Oracle® Solaris 11.2 での IP サービス品質の管理

Part No: E53876
2014 年 7 月

ORACLE

目次

このドキュメントの使用法 7

1 IPQoS の概要 9
IPQoS の基本 9
差別化サービスとは 10
IPQoS の機能 10
詳細情報の入手先 10
IPQoS によるサービス品質の提供 11
サービスレベル契約の実装 11
一般の組織にとってのサービス品質の保証 12
サービス品質ポリシーの紹介 12
IPQoS によるネットワーク効率の向上 13
ネットワークトラフィックへの帯域幅の影響 13
サービスクラスを使ったトラフィックの優先順位付け 13
差別化サービスモデル 16
クラシファイア (`ipgpc`) の概要 16
メーター (`tokenmt` および `tswtclmt`) の概要 17
マーカー (`dscpmk` および `dlcosmk`) の概要 18
フローアカウンティング (`flowacct`) の概要 19
トラフィックが IPQoS モジュールをどのように通過するか 19
IPQoS 対応ネットワークでのトラフィック転送 21
DS コードポイント 21
ホップ単位動作 21

2 IPQoS 対応ネットワークの計画 25
一般的な IPQoS の構成計画のタスクマップ 25
diffserv ネットワークトポロジの計画 26
diffserv ネットワークのハードウェア計画 26
IPQoS ネットワークトポロジ 27
サービス品質ポリシーの計画 29

QoS ポリシー計画の手掛かり 29
QoS ポリシーの計画のタスクマップ 30
IPQOS のネットワークの準備 31
QoS ポリシーのクラスの定義 31
フィルタの定義 32
▼ QoS ポリシーにフィルタを定義する方法 33
フロー制御の計画 34
転送動作の計画 37
▼ 転送動作を計画する方法 37
フローアカウンティングの計画 39
IPQoS の構成例の紹介 39
IPQoS トポロジ 40

3 IPQoS 構成ファイルの作成のタスク 43
QoS ポリシーの定義のタスクマップ 43
QoS ポリシー作成用のツール 44
基本 IPQoS 構成ファイル 45
Web サーバー用 IPQoS 構成ファイルの作成 45
▼ IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法 48
▼ IPQoS 構成ファイル内でフィルタを定義する方法 50
▼ IPQoS 構成ファイル内でトラフィック転送を定義する方法 52
▼ IPQoS 構成ファイル内でクラスのアカウンティングを有効にする方法 56
▼ ベストエフォート Web サーバー用の IPQoS 構成ファイルを作成する方法 58
アプリケーションサーバー用 IPQoS 構成ファイルの作成 61
▼ アプリケーションサーバー用 IPQoS 構成ファイルを作成する方法 63
▼ IPQoS 構成ファイル内でアプリケーショントラフィックの転送を構成する方法 66
▼ IPQoS 構成ファイル内でフロー制御を構成する方法 68
ルーター上での差別化サービスの提供 71

4 IPQoS の起動と保守のタスク 73
IPQoS の管理 73
▼ `ipqos` パッケージを追加する方法 74
▼ `ipqos` サービスを開始する方法 74
▼ ブート時に IPQoS メッセージを記録する方法 75
IPQoS エラーメッセージのトラブルシューティング 76

5 フローアカウンティングの使用と統計情報の収集のタスク 81

トラフィックフローに関する情報の記録 81
▼ フローアカウンティングデータ用のファイルを作成する方法 82
統計情報の収集 84

6 IPQoS の詳細のリファレンス 87
IPQoS アーキテクチャーと Diffserv モデル 87
クラシファイアモジュール 88
メーターモジュール 90
マーカーモジュール 93
`flowacct` モジュール 98
IPQoS 構成ファイル 101
`action` 文 102
モジュール定義 103
`class` 句 103
`filter` 句 104
`params` 句 104

