 2 つの組み合わせの 1 つまたは両方を使用できます。これらの 2 つのフィールドの組み合わせ、および組み合わせて使用する意味の詳細については、RFC 3168『*TheAdditionofExplicitCongestionNotification(ECN)toIP*』を参照してください。

ECN フィールドの組み合わせ 11 は、エンド ポイントに対する輻輳を示します。ルータの満杯のキューに到着するパケットはドロップされます。

ECN がイネーブルのときパケットはどのように処理されるか

- キュー内のパケット数が最小しきい値未満の場合、パケットが送信されます。これは ECN がイネーブルになっているかどうかに関係なく実行されます。この処理は、ネットワーク上で WRED だけが使用されている場合、パケットが受けるのと同一の処理です。
- キュー内のパケット数が最小しきい値と最大しきい値の間の場合、次の 3 つのシナリオのいずれかになる可能性があります。
  - パケットの ECN のフィールドにエンドポイントが ECN 対応であることが示されている（つまり、ECT ビットが 1 および CE ビットが 0 に設定されているか、または ECT ビットが 0 および CE ビットが 1 に設定されている）場合、および WRED アルゴリズムによってパケットが廃棄確率に基づいてドロップされると判断される場合には、パケットの ECT ビットと CE ビットが 1 に変更され、パケットが送信されます。これは、ECN

がイネーブルであり、パケットがドロップされる代わりにマークされているために発生します。

- パケットの ECN のフィールドによって、どちらのエンドポイントも ECN 対応ではないことが示されている（つまり、ECT ビットが 0 に設定され、CE ビットが 0 に設定されている）場合、パケットは、WRED 廃棄確率に基づいてドロップされる可能性があります。これは、ルータ上で ECN を設定せずに WRED がイネーブル化されている場合に、パケットが受けるのと同一の処理です。
- パケットの ECN のフィールドに、ネットワークで輻輳が発生していることが示されている（つまり、ECT ビットと CE ビットの両方が 1 に設定されている）場合、パケットが送信されます。これ以上のマーキングは必要ありません。

- キュー内のパケット数が最小しきい値を上回っている場合、パケットはドロップ確率に基づいてドロップされます。これは、ルータ上で ECN を設定せずに WRED がイネーブル化されている場合に、パケットが受けるのと同一の処理です。

WRED 明示的輻輳通知の利点

改善された輻輳回避の方法

この機能では、伝送用のパケットをキューからドロップするのではなく、ネットワークで後からマーキングできるようにして、輻輳回避の方法が改善されます。伝送用のパケットを後でマーキングすると、遅延またはパケット損失の影響を受けるアプリケーションに適応して、スループットとアプリケーション パフォーマンスが向上します。

拡張されたキュー管理

現在、ドロップされたパケットは、キューが満杯で、ネットワークに輻輳が発生していることを示します。ネットワークに輻輳が発生した場合、この機能を使用すると、ネットワークはCE ビットを使用してパケットの IP ヘッダーにマーク付けすることができます。このマーキングによって、順々に、適切な輻輳回避機能がトリガーされ、ネットワークはデータ キューをよりよく管理できます。この機能を使用すると、キューがオーバーフローしてパケットがドロップされる前に、ECN 対応ルータとエンド ホストが輻輳に対応でき、キューの管理が向上します。

WRED 明示的輻輳通知の設定方法

明示的輻輳通知の設定

ECN を設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **policy-map***policy-map-name*
4. **class**{*class-name*| **class-default**}
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*
6. **random-detect**
7. **random-detectecn**
8. **end**

手順の詳細

	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	**configureterminal** 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***policy-map-name* 例： `Router(config)# policy-map policy1`	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。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 マップの名前を入力します。
ステップ 4	**class**{*class-name*| **class-default**} 例： `Router(config-pmap)# class class-default`	作成または変更するポリシーのクラス名を指定するか、ポリシーを指定する前にデフォルト クラス（一般に class-default クラスといいます）を指定します。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。 • クラス名を入力するか、**class-default** キーワードを入力します。

	コマンドまたはアクション	目的

ステップ 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*
	
	例：
	`Router(config-pmap-c)#`
	`bandwidth percent 35`
		ポリシーマップに属するクラスに割り当てる帯域幅（kbps 単位またはパーセンテージ）を指定または変更します。
		• 帯域幅を KB/秒で入力するか、または帯域幅の割合を入力します。

ステップ 6	**random-detect**
	
	例：
	`Router(config-pmap-c)#`
	`random-detect`
		WRED または分散型 WRED（dWRED）をイネーブルにします。

ステップ 7	**random-detectecn**
	
	例：
	`Router(config-pmap-c)#`
	`random-detect ecn`
		ECN をイネーブルにします。

ステップ 8	**end**
	
	例：
	`Router(config-pmap-c)#`
	`end`
		（任意）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ます。

明示的輻輳通知の設定の確認

ECN 設定を確認するには、次の手順を実行します。

手順の概要

1. **enable**
2. **showpolicy-map**
3. **showpolicy-mapinterface**
4. **end**

手順の詳細

	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	**showpolicy-map** 例： `Router# show policy-map`	ECN がイネーブルであると、指定のポリシー マップに ECN マーキング情報が表示されます。
ステップ 3	**showpolicy-mapinterface** 例： `Router# show policy-map interface`	ECN がイネーブルであると、指定のインターフェイスに ECN マーキング情報が表示されます。

ステップ 4	**end**
	
	例：
	`Router#`
	`end`
		（任意）特権 EXEC モードを終了します。

WRED 明示的輻輳通知の設定例

ECN の有効化の例

次に、pol1 という名前のポリシー マップで ECN をイネーブルにする例を示します。

```
Router(config)# policy-map pol1
 Router(config-pmap)# class class-default
 Router(config-pmap-c)# bandwidth per 70
 Router(config-pmap-c)# random-detect
 Router(config-pmap-c)# random-detect ecn
```

ECN 設定の確認例

次は、**show policy-map** コマンドのサンプル出力です。出力に含まれる「explicit congestion notification」という語（およびECNマーキング情報）はECNがイネーブルにされていることを示します。

```
Router# show policy-map
   Policy Map pol1
     Class class-default
       Weighted Fair Queueing
             Bandwidth 70 (%)
             exponential weight 9
             explicit congestion notification
             class    min-threshold    max-threshold    mark-probability
             ----------------------------------------------------------
             ----------------------------------------------------------

             0        -                -                1/10
             1        -                -                1/10
             2        -                -                1/10
             3        -                -                1/10
             4        -                -                1/10
             5        -                -                1/10
             6        -                -                1/10
             7        -                -                1/10
             rsvp     -                -                1/10
```

次は、**show policy-map interface** コマンドのサンプル出力です。出力に含まれる「explicit congestion notification」という語はECNがイネーブルにされていることを示します。

```
Router# show policy-map interface
 Serial4/1
 Serial4/1
  Service-policy output:policy_ecn
        Class-map:prec1 (match-all)
          1000 packets, 125000 bytes
          30 second offered rate 14000 bps, drop rate 5000 bps
          Match:ip precedence 1
          Weighted Fair Queueing
            Output Queue:Conversation 42
            Bandwidth 20 (%)
            Bandwidth 100 (kbps)
            (pkts matched/bytes matched) 989/123625
        (depth/total drops/no-buffer drops) 0/455/0
             exponential weight:9
             explicit congestion notification
             mean queue depth:0
     class   Transmitted  Random drop  Tail drop   Minimum     Maximum     Mark
             pkts/bytes   pkts/bytes    pkts/bytes threshold   threshold   probability
       0       0/0          0/0          0/0          20          40        1/10
       1     545/68125      0/0          0/0          22          40        1/10
       2       0/0          0/0          0/0          24          40        1/10
       3       0/0          0/0          0/0          26          40        1/10
       4       0/0          0/0          0/0          28          40        1/10
       5       0/0          0/0          0/0          30          40        1/10
       6       0/0          0/0          0/0          32          40        1/10
       7       0/0          0/0          0/0          34          40        1/10
     rsvp      0/0          0/0          0/0          36          40        1/10
     class   ECN Mark
            pkts/bytes
       0     0/0
       1    43/5375
       2     0/0
       3     0/0
       4     0/0
       5     0/0
       6     0/0
```

```
  7      0/0
rsvp     0/0
```

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	『*Cisco IOS Quality of Service Solutions Command Reference*』
MQC	「Applying QoS Features Using the MQC」モジュール
輻輳回避の概念	「Congestion Avoidance Overview」モジュール

標準

規格	タイトル
新しい規格または変更された規格はサポートされていません。また、既存の規格に対するサポートに変更はありません。	--

MIB

MIB	MIB のリンク
新しい MIB または変更された MIB はサポートされていません。また、既存の MIB に対するサポートに変更はありません。	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 2309	『*InternetPerformanceRecommendation*』
RFC 2884	『*PerformanceEvaluationofExplicitCongestionNotification (ECN)inIPNetworks*』
RFC 3168	『*TheAdditionofExplicitCongestionNotification(ECN)toIP*』

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★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

WRED 明示的輻輳通知の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *5*：*WRED* 明示的輻輳通知の機能情報