索引 105

このドキュメントの使用法

- **概要** – IPQos サービスの構成方法について説明します。
- **対象読者** – 技術者、システム管理者、および認定サービスプロバイダ
- **前提知識** –Oracle Solaris の操作経験

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (http://www.oracle.com/pls/topic/lookup?ctx=E56342) に含まれています。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info を参照してください。聴覚に障害をお持ちの場合は、http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs を参照してください。

フィードバック

このドキュメントに関するフィードバックを http://www.oracle.com/goto/docfeedback からお聞かせください。

♦ ♦ ♦ 第 1 章

1 IPQoS の概要

IP サービス品質 (IPQoS) を使用すると、優先順位付け、管理、およびアカウンティング統計情報の収集を行うことができます。IPQoS によって、ネットワークのユーザーに一貫したサービスレベルを提供できます。また、ネットワークの輻輳を防ぐために、トラフィックを管理することもできます。

この章で扱う内容は、次のとおりです。

- 9 ページの「IPQoS の基本」
- 11 ページの「IPQoS によるサービス品質の提供」
- 13 ページの「IPQoS によるネットワーク効率の向上」
- 16 ページの「差別化サービスモデル」
- 21 ページの「IPQoS 対応ネットワークでのトラフィック転送」

注記 - IPQoS 機能は、将来のリリースで削除される可能性があります。代わりに、同様の帯域幅リソース制御機能をサポートしている、`dladm`、`flowadm`、および関連コマンドを使用するようにしてください。詳細は、『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理 』を参照してください。

IPQoS の基本

IPQoS は、Internet Engineering Task Force (IETF) の Differentiated Services Working Group によって定義されている差別化サービス (Diffserv) アーキテクチャーに対応しています。Oracle Solaris では、IPQoS は TCP/IP プロトコルスタックの IP レベルで実装されます。

差別化サービスとは

IPQoS を有効にすると、選択した顧客や選択したアプリケーションにさまざまなレベルのネットワークサービスを提供できます。この異なるレベルのサービスは、まとめて「*差別化サービス*」と呼ばれます。顧客に提供する差別化サービスは、ユーザーの企業が顧客に提供するサービスレベルの構造を基に決定できます。ネットワーク上のアプリケーションやユーザーに設定した優先順位に基づく場合もあります。

サービス品質 (QoS) を提供には、次のタスクを行います。

- 顧客や企業内の部署などのグループごとに異なるサービスレベルを提供する
- 特定のグループやアプリケーションにネットワークサービスを優先的に提供する
- ネットワーク上の障害やその他の輻輳の発生箇所を特定し、問題を取り除く
- ネットワークパフォーマンスをモニタリングし、パフォーマンスの統計情報を提供する
- ネットワークリソースに対する帯域幅を調整する

IPQoS の機能

IPQoS には次の機能があります。

- アクションを選択するクラシファイア。アクションは、組織の QoS ポリシーを構成するフィルタに基づいている
- Diffserv モデルに従ってネットワークトラフィックを測定するメータリングモジュール
- パケットの IP ヘッダーに転送情報を付ける機能に基づく、サービスの差別化
- トラフィックフローの統計情報を収集するフローアカウンティングモジュール
- UNIX の `kstat` コマンドによる、トラフィッククラスごとの統計情報の収集
- SPARC および x86 アーキテクチャーのサポート
- IPv4 および IPv6 アドレス指定のサポート
- IP セキュリティーアーキテクチャーの相互運用性 (IPsec)
- 仮想ローカルエリアネットワーク (VLAN) の 802.1D ユーザー優先順位マークのサポート

詳細情報の入手先

差別化サービスやサービス品質の詳細情報は、印刷物やオンラインで入手できます。

IPQoS は、次の RFC の仕様に準拠しています。

- RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (http://www.ietf.org/rfc/rfc2474.txt?number=2474) – 差別化サービスをサポートするための、IPv4 や IPv6 パケットヘッダーのサービスタイプ (ToS) フィールドまたは DS フィールドの拡張を説明している
- RFC 2475, An Architecture for Differentiated Services (http://www.ietf.org/rfc/rfc2475.txt?number=2475) – Diffserv アーキテクチャーの構成とモジュールについて詳細に解説している