機能名	ソフトウェア リリース	機能の設定情報
WRED 明示的輻輳通知	Cisco IOS XE Release 2.1	現在、Transmission Control Protocol（TCP）の輻輳管理と回避アルゴリズムは、ベストエフォート型サービス モデルを使用してデータを送信するネットワークの輻輳を適切に示すのはパケット損失であるという概念に基づいています。ネットワークがベストエフォート型サービス モデルを使用する場合、ネットワークは信頼性、遅延限界、スループットを保証せずに、可能であればデータを配信します。ただし、これらのアルゴリズムとベストエフォート型サービス モデルは、遅延やパケット損失の影響を受ける用途には適していません（例：Telnet、Web ブラウジング、音声およびビデオ データなどの双方向トラフィック転送）。Weighted Random Early Detection（WRED）および明示的輻輳通知（ECN）は、この問題の解決に役立ちます。 次のコマンドが導入または変更されました。 **random-detectecn**、**showpolicy-map**、**showpolicy-mapinterface**

第 7 章

WRED およびキュー制限に関する QoS 時間ベースしきい値

WRED およびキュー制限に関する QoS 時間ベースしきい値の機能を使用すると、重み付けランダム早期検出（WRED）の最小および最大しきい値またはキュー制限のしきい値をミリ秒（ms）単位で指定できます。以前は、これらのしきい値は、パケットまたはバイト単位でしか指定できませんでした。現在は、3つの測定単位をすべて使用できます。しきい値制限をポリシーマップで設定した後は、帯域幅の量が異なるものを含めて複数のインターフェイスでポリシー マップを使用できます。

- 機能情報の確認，49 ページ
- WRED およびキュー制限に関する QoS 時間ベースしきい値の前提条件，50 ページ
- WRED およびキュー制限に関する QoS 時間ベースしきい値の制約事項，50 ページ
- WRED およびキュー制限に関する QoS 時間ベースしきい値に関する情報，50 ページ
- WRED およびキュー制限に関する QoS 時間ベースしきい値を設定する方法，53 ページ
- WRED およびキュー制限に関する QoS 時間ベースしきい値の設定例，60 ページ
- その他の参考資料，64 ページ
- WRED およびキュー制限に関する QoS 時間ベースしきい値の機能情報，65 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。

WRED およびキュー制限に関する QoS 時間ベースしきい値の前提条件

この機能を設定する前に、トラフィック クラスを設定する必要があります。また、ポリシー マップが存在している必要があります。（適切な一致基準を指定する）トラフィック クラスとポリシー マップを作成するには、モジュラ Quality of Service（QoS）コマンドライン インターフェイス（MQC）を使用します。

WRED およびキュー制限に関する QoS 時間ベースしきい値の制約事項

この機能を使用すると、WRED しきい値またはキュー制限のしきい値をパケット（デフォルトの測定単位）、バイト、またはミリ秒（ms）単位で指定できます。ただし、これらの単位を混合することはできません。つまり、同じクラス、同じポリシー マップで測定単位を混合することはできません。たとえば、特定のクラスの最小しきい値をミリ秒単位で指定した場合は、そのクラスの最大しきい値もミリ秒単位で指定する必要があります。

WRED およびキュー制限に関する QoS 時間ベースしきい値に関する情報

WRED およびキュー制限に関する QoS 時間ベースしきい値の利点

追加の測定単位で指定されたキュー制限のしきい値

以前は、WRED のしきい値およびキュー制限のしきい値は、パケットまたはバイト単位でしか指定できませんでした。この機能を使用すると、しきい値をパケット、バイトまたはミリ秒単位で指定できます。これらの追加の測定単位によって、さらなる柔軟性が提供され、設定を微調整することができます。

ポリシー マップは複数のインターフェイスで必要に応じて再利用できます

WREDおよびキュー制限のしきい値は、ポリシーマップで指定され、設定されます。しきい値制限をポリシー マップで設定した後は、帯域幅の量が異なるものを含めて複数のインターフェイス

でポリシー マップを使用できます。これは、特定のインターフェイスで、あるクラスの帯域幅が使用可能な全帯域幅のパーセンテージとして指定されている場合に特に有用です。

WRED を使用したしきい値の設定

WRED は、輻輳回避メカニズムです。WRED は、ランダム早期検出（RED）アルゴリズムの機能を IP Precedence 機能と統合して、優先順位の高いパケットの優先的なトラフィック処理を実現します。WRED では、インターフェイスで輻輳が発生し始めると、優先順位が低いトラフィックを選択的に廃棄し、異なるサービス クラスに対して差別化したパフォーマンス特性を提供できます。

WRED は、キューイング戦略のような他の輻輳回避技術とは異なります。発生した輻輳をコントロールするのではなく、輻輳を予測し回避しようとするためです。

WRED は、**random-detect** コマンドを使用してイネーブルにします。その後、最小しきい値、最大しきい値、およびマーク確率値を設定して、適切なコマンドを使用することでパケットが受信する処理を決定できます。たとえば、**random-detect precedence** コマンドを使用して、特定の IP precedence のしきい値を決定できます。

queue-limit コマンドを使用したしきい値の設定

queue-limit コマンドを使用すると、ポリシーマップで設定されるクラス ポリシーに対してキューが保持できるパケットの最大数（つまり、しきい値）を指定または変更できます。クラスに属するパケットは、保証帯域割り当てと、トラフィック クラスの特性であるキュー制限に対応しています。**queue-limit** コマンドを使用すると、しきい値はクラス全体の集約しきい値となります。

キューが設定されたキュー制限に達した後、そのトラフィック クラスに追加パケットが入力されると、ポリシーマップの設定方法に応じてテール ドロップまたはWRED（設定されている場合）が発生します。（テール ドロップは輻輳を回避するための1つの手段であり、すべてのトラフィックが同等に扱われ、サービス クラスの間で差別化されません。）

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）。

テール ドロップは、輻輳を回避する手段としてWREDを使用してパケットをドロップするよう、明示的にサービス ポリシーを設定しない限り、分散型クラスベース均等化キューイング（DCBWFQ）トラフィック クラスで使用されます。サービス ポリシーを構成する 1 つまたは複数のトラフィック クラスで、テール ドロップの代わりに WRED を使用する場合、そのサービス ポリシーを適用するインターフェイスに WRED が設定されていないことを確認してください。

ミリ秒キーワードを使用した random-detect コマンド

この機能を使用すると、WREDの最小および最大しきい値をミリ秒（ms）単位で指定できます。次の表に示す **random-detect** コマンドで使用可能な **ms** キーワードを使用することで、しきい値をミリ秒単位で指定できます。

表 *6*：ミリ秒（*ms*）キーワードを使用した *random-detect* コマンド

コマンド	説明
random-detectclp	特定のセル損失率優先度（CLP）値、またはポリシー マップのクラス ポリシーに対する特定の CLP 値に対する WRED パラメータを設定します。
random-detectcos	特定のサービス クラス（CoS）値、またはポリシー マップのクラス ポリシーに対する特定の CoS 値に対する WRED パラメータを設定します。
random-detectdiscard-class	特定の discard-class、またはポリシー マップのクラス ポリシーに対する特定の discard-class に対する WRED パラメータを設定します。
random-detectdscp	特定の DiffServ コード ポイント（DSCP）値、またはポリシー マップのクラス ポリシーに対する特定の DSCP 値に対する WRED パラメータを設定します。
random-detectprecedence	特定の IP precedence、またはポリシー マップのクラス ポリシーに対する特定の IP precedence に対する WRED パラメータを設定します。

しきい値の測定単位の混合

この機能を使用すると、しきい値をパケット（デフォルトの測定単位）、バイト、またはミリ秒（ms）単位で指定できます。たとえば、WRED では、最小しきい値と最大しきい値をパケット、バイト、またはミリ秒単位で指定できます。ただし、単位を混合することはできません。たとえば、最小しきい値をミリ秒単位で指定した場合は、最大しきい値もミリ秒単位で指定する必要があります。

WRED およびキュー制限に関する QoS 時間ベースしきい値を設定する方法

WRED の有効化と WRED を使用したしきい値の指定

手順の概要

1. **enable**
2. **configureterminal**
3. **policy-map***policy-name*
4. **class** {*class-name***class-default**}
5. 設定を続けるには、帯域幅を指定するか、またはトラフィック シェーピングを有効にする必要があります。どちらかを選択します。
6. **bandwidth** {*bandwidth-kbps* | **remainingpercent***percentage* | **percent***percentage*}
7. **shape** [**average** | **peak**] *mean-rate* [*burst-size*] [*excess-burst-size*]
8. **random-detect**
9. **random-detectprecedence** {*precedence* | **rsvp**} *min-threshold* {**bytes**| **ms**|**packets**} *max-threshold*{**bytes** | **ms**| **packets**} [*mark-probability-denominator*]
10. **exit**

手順の詳細

	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	**configureterminal** 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***policy-name* 例： `Router(config)# policy-map policy1`	作成するポリシーマップの名前を指定します。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。