IPQoS のドキュメントには、次のマニュアルページが含まれます。

- `ipqosconf`(1M) - IPQoS 構成ファイルを設定するコマンドについて説明する
- `ipqos`(7ipp) – Diffserv アーキテクチャーモデルの IPQoS 実装について説明する
- `ipgpc`(7ipp) – Diffserv クラシファイアの IPQoS 実装について説明する
- `tokenmt`(7ipp) – IPQoS の `tokenmt` メーターについて説明する
- `tswtclmt`(7ipp) – IPQoS の `tswtclmt` メーターについて説明する
- `dscpmk`(7ipp) – DSCP マーカーモジュールについて説明する
- `dlcosmk`(7ipp) – IPQoS 802.1D ユーザー優先順位マーカーモジュールについて説明する
- `flowacct`(7ipp) – IPQoS フローアカウンティングモジュールについて説明する
- `acctadm`(1M) – Oracle Solaris 拡張アカウンティング機能を構成し、IPQoS 拡張を含むコマンドについて説明する。

IPQoS によるサービス品質の提供

IPQoS 機能を使用すると、インターネットサービスプロバイダ (ISP) やアプリケーションサービスプロバイダ (ASP) は、顧客ごとに異なるレベルのネットワークサービスを提供できます。同様に、一般企業や教育機関では、この機能を使って内部組織向けのサービスや主要なアプリケーションのサービスを優先できます。

サービスレベル契約の実装

ISP や ASP では、顧客に提示する「*サービスレベル契約* (Service-Level Agreement、SLA)」に基づいて IPQoS の構成を行うことができます。個々の SLA では、サービスプロバイダは、価

格体系に基づいた一定レベルのネットワークサービスを顧客に保証します。たとえば、プレミアム価格の SLA では、顧客はすべての種類のネットワークトラフィックに対して毎日 24 時間もっとも高い優先順位が与えられます。逆に、中ぐらいの価格の SLA では、業務時間に電子メールだけに高い優先順位が保証されている場合もあります。ほかのトラフィックはすべて、一日 24 時間、中ぐらいの優先順位になります。

一般の組織にとってのサービス品質の保証

一般企業や法人である場合でも、ネットワークにサービス品質機能を提供できます。つまり、特定のグループまたは特定のアプリケーションのトラフィックに対して高レベルまたは低レベルのサービスを保証できます。

サービス品質ポリシーの紹介

サービス品質を実装するには、「*サービス品質 (Quality-of-Service、QoS) ポリシー*」を定義します。QoS ポリシーでは、顧客またはアプリケーションの優先順位、さまざまなカテゴリのトラフィックを処理するアクションなど、各種のネットワーク属性を定義します。組織の QoS ポリシーは IPQoS 構成ファイルに実装します。このファイルは、Oracle Solaris のカーネルに入っている IPQoS モジュールを構成します。IPQoS ポリシーが適用されているホストは、「*IPQoS 対応システム*」とみなされます。

QoS ポリシーは、一般に次のことを定義します。

- 「*サービスクラス*」と呼ばれるネットワークトラフィックの個別グループ
- クラスごとにネットワークトラフィックの量を調整するための測定基準。これらの測定基準によって、「*メータリング*」と呼ばれるトラフィック測定プロセスが管理される。
- IPQoS システムおよび Diffserv ルーターがパケットフローに適用するアクション。このアクションは「*ホップ単位動作* (PHB)」と呼ばれる
- サービスのクラスで必要な統計の収集。たとえば、顧客または特定のアプリケーションが生成したトラフィックなどがある。

パケットがネットワークに渡されると、IPQoS 対応システムはパケットヘッダーを評価します。IPQoS システムが行うアクションは、作成した QoS ポリシーに応じて決まります。