	コマンドまたはアクション	目的
ステップ 4	**class** {*class-name***class-default**} 例： `Router(config-pmap)# class class1`	ポリシーを設定または変更できるようにクラスを指定します。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。 • クラス名を入力するか、デフォルト クラス（class-default）を指定します。
ステップ 5	設定を続けるには、帯域幅を指定するか、またはトラフィック シェーピングを有効にする必要があります。どちらかを選択します。	
ステップ 6	**bandwidth** {*bandwidth-kbps* | **remainingpercent***percentage* | **percent***percentage*} 例： `Router(config-pmap-c)# bandwidth percent 40`	（任意）ポリシーマップに属するクラスに割り当てる帯域幅を指定または変更します。 • 設定または変更する帯域幅を入力します。
ステップ 7	**shape** [**average** | **peak**] *mean-rate* [*burst-size*] [*excess-burst-size*] 例： `Router(config-pmap-c)# shape average 51200`	（オプション）平均またはピーク レート トラフィック シェーピングを有効にします。 • 平均またはピーク トラフィック シェーピングのどちらかを指定します。
ステップ 8	**random-detect** 例： `Router(config-pmap-c)# random-detect`	WRED または分散型 WRED（DWRED）をイネーブルにします。
ステップ 9	**random-detectprecedence** {*precedence* | **rsvp**} *min-threshold* {**bytes**| **ms**|**packets**} *max-threshold*{**bytes** | **ms**| **packets**} [*mark-probability-denominator*] 例： `Router(config-pmap-c)# random-detect precedence 2 512 ms 1020 ms`	特定の IP precedence に対する WRED パラメータと DWRED パラメータを設定します。 • 必要に応じて、IP precedence または RSVP 値、およびしきい値を指定します。 （注） この例では、WRED パラメータが特定の IP precedence 値でトラフィックに指定されました。他の値は、他の **random-detect** コマンドを使用して指定できます。

	コマンドまたはアクション	目的
ステップ 10	**exit** 例： `Router(config-pmap-c)# exit`	（任意）ポリシー マップ クラス コンフィギュレーション モードを終了します。

queue-limit コマンドを使用したしきい値の指定

手順の概要

1. **enable**
2. **configureterminal**
3. **policy-map***policy-name*
4. **class**{*class-name***class-default**}
5. 設定を続けるには、帯域幅（queue-limit コマンドを使用したしきい値の指定，（55 ページ））を指定するか、またはトラフィック シェーピング（queue-limit コマンドを使用したしきい値の指定，（55 ページ））を有効にする必要があります。どちらかを選択します。
6. **bandwidth** {*bandwidth-kbps* | **remainingpercent***percentage* | **percent***percentage*}
7.
8. **shape** [**average** | **peak**] *mean-rate* [[*burst-size*] [*excess-burst-size*]]
9. **queue-limit***number-of-packets* [**bytes** | **ms** | **packets**]
10. **exit**

手順の詳細

	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	**configureterminal** 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。

	コマンドまたはアクション	目的

ステップ 3	**policy-map***policy-name*
	
	例：
	`Router(config)#`
	`policy-map`
	`policy1`
		作成するポリシー マップの名前を指定します。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**{*class-name***class-default**} 例： `Router(config-pmap)# class class1`	ポリシーを設定または変更できるようにクラスを指定します。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。 • クラス名を入力するか、デフォルト クラス（class-default）を指定します。
ステップ 5	設定を続けるには、帯域幅（queue-limit コマンドを使用したしきい値の指定，（55 ページ））を指定するか、またはトラフィック シェーピング（queue-limit コマンドを使用したしきい値の指定，（55 ページ））を有効にする必要があります。どちらかを選択します。	
ステップ 6	**bandwidth** {*bandwidth-kbps* | **remainingpercent***percentage* | **percent***percentage*} 例： `Router(config-pmap-c)# bandwidth percent 40`	（任意）ポリシー マップに属するクラスに割り当てる帯域幅を指定または変更します。 • 設定または変更する帯域幅を入力します。
ステップ 7		
ステップ 8	**shape** [**average** | **peak**] *mean-rate* [[*burst-size*] [*excess-burst-size*]] 例： `Router(config-pmap-c)# shape average 51200`	（オプション）平均またはピーク レート トラフィック シェーピングを有効にします。 • 平均またはピーク トラフィック シェーピングのどちらかを指定します。
ステップ 9	**queue-limit***number-of-packets* [**bytes** | **ms** | **packets**] 例： `Router(config-pmap-c)# queue-limit 200 ms`	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（つまり、キュー制限）を指定または変更します。 • キュー制限を入力します。測定単位は、バイト、ミリ秒、またはパケットのいずれかです。

	コマンドまたはアクション	目的
ステップ 10	**exit** 例： `Router(config-pmap-c)# exit`	（任意）ポリシーマップ クラス コンフィギュレーション モードを終了します。

WRED 設定の QoS 時間ベースしきい値におけるインターフェイスへのポリシー マップのアタッチ

（注） ネットワークのニーズにより、ポリシー マップをサブインターフェイス、ATM PVC、フレームリレー DLCI、または他のタイプのインターフェイスにアタッチする必要がある場合もあります。

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber*
4. **pvc** [*name*] *vpi* / *vci* [**ilmi** | **qsaal** | **smds**]
5. **service-policy** {**input**| **output**} *policy-map-name*
6. **exit**

手順の詳細

	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	**configureterminal** 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。

	コマンドまたはアクション	目的

ステップ 3	**interface***typenumber*
	
	例：
	`Router(config)#`
	`interface serial4/0`
		インターフェイス（サブインターフェイス）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。
		• インターフェイスのタイプ番号を入力します。

ステップ 4	**pvc** [*name*] *vpi* / *vci* [**ilmi** | **qsaal** | **smds**] 例： `Router(config-if)# pvc cisco 0/16 ilmi`	（任意）ATM PVC に名前を作成するか割り当て、ATM PVC でカプセル化タイプを指定します。ATM VC コンフィギュレーション モードを開始します。 （注） この手順は、ポリシー マップを ATM PVC に適用する場合にのみ必要です。ATM PVC にポリシー マップを関連付けない場合は、この手順をスキップして、ステップ 5 に進みます。

ステップ 5	**service-policy** {**input**| **output**} *policy-map-name*
	
	例：
	`Router(config-if)#`
	`service-policy output policy1`
	
	例：
		インターフェイスの入力方向または出力方向にアタッチするポリシー マップの名前を指定します。
		（注） ポリシー マップは、入力または出力ルータで設定できます。また、入力方向または出力方向のインターフェイスにも適用できます。ポリシーマップを適用する方向（入力または出力）とルータ（入力または出力）は、ネットワーク構成に従って変わります。**service-policy** コマンドを使用してポリシー マップをインターフェイスに適用する場合、ネットワーク構成に適したルータおよびインターフェイスの方向を選択してください。
		• ポリシー マップ名を入力します。

ステップ 6	**exit** 例： `Router(config-if)# exit`	（任意）インターフェイス コンフィギュレーション モードを終了します。

WRED およびキュー制限に関する QoS 時間ベースしきい値設定の確認

手順の概要

1. **enable**
2. **showpolicy-map**[*policy-map*]
3. and/or
4. **showpolicy-mapinterface***interface-name*
5. **exit**

手順の詳細

	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	**showpolicy-map**[*policy-map*] 例： `Router# show policy-map policy1`	一致基準を含めて、クラスマップに関するすべての情報が表示されます。 • クラス マップ名を入力します。
ステップ 3	and/or	

ステップ 4	**showpolicy-mapinterface***interface-name*
	
	例：
	
	`Router#`
	`show policy-map interface serial4/0`
		指定したインターフェイスまたはサブインターフェイス上か、インターフェイス上の特定の PVC に対し、すべてのサービス ポリシーに対して設定されているすべてのクラスのパケット統計情報を表示します。
		
		• インターフェイス名を入力します。

ステップ 5	**exit** 例： `Router# exit`	（任意）特権 EXEC モードを終了します。

トラブルシューティングのヒント

「設定の確認」の項に示すコマンドを使用すると、意図した設定を実現し、機能が正しく動作していることを確認できます。上記の **show** コマンドの使用後に、設定が正しくない、または機能が予想どおりに働いていないと判明した場合は、次の操作を実行します。

意図したとおりに設定が行われていない場合は、次の手順を完了します。

1. **show running-config** コマンドを使用して、コマンドの出力を分析します。
2. ポリシー マップが **show running-config** コマンドの出力に表示されない場合は、**logging console** コマンドをイネーブルにします。
3. ポリシー マップをインターフェイスに再度アタッチします。

パケットが正確に一致していない場合は（たとえば、パケット カウンタが正しく増加していないなど）、次の手順を完了します。

1. **show policy-map** コマンドを実行し、コマンドの出力を分析します。
2. **show running-config** コマンドを実行し、コマンドの出力を分析します。
3. **show policy-map interface** コマンドを実行し、コマンドの出力を分析します。次の内容を確認します。
   1. ポリシー マップにキューイングが適用され、パケットが正しいクラスに一致しているにもかかわらず、予期しない結果が生じる場合は、キューのパケット数と一致したパケット数を比較します。

インターフェイスが混雑していて、一致するパケット数が少ない場合、送信（tx）リングの調整を確認し、tx リングでキューイングが実行されているかどうかを評価します。そのためには、**show controllers** コマンドを使用し、コマンドの出力で tx 回数の値を確認します。