QoS ポリシーの設計タスクについては、29 ページの「サービス品質ポリシーの計画」に説明があります。

IPQoS によるネットワーク効率の向上

IPQoS には、サービス品質の実装に伴ってネットワークパフォーマンスの効率を向上させるのに役立つ機能が含まれています。たとえば、企業や法人の場合は、効率的なネットワークを維持して、トラフィックに関する障害の発生を防ぐ必要があります。また、グループやアプリケーションが割り当てられた以上の帯域幅を消費しないようにする必要もあります。ISP や ASP は、顧客が料金分のレベルのネットワークサービスを確実に受けられるようにネットワークパフォーマンスを管理する必要があります。

超過したネットワークの症状には、データの損失やトラフィックの輻輳などがあります。どちらの症状も応答時間を遅らせます。以前は、システム管理者は、帯域幅、つまり、ネットワークリンクまたはデバイスが完全に使用された場合に転送できるデータの最大量を追加することで、ネットワークトラフィックの問題を処理していました。ただし、リンクごとのトラフィックのレベルには、大きなばらつきが見られがちでした。IPQoS を使用すると、既存のネットワーク上のトラフィックを管理しながら、ネットワークの拡大が必要かどうか、またどこに必要かを評価できます。

ネットワークトラフィックへの帯域幅の影響

サービス品質を顧客またはユーザーに提供するには、QoS ポリシーで、帯域幅の使用に優先順位を付ける必要があります。IPQoS のメータリングモジュールを使用すると、IPQoS 対応ホスト上の各トラフィッククラスへの帯域幅の割り当て量を測定および管理できます。

ネットワークのトラフィックを効率的に管理する方法を判断する場合は、次の質問を考慮します。

- ローカルネットワークのトラフィック問題の発生箇所はどこか
- 帯域幅を最適利用するために行うべきことは何か
- サイト内で最優先するべき、重要なアプリケーションはどれか
- 輻輳が発生しやすいアプリケーションはどれか
- 優先順位を下げてもよい、重要度の低いアプリケーションはどれか

サービスクラスを使ったトラフィックの優先順位付け

サービス品質を実現するには、ネットワークトラフィックを分析して、トラフィックを分類する大まかなグループ分けを決定します。次に、それぞれ特徴と優先順位を持つサービスクラスに各グ

ループ分けを整理します。サービスのクラスは、組織の QoS ポリシーの基準となる基本カテゴリを形成し、制御する必要のあるトラフィックグループを表します。

ネットワークトラフィックを分析する場合は、次のガイドラインを考慮します。

- **顧客にサービスレベル契約を提示しているか**

  提示する場合は、顧客に提供する複数の SLA 間の相対的な優先レベルを評価します。保証されている優先レベルが異なる顧客に同じアプリケーションを提供する場合もあります。

  たとえば、企業が各顧客に対して Web サイトの運営サービスを提供するとします。この場合は、顧客の Web サイトごとに 1 つのクラスを定義する必要があります。SLA は、サービスレベルの 1 つとしてプレミアム Web サイトを提供するとします。別の SLA は、割引顧客に「ベストエフォート型」のパーソナル Web サイトを提供するとします。この場合は、Web サイトのクラスが異なるだけでなく、クラスに割り当てられる PHB も異なる可能性があります。

- **IPQoS システムでは、フロー制御を必要としそうなよく使われるアプリケーションを提供しているか**

  よく使われるアプリケーションを提供しているために大量のトラフィックが生成されるサーバーの場合、IPQoS を有効にするとネットワークパフォーマンスが向上します。そのようなアプリケーションの例として、電子メール、ネットワークニュース、FTP などがあげられます。該当する場合は、サービスの種類ごとに着信トラフィックと発信トラフィックのクラスを別々に作成することを検討してください。たとえば、メールサーバーの QoS ポリシーに対して、`mail-in` クラスと `mail-out` クラスを作成します。