WRED およびキュー制限に関する QoS 時間ベースしきい値の設定例

WRED を使用したしきい値の設定例

次の例では、WRED は「policy1」と呼ばれるポリシー マップに設定されています。この WRED 設定では、帯域幅はパーセンテージ（80%）として指定され、IP precedence 2 の最小および最大しきい値はそれぞれ 512 ミリ秒と 1020 ミリ秒に設定されています。

```
Router> enable
Router# configure terminal
Router(config)#

policy-map policy1
```

```
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth percent 80
Router(config-pmap-c)# random-detect
Router(config-pmap-c)# random-detect precedence 2 512 ms 1020 ms
Router(config-pmap-c)# exit

Router(config-pmap)# exit

Router(config)# interface s4/0
Router(config-if)#

service-policy output policy1
Router(config-if)# end
```

queue-limit コマンドを使用したしきい値の設定例

次の例では、「policy2」という名前のポリシーマップが設定されました。ポリシーマップ policy2 には、「class1」という名前のクラスが含まれています。このクラスの帯域幅はパーセンテージ（80%）として指定され、しきい値を 200 ミリ秒に設定するために **queue-limit** コマンドが使用されました。

```
Router> enable
Router# configure terminal
Router(config)#

policy-map policy2
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth percent 80
Router(config-pmap-c)# queue-limit 200 ms
Router(config-pmap-c)# exit

Router(config-pmap)# exit

Router(config)# interface s4/0
Router(config-if)#

service-policy output policy1
Router(config-if)# end
```

設定の確認例

この機能が正しく設定されていることを確認するには、**show policy-map** コマンドまたは **show policy-map interface** コマンドのいずれかを使用します。

この項には、**show policy-map interface** コマンドおよび **show policy-map** コマンドからのサンプル出力の 2 つのセットが含まれています。1 つのセットは、機能の設定に WRED が使用された場合の出力を示し、もう 1 つのセットは、機能の設定に **queue-limit** コマンドが使用された場合の出力を示します。

WRED しきい値設定サンプル出力の例

次に、しきい値を指定するために WRED が使用されている場合の、**show policy-map** コマンドのサンプル出力を示します。「時間ベース WRED」という語は、しきい値がミリ秒（ms）単位で指定されていることを示します。

```
Router# show policy-map
  Policy Map policy1
    Class class1
      bandwidth 80 (%)
       time-based wred, exponential weight 9
      class    min-threshold    max-threshold    mark-probability
      ----------------------------------------------------------
      0       -                -                1/10
      1       -                -                1/10
      2       512              1024             1/10
      3       -                -                1/10
      4       -                -                1/10
      5       -                -                1/10
      6       -                -                1/10
      7       -                -                1/10
```

次に、しきい値を指定するために WRED が使用されている場合の、**show policy-map interface** コマンドのサンプル出力を示します。

```
Router# show policy-map interface Ethernet2/0
 Ethernet2/0
Service-policy output: policy1 (1100)
    Class-map: class1 (match-all) (1101/1)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol ftp (1102)
      Queueing
      queue limit 16 ms/ 16000 bytes
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
      bandwidth 80.00% (%) (8000 kbps)
        Exp-weight-constant: 9 (1/512)
        Mean queue depth: 0 ms/ 0 bytes
        class    Transmitted  Random drop   Tail drop    Minimum       Maximum       Mark
                 pkts/bytes   pkts/bytes    pkts/bytes   thresh        thresh        prob
                                                         ms/bytes      ms/bytes
        0        0/0          0/0           0/0           4/4000        8/8000       1/10
        1        0/0          0/0           0/0           4/4500        8/8000       1/10
        2        0/0          0/0           0/0         512/512000   1024/1024000    1/10
        3        0/0          0/0           0/0           5/5500        8/8000       1/10
        4        0/0          0/0           0/0           6/6000        8/8000       1/10
        5        0/0          0/0           0/0           6/6500        8/8000       1/10
        6        0/0          0/0           0/0           7/7000        8/8000       1/10
        7        0/0          0/0           0/0           7/7500        8/8000       1/10
    Class-map: class-default (match-any) (1105/0)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any  (1106)
        0 packets, 0 bytes
        5 minute rate 0 bps

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
```

しきい値をミリ秒からバイトに変換する式

しきい値をミリ秒からバイトに変換する場合は、次の式を使用します。

ミリ秒 * （クラスに設定されている帯域幅）/ 8 = 合計バイト数

この例では、次の数値が式に使用されます。

512 ms * 8000 kbps / 8 = 512000 バイト

（注） Class1 の帯域幅は 8000 kbps です。

queue-limit コマンドしきい値設定サンプル出力例

次に、しきい値をミリ秒単位で指定するために **queue-limit** コマンドが使用された場合の **show policy-map** コマンドの出力例を示します。

```
Router# show policy-map
  Policy Map policy1
    Class class1
      bandwidth 80 (%)
      queue-limit 200 ms
```

次に、しきい値を指定するために **queue-limit** コマンドが使用された場合の **show policy-map interface** コマンドからの出力例を示します。

```
Router# show policy-map interface
 Ethernet2/0
  Service-policy output: policy1 (1070)
    Class-map: class1 (match-all) (1071/1)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol ftp (1072)
      Queueing
      queue limit 200 ms/ 200000 bytes
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
      bandwidth 80.00% (%) (8000 kbps)
    Class-map: class-default (match-any) (1075/0)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
Match: any  (1076)
        0 packets, 0 bytes
        5 minute rate 0 bps

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
```

しきい値をミリ秒からバイトに変換する式

しきい値をミリ秒からバイトに変換する場合は、次の式を使用します。

ミリ秒 * （クラスに設定されている帯域幅）/ 8 = 合計バイト数

この例では、次の数値が式に使用されます。

200 ms * 8000 kbps / 8 = 200000 バイト

（注） Class1 の帯域幅は 8000 kbps です。

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』
Quality of Service（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	『*Cisco IOS Quality of Service Solutions Command Reference*』
テール ドロップ、RED および WRED などの輻輳回避メカニズム	『*Cisco IOS Quality of Service Solutions Configuration Guide*』
CBWFQ および DCBWFQ などの輻輳管理メカニズム	『*Cisco IOS Quality of Service Solutions Configuration Guide*』
バイトベースの WRED	「Byte-Based Weight Random Early Detection」モジュール

標準

規格	タイトル
新しい規格または変更された規格はサポートされていません。また、既存の規格に対するサポートに変更はありません。	--

MIB

MIB	MIB のリンク
新しい MIB または変更された MIB はサポートされていません。また、既存の MIB に対するサポートに変更はありません。	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
新しい RFC または変更された RFC はサポートされていません。また、既存の RFC に対するサポートに変更はありません。	--

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★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

WRED およびキュー制限に関する QoS 時間ベースしきい値の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *7*：***WRED*** **およびキュー制限に関する** ***QoS*** **時間ベースしきい値の機能情報**

機能名	リリース	機能情報
WRED およびキュー制限に関する QoS 時間ベースしきい値	Cisco IOS XE Release 3.2S	WRED およびキュー制限に関する QoS 時間ベースしきい値の機能を使用すると、重み付けランダム早期検出（WRED）の最小および最大しきい値またはキュー制限のしきい値をミリ秒（ms）単位で指定できます。 **queue-limit**、**random-detectprecedence**、**showpolicy-map**、**showpolicy-mapinterface** の各コマンドが導入または変更されました。

第 8 章

DiffServ 準拠の WRED

DiffServ 準拠の WRED は、重み付けランダム早期検出の機能を拡張し、DiffServ と保証型転送（AF）の Per-Hop Behavior（PHB）のサポートを可能にします。この機能により、Differentiated Services Code Point（DSCP; DiffServ コードポイント）の値に従ってパケットをカラリングし、それらのパケットに優先ドロップ確率を割り当てることで、AF PHB を実装できます。

（注） この機能は、IP パケットでのみ使用できます。マルチプロトコルラベルスイッチング（MPLS）カプセル化パケットで使用することは意図されていません。

- 機能情報の確認，67 ページ
- DiffServ 準拠の WRED に関する情報，68 ページ
- DiffServ 準拠の WRED を設定する方法，68 ページ
- DiffServ 準拠の WRED の設定例，72 ページ
- その他の参考資料，72 ページ
- DiffServ 準拠の WRED の機能情報，74 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。

DiffServ 準拠の WRED に関する情報

WRED の差別化サービス

差別化サービスは、異なる Quality of Service（QoS）要件を満たすことができる複数のサービスモデルです。差別化サービスでは、ネットワークは各パケットによって指定された QoS に基づいて特定の種類のサービスを提供しようとします。この仕様は、さまざまな方法で実行できます。DiffServ 準拠の WRED 機能を使用すると、WRED は、パケットの廃棄確率を計算する際、IP パケットで 6 ビットの DiffServ コードポイント（DSCP）または IP Precedence 設定を使用できます。DSCP 値は、IP タイプ オブ サービス（ToS）バイトの最初の 6 ビットです。

DiffServ 準拠の WRED に関する使用上のガイドライン

DiffServ 準拠の WRED 機能を設定するには、最初にポリシーマップを指定し、クラスを追加してから、クラスに対して帯域幅またはシェイプを設定します。廃棄確率を計算する際に WRED で DSCP 値を使用する場合は、**random-detect** コマンドで *dscp-based* 引数を使用して DSCP 値を指定し、次に **random-detectdscp** コマンドを使用して DSCP 値のデフォルトの最小および最大しきい値を変更します。廃棄確率を計算する際に WRED で IP Precedence 値を使用する場合は、**random-detect** コマンドで *precedence-based* 引数を使用して IP Precedence 値を指定します。この設定は、ポリシーマップが適用される場所ではどこでも適用できます（インターフェイス レベル、VC 単位レベル、シェーパー レベルなど）。