- **ネットワークでもっとも高い優先順位の転送動作を必要とする特定のアプリケーションを実行しているか**

  優先順位がもっとも高い転送動作を必要とする重要なアプリケーションには、ルーターのキューでもっとも高い優先順位を与える必要があります。一般的な例は、ストリーミングビデオやストリーミングオーディオです。

  まず、優先順位の高いこれらのアプリケーションに対して、それぞれ着信クラスと発信クラスを定義します。次に、定義したクラスを、それらのアプリケーションを提供する IPQoS 対応システムと Diffserv ルーターの両方の QoS ポリシーに追加します。

- **帯域幅を大量に消費するため、ネットワークで制御を必要とするトラフィックフローが発生したことがあるか**

  `netstat`、`snoop` などのネットワークモニタリングユーティリティーを使用して、ネットワーク上で問題のあるトラフィックの種類を検出します。これまでに作成したクラスを確認し、問題のトラフィックカテゴリが未定義である場合は、このカテゴリに対して新しいクラスを作成しま

す。問題のトラフィックカテゴリのクラスが定義済みである場合は、このトラフィックを制御するメーターの速度を定義します。

問題のあるトラフィックのクラスは、ネットワーク上の IPQoS 対応システムごとに作成します。これによって、各 IPQoS システムは、問題のあるトラフィックを受け取った場合に、トラフィックフローをネットワークに送出する速度を制限できるようになります。また、Diffserv ルーターの QoS ポリシーにもこれらの問題のあるクラスを必ず定義してください。これによって、diffserv ルーターは、QoS ポリシーの構成に従って、問題のあるフローをキューに入れたりスケジュールしたりできるようになります。

- **特定の種類のトラフィックに対して統計情報を取得する必要があるか**

  SLA をざっと確認すると、どのタイプの顧客のトラフィックにアカウンティングが必要であるかがわかります。自分のサイトで SLA を提供している場合は、アカウンティングを必要とするトラフィックのクラスはおそらく作成済みです。また、クラスを定義して、モニタリングしているトラフィックフローの統計収集を可能にすることもできます。さらに、セキュリティー上の理由でアクセスを制限するトラフィックのクラスも作成できます。

たとえば、プロバイダがプラチナ、ゴールド、シルバー、ブロンズの各レベルのサービスをそれぞれ異なる利用料金で提供するとします。プラチナレベルの SLA では、プロバイダが顧客用に運営している Web サイト宛ての着信トラフィックに対して、もっとも高い優先順位を保証します。

企業では、次の例のようなサービスのクラスを作成できます。

- 特定のサーバー宛ての電子メールや発信 FTP などの、よく使われるアプリケーション。アプリケーションごとに 1 つのクラスを構成できます。これらのアプリケーションは従業員によって絶えず使用されるため、QoS ポリシーで、電子メールと発信 FTP には少量の帯域幅と低い優先順位を割り当てます。
- 一日 24 時間実行の必要がある注文入力データベース。企業にとってのデータベースアプリケーションの重要度に応じて、大量の帯域幅と高い優先順位を割り当てます。
- 人事部門などの、極めて重要な業務または機密業務を行う部署。組織にとっての部署の重要度に応じて、割り当てる優先順位と帯域幅の大きさを決めます。
- 企業の外部向け Web サイトへの呼び出し。このクラスには、適度な大きさの帯域幅と低い優先順位を割り当てる

差別化サービスモデル

IPQoS には、RFC 2475 に定義されている Diffserv アーキテクチャーの一部である次のモジュールがあります。

- クラシファイア
- メーター
- マーカー

IPQoS では、次の拡張機能が Diffserv モデルに追加されています。

- フローアカウンティングモジュール
- 802.1D データグラムマーカー

このセクションでは、IPQoS で使用する Diffserv モジュールについて簡単に説明します。各モジュールの詳細は、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」を参照してください。