この機能に含まれるコマンドを使用する際は、次の点に留意してください。

- *dscp-based* 引数を使用する場合、WRED では廃棄確率の計算に DSCP 値が使用されます。
- *precedence-based* 引数を使用する場合、WRED では廃棄確率の計算に IP Precedence 値が使用されます。
- *dscp-based* 引数と *precedence-based* 引数は、相互に排他的です。
- いずれの引数も指定しない場合、WRED では廃棄確率の計算に IP Precedence 値が使用されます（デフォルトの方式です）。

DiffServ 準拠の WRED を設定する方法

DiffServ 準拠の WRED の設定

この例では、パケットの廃棄確率を計算する場合に DSCP 値を使用するように DiffServ 準拠の WRED を設定します。

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber* [**name-tag**]
4. **class-map***class-map-name*
5. **match***match-criterion*
6. **policy-map***policy-map-name*
7. **class** {*class-name* | **class-default**}
8. **bandwidth** {*kbps* | **remaining***percentage* | **percent***percentage*}
9. **random-detect**[**dscp-based** | **precedence-based**]
10. **random-detectdscp***dscp-valuemin-thresholdmax-threshold* [*mark-probability-denominator*]
11. **exit**
12. **exit**
13. **interface***typenumber* [**name-tag**]
14. **service-policy output***policy-map-name*
15. **end**
16. **showpolicy-mapinterface***typenumber*
17. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface GigabitEthernet 0/0/0`	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。 • インターフェイス タイプと番号を入力します。

	コマンドまたはアクション	目的
ステップ 4	**class-map***class-map-name* 例： `Device(config-if)# class-map diffservclass`	作成するクラス マップの名前を指定し、QoS 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 5	**match***match-criterion* 例： `Device(config-cmap)# match any`	クラス マップの一致基準を設定します。
ステップ 6	**policy-map***policy-map-name* 例： `Device(config-cmap)# policy-map diffservpm`	サービス ポリシーを指定するために 1 つ以上のインターフェイスに適用可能なポリシー マップを作成または修正し、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。
ステップ 7	**class** {*class-name* | **class-default**} 例： `Device(config-pmap)# class diffservclass`	作成または変更するポリシーのクラス名を指定するか、ポリシーを指定する前にデフォルト クラス（一般に class-default クラスといいます）を指定します。 • 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。
ステップ 8	**bandwidth** {*kbps* | **remaining***percentage* | **percent***percentage*} 例： `Device(config-pmap-c)# bandwidth percent 30`	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 9	**random-detect**[**dscp-based** | **precedence-based**] 例： `Device(config-pmap-c)# random-detect dscp-based`	ポリシー マップ内のクラスに WRED を設定します。
ステップ 10	**random-detectdscp***dscp-valuemin-thresholdmax-threshold* [*mark-probability-denominator*] 例： `Device(config-pmap-c)# random-detect dscp af11 10000 30000 25`	DiffServ コード ポイント（DSCP）値の最小および最大パケットしきい値を変更します。

	コマンドまたはアクション	目的
ステップ 11	**exit** 例： `Device(config-pmap-c)# exit`	QoS ポリシーマップ クラス コンフィギュレーション モードを終了します。
ステップ 12	**exit** 例： `Device(config-pmap)# exit`	QoS ポリシーマップ コンフィギュレーション モードを終了します。
ステップ 13	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface GigabitEthernet 0/0/0`	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。 • インターフェイス タイプと番号を入力します。
ステップ 14	**service-policy output***policy-map-name* 例： `Device(config-if)# service-policy output policy1`	ポリシー マップを出力インターフェイスに付加します。 • ポリシー マップ名を入力します。 （注） ポリシー マップは、入力または出力ルータで設定できます。また、入力方向または出力方向のインターフェイスにも適用できます。ポリシー マップを適用する方向（入力または出力）とルータ（入力または出力）は、ネットワーク構成に従って変わります。**service-policy** コマンドを使用してポリシー マップをインターフェイスに適用する場合、ネットワーク構成に適したルータおよびインターフェイスの方向を選択してください。
ステップ 15	**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 16	**showpolicy-mapinterface***typenumber* 例： `Device# show policy-map interface GigabitEthernet 0/0/0`	（任意）指定されたインターフェイスまたはサブインターフェイスとインターフェイス上の特定の PVC のどちらかで、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のトラフィック統計情報を表示します。 • インターフェイス タイプと番号を入力します。
ステップ 17	**exit** 例： `Device# exit`	（任意）特権 EXEC モードを終了します。

DiffServ 準拠の WRED の設定例

例：DiffServ 準拠の WRED

次に、WRED がクラス c1 に DSCP 値 8 を使用できるようにする例を示します。DSCP 値 8 の最小しきい値は 24 で、最大しきい値は 40 です。最後の行は、トラフィック ポリシーを出力インターフェイスまたは VC p1 に適用します。

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# class-map c1
Device(config-cmap)# match ip precedence 1
Device(config-cmap)# policy-map p1
Device(config-pmap)# class c1
Device(config-pmap-c)# bandwidth 48
Device(config-pmap-c)# random-detect dscp-based
Device(config-pmap-c)# random-detect dscp 8 24 40 (bytes/ms)
Device(config-if)# service-policy output p1
```

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』

関連項目	マニュアル タイトル
QoS コマンド	『*Cisco IOS Quality of Service Solutions Command Reference*』
MQC	『*QoS: Modular QoS: Command-Line Interface Configuration Guide*』

標準および RFC

標準/RFC	タイトル
RFC 2474	『*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*』
RFC 2475	『*An Architecture for Differentiated Services Framework*』
RFC 2597	『*Assured Forwarding PHB*』
RFC 2598	『*An Expedited Forwarding PHB*』

MIB

MIB	MIB のリンク
CISCO-CLASS-BASED-QOS-MIB CISCO-CLASS-BASED-QOS-CAPABILITY-MIB	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★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

DiffServ 準拠の WRED の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *8*：*DiffServ* 準拠の *WRED* の機能情報

機能名	リリース	機能情報
DiffServ 準拠の WRED	Cisco IOS XE Release 3.6S	DiffServ 準拠の WRED で、DiffServ および AF Per-Hop Behavior をサポートできるよう WRED の機能を拡張します。 Cisco IOS XE リリース 3.6S では、Cisco ASR 903 ルータのサポートが追加されました。 **random-detect**、**random-detect dscp**、**random-detect precedence** の各コマンドが導入または変更されました。

第 9 章

ダイヤラ インターフェイス上のシェーピング

ダイヤラ インターフェイス上のシェーピング機能では、ダイヤラ インターフェイスでの Point-to-Point Protocol over Ethernet（PPPoE）設定および非同期転送モードを介した Point-to-Point Protocol（PPPoA）設定に対するサポートが提供されます。この機能では、モジュラ QoS CLI（MQC）ベースのキューイングおよびシェーピングに対するサポートが提供され、顧客ごとの Quality of Service（QoS）がサポートされます。親ポリシーはファースト マイルのイーサネット（EFM）インターフェイスにアタッチされ、子ポリシーは個々のダイヤラ インターフェイスにアタッチされます。サービス クラス（CoS）値は、ダイヤラ インターフェイスにポリシーを適用することによって設定されます。この機能では、ダイヤラ インターフェイスのキューイング統計情報の収集と、ダイヤラ インターフェイスのトラフィック カウンタのポーリングも有効になります。

- 機能情報の確認，75 ページ
- ダイヤラ インターフェイス上のシェーピングに関する制約事項，76 ページ
- ダイヤラ インターフェイス上のシェーピングに関する情報，76 ページ
- ダイヤラ インターフェイス上のシェーピングを設定する方法，77 ページ
- ダイヤラ インターフェイス上のシェーピングの設定例，98 ページ
- ダイヤラ インターフェイス上のシェーピングに関する追加の参考資料，100 ページ
- ダイヤラ インターフェイス上のシェーピングに関する機能情報，101 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。

ダイヤラインターフェイス上のシェーピングに関する制約事項

- 出力キューイング ポリシーには、親の class-default シェーパーが必要であり、他のキューイング アクションは子ポリシーで設定する必要があります。

ダイヤラインターフェイス上のシェーピングに関する情報

ダイヤラ インターフェイス上の PPP セッションでの QoS

ダイヤラ インターフェイス上のシェーピング機能では、出力インターフェイス（すべてのキューイング機能が動作している）で出力キューイングと分類が統合されます。ポリシング機能および設定機能（CoS マーキングなど）も、出力パスで動作します。

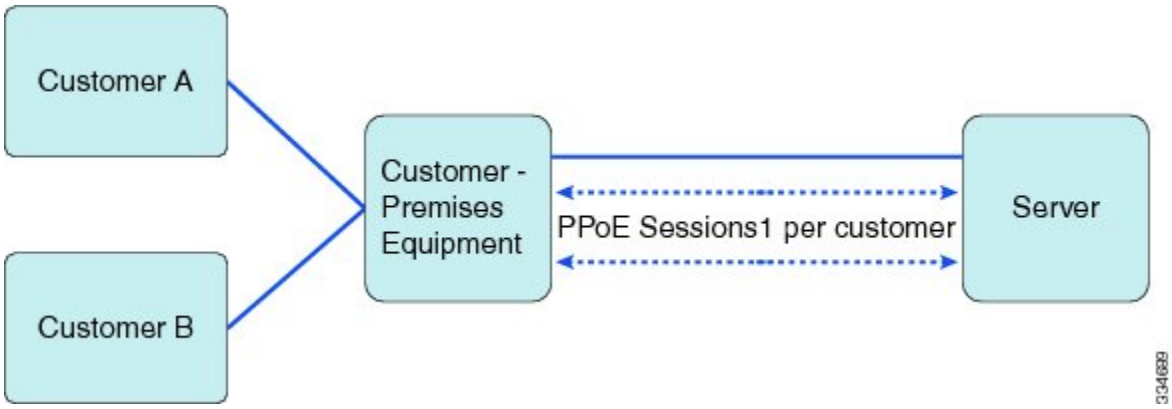
MQC ベースの QoS キューイングおよびシェーピング機能を使用して、フラットな class-default 型のポリシーを EFM にアタッチでき、HQoS の親型のポリシーをダイヤラ インターフェイスにアタッチできます。

ポリシーは、**service-policy** コマンドを使用してダイヤラ インターフェイスに適用されます。また、関連する show コマンドと debug コマンドによって、ダイヤラ ターゲットに関連付けられたポリシーとキューイングの統計情報が表示されます。

QoS のダイヤラ インターフェイス トポロジ

次の図に、ダイヤラ インターフェイス上のシェーピング機能に対してサポートされているトポロジを示します。

図 *3*：ダイヤラ インターフェイス上のシェーピング トポロジ



顧客宅内機器（CPE）は、複数の顧客間で共有されます。各顧客は、ギガビット イーサネット ポートの VLAN を介して CPE に接続します。CPE は EFM インターフェイス（これはイーサーネット接続のように見えますが DSL を使用します）を使用して DSL 経由でサービスに接続し、それを経由してすべての着信 VLAN が転送されます。各 VLAN（顧客）のトラフィックは個別の PPP セッションで送信されます。各セッションはダイヤラ インターフェイスを使用して設定されます。

ダイヤラインターフェイス上のシェーピングを設定する方法

ダイヤラ インターフェイスの出力キューイング ポリシーの設定

はじめる前に

ダイヤラ ターゲットはダイナミック ターゲット API に追加されるので、出力キューイング ポリシーには、親の class-default シェーパーが必要であり、他のキューイング アクションは子ポリシーに設定される必要があります。

手順の概要

1. **enable**
2. **configureterminal**
3. **policy-map***policy-map-name*
4. **class***class-name*
5. **prioritypercent***percentage*
6. **exit**
7. **class***class-name*
8. **bandwidthpercent***percentage*
9. **exit**
10. **class** {*class-name* | **class-default**}
11. **fair-queue**
12. **exit**
13. **exit**
14. **policy-map***policy-map-name*
15. **classclass-default**
16. **shape average** *target-bit-rate*
17. **service-policy** *policy-map-name*
18. **exit**
19. **exit**
20. **interface***typenumber*
21. **service-policy output** *policy-name*
22. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**policy-map***policy-map-name* 例： `Device(config)# policy-map child`	事前に作成したポリシー マップの名前を指定して、ポリシー マップ コンフィギュレーション モードに入ります。 • ポリシー マップ名を入力します。
ステップ 4	**class***class-name* 例： `Device(config-pmap)# class voice`	作成するポリシーのクラス名を指定し、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。このクラスは、以前に作成したクラス マップと関連付けられます。 • クラス名を入力するか、**class-default** キーワードを入力します。
ステップ 5	**prioritypercent***percentage* 例： `Device(config-pmap-c)# priority percent 30`	保証帯域幅の分量が、使用可能な帯域幅の割合（%）によって指定されることを、指定します。
ステップ 6	**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 7	**class***class-name* 例： `Device(config-pmap)# class video`	作成するポリシーのクラス名を指定し、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。このクラスは、以前に作成したクラス マップと関連付けられます。 • クラス名を入力するか、**class-default** キーワードを入力します。
ステップ 8	**bandwidthpercent***percentage* 例： `Device(config-pmap-c)# bandwidth percent 50`	保証帯域幅の分量が合計帯域幅の割合によって指定されるよう指定します。
ステップ 9	**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。

	コマンドまたはアクション	目的
ステップ 10	**class** {*class-name* | **class-default**} 例： `Device(config-pmap)# class class-default`	作成するポリシーのクラス名を指定し、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。このクラスは、以前に作成したクラス マップと関連付けられます。 • クラス名を入力するか、**class-default** キーワードを入力します。
ステップ 11	**fair-queue** 例： `Device(config-pmap-c)# fair-queue`	このクラスでのフローベース均等化キューイングを有効にします。
ステップ 12	**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 13	**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 14	**policy-map***policy-map-name* 例： `Device(config)# policy-map parent`	ポリシー マップの名前を指定して、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 15	**classclass-default** 例： `Device(config-pmap)# class class-default`	class-default クラスを作成します。
ステップ 16	**shape average** *target-bit-rate* 例： `Device(config-pmap-c)# shape average 1000000`	インターフェイスでの平均レート トラフィック シェーピングをビット□秒として指定します。
ステップ 17	**service-policy** *policy-map-name* 例： `Device(config-pmap-c)# service policy child`	サービス ポリシーのポリシー 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 18	**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 19	**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 20	**interface***typenumber* 例： `Device(config)# interface Dialer 0`	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。 • インターフェイス タイプと番号を入力します。
ステップ 21	**service-policy output** *policy-name* 例： `Device(config-if)# service-policy output parent`	ポリシーマップを出力インターフェイスに適用し、このインターフェイスでサービスポリシーとして使用されるようにします。
ステップ 22	**exit** 例： `Device(config-if) 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。

ダイヤラ インターフェイスの PPPoEoA 用の QoS の設定

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber* [**name-tag**]
4. **no ip address**
5. **no atm ilmi-keepalive**
6. **exit**
7. **interface***typenumber* [**name-tag**]
8. **pvc** *vpi*/*vci*
9. **vbr-nrt** *output-pcr output-scr*
10. **pppoe-client dial-pool-number** *number*
11. **exit**
12. **exit**
13. **interface***typenumber* [**name-tag**]
14. **mtu** *ip-address*
15. **ip address** *ip-address mask*
16. **encapsulation** *encapsulation-type*
17. **dialer pool** *number*
18. **dialer-group** *number*
19. **service-policy output** *name*
20. **exit**
21. **dialer-list** *dialer-group* **protocol** *protocol-name* **permit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface ATM 0`	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。 • インターフェイス タイプと番号を入力します。
ステップ 4	**no ip address** 例： `Device(config-if)# no ip address`	インターフェイスで IP 処理をディセーブルにします。
ステップ 5	**no atm ilmi-keepalive** 例： `Device(config-if)# no atm ilmi-keepalive`	インターフェイスで暫定ローカル管理インターフェイス（ILMI）キープアライブをディセーブルにします。
ステップ 6	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 7	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface ATM 0.1 point-to-point`	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。 • インターフェイスのタイプ、番号、および名前を入力します。
ステップ 8	**pvc** *vpi*/*vci* 例： `Device(config-if)# pvc 4/46`	ATM 相手先固定接続（PVC）を作成して、ATM 仮想回線コンフィギュレーション サブモードを開始します。 • この PVC の ATM ネットワーク仮想パス識別子（VPI）および ATM ネットワーク仮想チャネル識別子（VCI）を入力します。
ステップ 9	**vbr-nrt** *output-pcr output-scr* 例： `Device(config-if-atm-vc)# vbr-nrt 738 738`	可変ビット レート非リアルタイム（VBR-NRT）Quality of Service（QoS）を設定し、ATM 相手先固定接続（PVC）の出力ピーク セル レート（PCR）および出力平均セル レート（SCR）を指定します。

	コマンドまたはアクション	目的
ステップ 10	**pppoe-client dial-pool-number** *number* 例： `Device(config-if-atm-vc)# pppoe-client dial-pool-number 1`	PPP over Ethernet（PPPoE）クライアントを設定し、ダイヤルオンデマンド ルーティング（DDR）の機能を指定します。
ステップ 11	**exit** 例： `Device(config-if-atm-vc)# exit`	ATM 仮想回線コンフィギュレーション モードを終了します。
ステップ 12	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 13	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 0`	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。 • インターフェイス タイプと番号を入力します。
ステップ 14	**mtu** *ip-address* 例： `Device(config-if)# mtu 1200`	最大パケット サイズまたは最大伝送ユニット（MTU）サイズを調整します。
ステップ 15	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 172.16.0.0 255.0.0.0`	インターフェイスのプライマリ IP アドレスを設定します。 • IP アドレスと IP アドレス マスクを入力します。
ステップ 16	**encapsulation** *encapsulation-type* 例： `Device(config-if)# encapsulation ppp`	インターフェイスで使用するカプセル化方式を設定します。
ステップ 17	**dialer pool** *number* 例： `Device(config-if)# dialer pool 1`	ダイヤラ インターフェイスが特定の宛先サブネットワークへ接続する際に使用するダイヤリング プールを指定します。

	コマンドまたはアクション	目的
ステップ 18	**dialer-group** *number* 例： `Device(config-if)# dialer-group 1`	特定のダイヤリング グループに属するようインターフェイスを設定して、アクセスを制御します。
ステップ 19	**service-policy output** *name* 例： `Device(config-if)# service-policy output dialer-output-sp`	ポリシーマップを出力インターフェイスに適用し、このインターフェイスでサービスポリシーとして使用されるようにします。
ステップ 20	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 21	**dialer-list** *dialer-group* **protocol** *protocol-name* **permit** 例： `Device(config)# dialer-list 1 protocol ip permit`	プロトコルによって、またはプロトコルと以前に定義したアクセスリストの組み合わせによって、ダイヤルするためのダイヤルオンデマンドルーティング（DDR）ダイヤラリストを定義します。

ダイヤラ インターフェイスの PPPoE 用の QoS の設定

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber* [**name-tag**]
4. **ppp enable group** *group-name*
5. **pppoe-client dial-pool-number** *number*
6. **exit**
7. **interface***typenumber* [**name-tag**]
8. **mtu** *ip-address*
9. **ip address** *ip-address mask*
10. **encapsulation** *encapsulation-type*
11. **dialer pool** *number*