クラシファイア (ipgpc) の概要

Diffserv モデルでは、「*クラシファイア*」がネットワークトラフィックフローからパケットを選択します。「*トラフィックフロー*」は、次の IP ヘッダーフィールド内に同一の情報を持つパケットのグループで構成されます。

- 発信元アドレス
- 着信先アドレス
- 発信元ポート
- 着信先ポート
- プロトコル番号

IPQoS では、これらのフィールドを「*5 タプル*」と呼びます。

IPQoS クラシファイアモジュール `ipgpc` は、トラフィックフローを、IPQoS 構成ファイルに構成されている特性に基づいたクラスに分類します。

`ipgpc` の詳細については、88 ページの「クラシファイアモジュール」を参照してください。

IPQoS クラス

トラフィックをいくつかのクラスに分類することは、QoS ポリシーを計画する際に欠かせない作業の 1 つです。`ipqosconf` ユーティリティーを使用してクラスを作成するときは、実際には`ipgpc` クラシファイアを構成しています。

クラスを定義する方法については、31 ページの「QoS ポリシーのクラスの定義」を参照してください。

IPQoS フィルタ

「*フィルタ*」は、「*セレクタ*」と呼ばれるパラメータを含む規則のセットです。各フィルタは、必ず 1 つのクラスを指定する必要があります。IPQoS は、パケットを各フィルタのセレクタと突き合わせて、パケットがフィルタのクラスに属しているかどうかを調べます。さまざまなセレクタを使用してパケットにフィルタをかけることができます。セレクタの例として、IPQoS 5 タプルなどのよく使うパラメータを次に示します。

- 発信元および着信先のアドレス
- 発信先および着信先のポート
- プロトコル番号
- ユーザー ID
- プロジェクト ID
- 差別化サービスコードポイント (DSCP)
- インタフェースインデックス

たとえば、簡単なフィルタに値が 80 の宛先ポートが含まれているとします。`ipgpc` クラシファイアは、宛先ポート 80 (HTTP) 向けのパケットをすべて選択し、QoS ポリシーの指示どおりに選択したパケットを処理します。

フィルタの作成については、33 ページの「QoS ポリシーにフィルタを定義する方法」を参照してください。

メーター (`tokenmt` および `tswtclmt`) の概要

Diffserv モデルでは、「*メーター*」はトラフィックフローの転送速度をクラス単位で追跡します。メーターは、該当する結果 (outcome) を得るために、フローの実際の転送速度が構成された

速度にどれだけ適合しているかを評価します。トラフィックフローの結果に基づいて、メーターは、別のアクションにパケットを送信したり、追加処理なしでネットワークにパケットを戻したりするなど、後続のアクションを選択します。

IPQoS のメーターは、ネットワークフローが、QoS ポリシーでそのクラスに定義されている転送速度に適合しているかどうかを調べます。IPQoS には、次の 2 つのメータリングモジュールがあります。

- `tokenmt` - 2 トークンバケットメータリングスキームを使用
- `tswtclmt` - タイムスライディングウィンドウメータリングスキームを使用

どちらのメータリングモジュールも、赤、黄、緑という 3 つの結果を識別します。結果ごとに実行させたいアクションは、`red_action_name`、`yellow_action_name`、および `green_action_name` のパラメータで定義します。

また、`tokenmt` をカラーアウェアとして構成することもできます。カラーアウェアとして構成されているメータリングインスタンスでは、パケットのサイズ、DSCP、トラフィックの転送速度、および構成されたパラメータを使って結果を求めます。メーターは、DSCP を使用してパケットの結果を緑、黄、赤にマッピングします。

IPQoS メーターのパラメータの定義については、34 ページの「フロー制御を計画する方法」を参照してください。

マーカー (`dscpmk` および `dlcosmk`) の概要

Diffserv モデルでは、「*マーカー*」は転送動作を表す値をパケットに付けます。「*マーキング*」とは、パケットをネットワークに転送する方法を示す値を、そのパケットのヘッダーに付加するプロセスのことです。