12. **dialer-group** *number*
13. **service-policy output** *name*
14. **exit**
15. **dialer-list** *dialer-group* **protocol** *protocol-name* **permit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Ethernet 0/0`	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。 • インターフェイスのタイプ、番号、および名前を入力します。

	コマンドまたはアクション	目的
ステップ 4	**ppp enable group** *group-name* 例： `Device(config-if)# ppp enable group global`	イーサネット インターフェイスまたはサブインターフェイスで PPPoE セッションをイネーブルにします。
ステップ 5	**pppoe-client dial-pool-number** *number* 例： `Device(config-if)# pppoe-client dial-pool-number 1`	PPPoE クライアントを設定し、ダイヤルオンデマンドルーティング（DDR）の機能を指定します。
ステップ 6	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 7	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 0`	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。 • インターフェイス タイプと番号を入力します。
ステップ 8	**mtu** *ip-address* 例： `Device(config-if)# mtu 1200`	最大パケット サイズまたは最大伝送ユニット（MTU）サイズを調整します。
ステップ 9	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 172.16.0.0 255.0.0.0`	インターフェイスのプライマリ IP アドレスを設定します。 • IP アドレスと IP アドレス マスクを入力します。
ステップ 10	**encapsulation** *encapsulation-type* 例： `Device(config-if)# encapsulation ppp`	インターフェイスで使用するカプセル化方式を設定します。
ステップ 11	**dialer pool** *number* 例： `Device(config-if)# dialer pool 1`	ダイヤラ インターフェイスが特定の宛先サブネットワークへ接続する際に使用するダイヤリング プールを指定します。

	コマンドまたはアクション	目的
ステップ 12	**dialer-group** *number* 例： `Device(config-if)# dialer-group 1`	特定のダイヤリング グループに属するようインターフェイスを設定して、アクセスを制御します。
ステップ 13	**service-policy output** *name* 例： `Device(config-if)# service-policy output dialer-output-sp`	ポリシー マップを出力インターフェイスに適用し、このインターフェイスでサービス ポリシーとして使用されるようにします。
ステップ 14	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 15	**dialer-list** *dialer-group* **protocol** *protocol-name* **permit** 例： `Device(config)# dialer-list 1 protocol ip permit`	プロトコルによって、またはプロトコルと以前に定義したアクセス リストの組み合わせによって、ダイヤルするためのダイヤルオンデマンド ルーティング（DDR）ダイヤラ リストを定義します。

ダイヤラ インターフェイスの PPPoA 用の QoS の設定

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber* [**name-tag**]
4. **pvc** *vpi*/*vci*
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **dialer pool-member** *number*
7. **protocol** *protocol*
8. **exit**
9. **exit**
10. **interface***typenumber* [**name-tag**]
11. **mtu** *ip-address*
12. **ip address** *ip-address mask*
13. **encapsulation** *encapsulation-type*
14. **dialer pool** *number*
15. **dialer-group** *number*
16. **service-policy output** *name*
17. **exit**
18. **dialer-list** *dialer-group* **protocol** *protocol-name* **permit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface ATM 0.1 point-to-point`	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。 • インターフェイスのタイプ、番号、および名前を入力します。

	コマンドまたはアクション	目的
ステップ 4	**pvc** *vpi/vci* 例： `Device(config-if)# pvc 4/46`	ATM 相手先固定接続（PVC）を作成して、ATM 仮想回線コンフィギュレーション サブモードを開始します。 • この PVC の ATM ネットワーク仮想パス識別子（VPI）および ATM ネットワーク仮想チャネル識別子（VCI）を入力します。
ステップ 5	**vbr-nrt** *output-pcr output-scr output-maxburstsize* 例： `Device(config-if-atm-vc)# vbr-nrt 738 738 32`	可変ビット レート非リアルタイム（VBR-NRT）Quality of Service（QoS）を設定し、ATM 相手先固定接続（PVC）の出力ピーク セル レート（PCR）、出力平均セル レート（SCR）、および出力最大バースト セル サイズを指定します。
ステップ 6	**dialer pool-member** *number* 例： `Device(config-if-atm-vc)# dialer pool-member 1`	ダイヤラ プロファイルのダイヤリング プールのメンバーにする物理インターフェイスを設定します。
ステップ 7	**protocol** *protocol* 例： `Device(config-if-atm-vc)# protocol ppp dialer`	ATM 相手先固定接続（PVC）、相手先選択接続（SVC）、または仮想回線（VC）クラスのスタティック マップを設定します。
ステップ 8	**exit** 例： `Device(config-if-atm-vc)# exit`	ATM 仮想回線コンフィギュレーション モードを終了します。
ステップ 9	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 10	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 0`	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。 • インターフェイス タイプと番号を入力します。

	コマンドまたはアクション	目的
ステップ 11	**mtu** *ip-address* 例： `Device(config-if)# mtu 1200`	最大パケット サイズまたは最大伝送ユニット（MTU）サイズを調整します。
ステップ 12	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 172.16.0.0 255.0.0.0`	インターフェイスのプライマリ IP アドレスを設定します。 • IP アドレスと IP アドレス マスクを入力します。
ステップ 13	**encapsulation** *encapsulation-type* 例： `Device(config-if)# encapsulation ppp`	インターフェイスで使用するカプセル化方式を設定します。
ステップ 14	**dialer pool** *number* 例： `Device(config-if)# dialer pool 1`	ダイヤラ インターフェイスが特定の宛先サブネットワークへ接続する際に使用するダイヤリングプールを指定します。
ステップ 15	**dialer-group** *number* 例： `Device(config-if)# dialer-group 1`	特定のダイヤリング グループに属するようインターフェイスを設定して、アクセスを制御します。
ステップ 16	**service-policy output** *name* 例： `Device(config-if)# service-policy output dialer-output-sp`	ポリシー マップを出力インターフェイスに適用し、このインターフェイスでサービス ポリシーとして使用されるようにします。
ステップ 17	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 18	**dialer-list** *dialer-group* **protocol** *protocol-name* **permit** 例： `Device(config)# dialer-list 1 protocol ip permit`	プロトコルによって、またはプロトコルと以前に定義したアクセス リストの組み合わせによって、ダイヤルするためのダイヤルオンデマンドルーティング（DDR）ダイヤラ リストを定義します。

ダイヤラ インターフェイス上の複数セッション用の QoS の設定

手順の概要

1. **enable**
2. **configureterminal**
3. **interface***typenumber* [**name-tag**]
4. **ppp enable group** *group-name*
5. **pppoe-client dial-pool-number** *number*
6. **pppoe-client dial-pool-number** *number*
7. **pppoe-client dial-pool-number** *number*
8. **exit**
9. **interface***typenumber* [**name-tag**]
10. **dialer pool** *number*
11. **service-policy output** *name*
12. **exit**
13. **interface***typenumber* [**name-tag**]
14. **dialer pool** *number*
15. **service-policy output** *name*
16. **exit**
17. **interface***typenumber* [**name-tag**]
18. **dialer pool** *number*
19. **service-policy output** *name*
20. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Ethernet 0/0`	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。 • インターフェイスのタイプ、番号、および名前を入力します。
ステップ 4	**ppp enable group** *group-name* 例： `Device(config-if)# ppp enable group global`	イーサネット インターフェイスまたはサブインターフェイスで PPPoE セッションをイネーブルにします。
ステップ 5	**pppoe-client dial-pool-number** *number* 例： `Device(config-if)# pppoe-client dial-pool-number 1`	PPPoE クライアントを設定し、ダイヤルオンデマンド ルーティング（DDR）の機能を指定します。
ステップ 6	**pppoe-client dial-pool-number** *number* 例： `Device(config-if)# pppoe-client dial-pool-number 2`	PPPoE クライアントを設定し、ダイヤルオンデマンド ルーティング（DDR）の機能を指定します。
ステップ 7	**pppoe-client dial-pool-number** *number* 例： `Device(config-if)# pppoe-client dial-pool-number 3`	PPPoE クライアントを設定し、ダイヤルオンデマンド ルーティング（DDR）の機能を指定します。
ステップ 8	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 9	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 0`	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。 • インターフェイス タイプと番号を入力します。
ステップ 10	**dialer pool** *number* 例： `Device(config-if)# dialer pool 1`	ダイヤラ インターフェイスが特定の宛先サブネットワークへ接続する際に使用するダイヤリング プールを指定します。

	コマンドまたはアクション	目的
ステップ 11	**service-policy output** *name* 例： `Device(config-if)# service-policy output dialer-output-sp`	ポリシー マップを出力インターフェイスに適用し、このインターフェイスでサービス ポリシーとして使用されるようにします。
ステップ 12	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 13	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 1`	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。 • インターフェイス タイプと番号を入力します。
ステップ 14	**dialer pool** *number* 例： `Device(config-if)# dialer pool 2`	ダイヤラ インターフェイスが特定の宛先サブネットワークへ接続する際に使用するダイヤリング プールを指定します。
ステップ 15	**service-policy output** *name* 例： `Device(config-if)# service-policy output dialer-output-sp`	ポリシー マップを出力インターフェイスに適用し、このインターフェイスでサービス ポリシーとして使用されるようにします。
ステップ 16	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。
ステップ 17	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 2`	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。 • インターフェイス タイプと番号を入力します。
ステップ 18	**dialer pool** *number* 例： `Device(config-if)# dialer pool 3`	ダイヤラ インターフェイスが特定の宛先サブネットワークへ接続する際に使用するダイヤリング プールを指定します。

	コマンドまたはアクション	目的
ステップ 19	**service-policy output** *name* 例： `Device(config-if)# service-policy output dialer-output-sp`	ポリシー マップを出力インターフェイスに適用し、このインターフェイスでサービス ポリシーとして使用されるようにします。
ステップ 20	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了します。

ダイヤラ インターフェイスへの CoS 値の適用

サービス クラス（CoS）値は、ダイヤラ インターフェイスにポリシーを適用することによって設定されます。

手順の概要

1. **enable**
2. **configureterminal**
3. **policy-map***policy-map-name*
4. **class class-default**
5. **set cos***cos-value*
6. **exit**
7. **exit**
8. **interface***typenumber* [**name-tag**]
9. **service-policy output** *name*
10. **exit**
11. **interface***typenumber* [**name-tag**]
12. **encapsulation** *encapsulation-type*
13. **pppoe-client dial-pool-number** *number*
14. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> 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	**configureterminal** 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 3	**policy-map***policy-map-name* 例： `Device(config)# policy-map output_cos`	事前に作成したポリシーマップの名前を指定して、ポリシーマップ コンフィギュレーション モードに入ります。 • ポリシー マップ名を入力します。
ステップ 4	**class class-default** 例： `Device(config-pmap)# class class-default`	トラフィック分類用のデフォルトのクラスを作成し、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。
ステップ 5	**set cos***cos-value* 例： `Device(config-pmap-c)# set cos 1`	0 ～ 7 の IEEE 802.1Q CoS 値を指定します。
ステップ 6	**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 7	**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 8	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Dialer 1`	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。 • インターフェイス タイプと番号を入力します。

	コマンドまたはアクション	目的
ステップ 9	**service-policy output** *name* 例： `Device(config-if)# service-policy output output-cos`	ポリシーマップを出力インターフェイスに適用し、このインターフェイスでサービスポリシーとして使用されるようにします。
ステップ 10	**exit** 例： `Device(config-if)# exit`	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ます。
ステップ 11	**interface***typenumber* [**name-tag**] 例： `Device(config)# interface Ethernet 0.10`	インターフェイスタイプを設定して、サブインターフェイス コンフィギュレーション モードを開始します。 • インターフェイス タイプと番号を入力します。
ステップ 12	**encapsulation** *encapsulation-type* 例： `Device(config-subif)# encapsulation dot1q 10`	インターフェイスで使用するカプセル化方式を設定します。
ステップ 13	**pppoe-client dial-pool-number** *number* 例： `Device(config-subif)# pppoe-client dial-pool-number 1`	PPPoE クライアントを設定し、ダイヤルオンデマンドルーティング（DDR）の機能を指定します。
ステップ 14	**exit** 例： `Device(config-subif)# 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。

ダイヤラ インターフェイス上のシェーピングの設定例

例：ダイヤラ インターフェイス用の出力キューイング ポリシーの設定

次の例に、親と子のポリシーマップを設定する方法と、親マップをダイヤラインターフェイスにアタッチする方法を示します。