IPQoS には、次の 2 つのマーカーモジュールが含まれています。

- `dscpmk` – IP パケットヘッダーの DS フィールドに*差別化サービスコードポイント* (*DSCP*) と呼ばれる数値を付けます。Diffserv 対応ルーターは、この DS コードポイントを使って、適切な転送動作をパケットに適用できます。
- `dlcosmk` – Ethernet フレームヘッダーの仮想ローカルエリアネットワーク (VLAN) タグに*ユーザー優先順位*と呼ばれる数値を付けます。ユーザー優先順位は、データグラムに適用される適切な転送動作を定義する「*サービスクラス (CoS)*」のことです。

  `dlcosmk` は、IETF Diffserv モデルの一部ではない IPQoS の追加機能です。

QoS ポリシーのマーカー戦略の実装については、37 ページの「転送動作の計画」を参照してください。

フローアカウンティング (flowacct) の概要

IPQoS では、flowacct アカウンティングモジュールが Diffserv モデルに追加されます。flowacct を使用すると、トラフィックフローに関する統計情報を取得し、SLA に合わせて顧客に課金できます。フローアカウンティングは、容量計画やシステムのモニタリングにも役立ちます。

flowacct モジュールを acctadm コマンドと組み合わせて、アカウンティングログファイルを作成します。基本的なログには、IPQoS 5 タプルのほかに 2 つの属性が記録されます。

- 発信元アドレス
- 発信元ポート
- 着信先アドレス
- 着信先ポート
- プロトコル番号
- パケット数
- バイト数

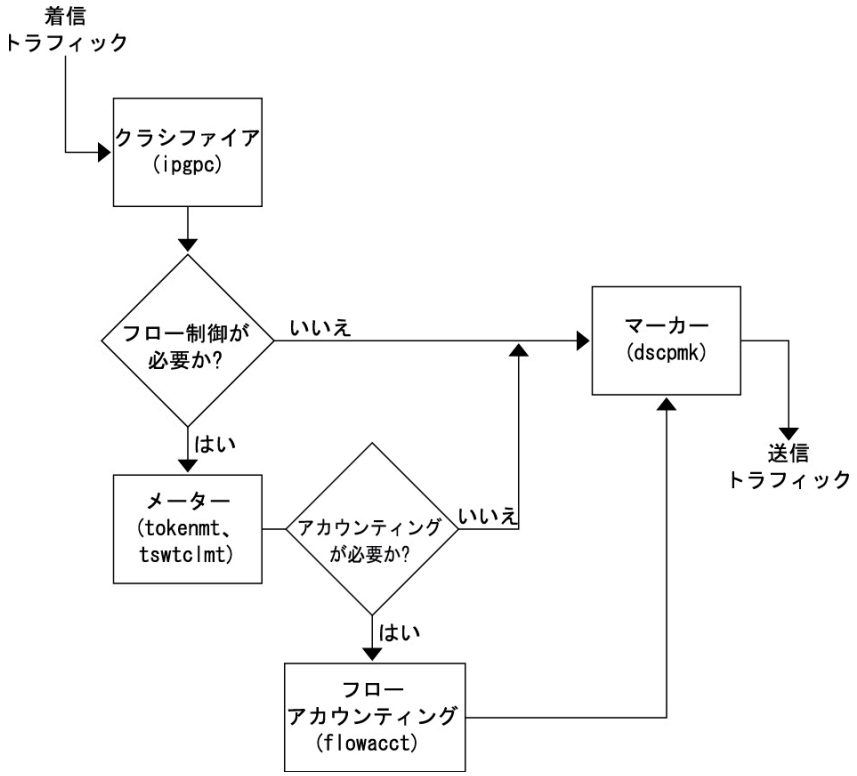
81 ページの「トラフィックフローに関する情報の記録」および flowacct(7ipp) や acctadm(1M) のマニュアルページに説明されているとおり、その他の属性の統計を集めることもできます。