```
Device(config)# policy-map childExample
 Device(config-pmap)# class voice
  Device(config-pmap-c)# priority percent 30
  Device(config-pmap-c)# exit

 Device(config-pmap)# class video
  Device(config-pmap-c)# bandwidth percent 50
  Device(config-pmap-c)# exit

 Device(config-pmap)# class class-default
  Device(config-pmap-c)# fair-queue
  Device(config-pmap-c)# 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)# exit

Device(config)# interface dialer 0
 Device(config-if)# service-policy output parent
```

例：ダイヤラ インターフェイスの PPPoEoA 用の QoS の設定

```
Device(config)# interface ATM 0
 Device(config-if)# no ip address
 Device(config-if)# no atm ilmi-keepalive
 Device(config-if)# exit

Device(config)# interface ATM 0.1 point-to-point
 Device(config-if)# ip address 192.168.0.0 255.255.255.224
 Device(config-if)# pvc 4/46
  Device(config-if-atm-vc)# vbr-nrt 738 738
  Device(config-if-atm-vc)# pppoe-client dial-pool-number 1
  Device(config-if-atm-vc)# exit
 Device(config-if)# exit

Device(config)# interface Dialer 0
 Device(config-if)# mtu 1200
 Device(config-if)# ip address 172.16.0.0 255.0.0.0
 Device(config-if)# encapsulation ppp
 Device(config-if)# dialer pool 1
 Device(config-if)# dialer-group 1
 Device(config-if)# service-policy output dialer-output-sp
!
Device(config)# dialer-list 1 protocol ip permit
```

例：ダイヤラ インターフェイス上の PPPoE 用の QoS の設定

```
Device(config)# interface ethernet 0/0
 Device(config-if)# pppoe enable group global
 Device(config-if)# pppoe-client dial-pool-number 1
 Device(config-if)# exit

Device(config)# interface Dialer 0
 Device(config-if)# mtu 1200
 Device(config-if)# ip address 172.16.0.0 255.0.0.0
 Device(config-if)# encapsulation ppp
 Device(config-if)# dialer pool 1
 Device(config-if)# dialer-group 1
 Device(config-if)# service-policy output dialer-output-sp
 Device(config-if)# exit

Device(config)# dialer-list 1 protocol ip permit
```

例：ダイヤラ インターフェイス上の PPPoA 用の QoS の設定

```
Device(config)# interface ATM 0.1 point-to-point
 Device(config-if)# ip address 192.168.0.0 255.255.255.224
 Device(config-if)# pvc 4/46
  Device(config-if-atm-vc)# vbr-nrt 738 738
  Device(config-if-atm-vc)# dialer pool-member 1
  Device(config-if-atm-vc)# protocol ppp dialer
  Device(config-if-atm-vc)# exit
 Device(config-if)# exit

Device(config)# interface Dialer 0
 Device(config-if)# mtu 1200
 Device(config-if)# ip address 172.16.0.0 255.0.0.0
 Device(config-if)# encapsulation ppp
 Device(config-if)# dialer pool 1
 Device(config-if)# dialer-group 1
 Device(config-if)# service-policy output dialer-output-sp
 Device(config-if)# exit

Device(config)# dialer-list 1 protocol ip permit
```

例：ダイヤラ インターフェイス上の複数セッション用の QoS の設定

```
Device(config)# interface ethernet 0/0
 Device(config-if)# pppoe enable group global
 Device(config-if)# pppoe-client dial-pool-number 1
 Device(config-if)# pppoe-client dial-pool-number 2
 Device(config-if)# pppoe-client dial-pool-number 3
 Device(config-if)# exit

Device(config)# interface Dialer 0
 Device(config-if)# dialer pool 1
 Device(config-if)# service-policy output dialer-output-sp
 Device(config-if)# exit

Device(config)# interface Dialer 1
 Device(config-if)# dialer pool 2
 Device(config-if)# service-policy output dialer-output-sp
 Device(config-if)# exit
```

```
Device(config)# interface Dialer 2
 Device(config-if)# dialer pool 3
 Device(config-if)# service-policy output dialer-output-sp
 Device(config-if)# exit
```

例：ダイヤラ インターフェイスへの CoS 値の適用

```
Device> enable
Device# configure terminal
Device(config)# policy-map output_cos
Device(config-pmap)# class class-default
Device(config-pmap-c)# set cos 1
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface Dialer 1
Device(config-if)# service-policy output output-cos
Device(config-if)# exit
Device(config)# interface Ethernet 0.10
Device(config-subif)# encapsulation dot1q 10
Device(config-subif)# pppoe-client dial-pool-number 1
Device(config-subif)# exit
```

ダイヤラインターフェイス上のシェーピングに関する追加の参考資料

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	『*Cisco IOS Quality of Service Solutions Command Reference*』
MQC	『*QoS: Modular QoS: Command-Line Interface Configuration Guide*』

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★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

ダイヤラインターフェイス上のシェーピングに関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、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 リリースだけを示しています。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。

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。

表 *9*：ダイヤラ インターフェイス上のシェーピングに関する機能情報

機能名	リリース	機能情報
ダイヤラ インターフェイス上のシェーピング	15.3(1)T Cisco IOS XE リリース 3.13S	ダイヤラ インターフェイス上のシェーピング機能では、ダイヤラ インターフェイスでの PPPoE/A 設定に対するサポートが提供されます。