フローアカウンティング戦略の計画については、39 ページの「フローアカウンティングの計画」を参照してください。

トラフィックが IPQoS モジュールをどのように通過するか

次の図は、着信トラフィックが IPQoS モジュールのいくつかを通過するときに取りうる経路を示しています。

図 1-1 diffserv モデルの IPQoS 実装を通過するトラフィックフロー



この図は、IPQoS 対応マシンにおける一般的なトラフィックフローシーケンスを示しています。

1. クラシファイアが、システムの QoS ポリシーのフィルタリング条件に適合するすべてのパケットをパケットストリームから選択します。
2. 選択したパケットが評価されて、次に実行されるアクションが決められます。
3. クラシファイアが、フロー制御を必要としないトラフィックをマーカーに送信します。
4. フロー制御が必要なトラフィックは、メーターに送信されます。
5. メーターは、構成速度を実施します。次に、トラフィックの適合値をフロー制御されているパケットに割り当てます。
6. フロー制御されるパケットが評価されて、アカウンティングが必要であるかどうかが判断されます。
7. メーターが、フローアカウンティングを必要としないトラフィックをマーカーに送信します。

8. フローアカウンティングモジュールが、受信したパケットに関する統計情報を収集します。次に、それらのパケットをマーカーに送信します。
9. マーカーが DS コードポイントをパケットヘッダーに割り当てます。この DSCP は、Diffserv 対応システムがパケットに適用するべきホップ単位動作 (PHB) を示します。

IPQoS 対応ネットワークでのトラフィック転送

このセクションでは、IPQoS 対応ネットワークでのパケット転送に関係するいくつかの要素について簡単に説明します。IPQoS 対応システムは、着信先としてそのシステムの IP アドレスを持つ、ネットワークストリーム上のパケットを処理します。そして、IPQoS システムの QoS ポリシーをパケットに適用して、差別化サービスを確立します。

DS コードポイント

DS コードポイント (DSCP) は、マークされたパケットに対して Diffserv 対応システムが実行するアクションをパケットヘッダーに定義します。Diffserv アーキテクチャーは、使用する IPQoS 対応システムと Diffserv ルーターに対して一連の DS コードポイントを定義します。Diffserv アーキテクチャーでは、DSCP に対応する*転送動作*も定義します。IPQoS 対応システムは、パケットヘッダーにある DS フィールドの優先度ビットに DSCP を付けます。DSCP 値を持つパケットを受信すると、ルーターは、その DSCP と関連付けられた転送動作を実行します。次にパケットはネットワーク上に送出されます。

注記 - `dlcosmk` マーカーは、DSCP を使用しません。代わりに、`dlcosmk` は Ethernet フレームヘッダーに CoS 値を付加します。VLAN デバイスを使用するネットワークで IPQoS を構成する予定の場合は、93 ページの「マーカーモジュール」を参照してください。

ホップ単位動作

Diffserv 用語では、DSCP に割り当てられる転送動作を*ホップ単位動作 (Per-Hop Behavior、PHB)* と呼びます。PHB は、Diffserv 対応システム上で、マークされたパケットの転送がほかのトラフィックに比べて優先される度合いを定義します。この優先度によって、IPQoS 対応システムまたは Diffserv ルーターが、マークされたパケットを転送するかドロップするかが最終的に決まります。パケットが転送された場合、パケットがその着信先への途中で通過する各 Diffserv ルーターは、別の Diffserv システムが DSCP を変更していないかぎり、同じ

PHB をそのパケットに適用します。PHB の詳細については、93 ページの「パケット転送での dscpmk マーカーの使用」を参照してください。

PHB の目的は、指定された量のネットワークリソースを連続したネットワーク上のトラフィッククラスに提供することです。QoS ポリシーで、トラフィックフローが IPQoS 対応システムを離れたときのトラフィッククラスの優先順位を示す DSCP を定義します。優先度は、高い優先度 (ドロップ率が低い) から低い優先度 (ドロップ率が高い) の範囲になります。

たとえば、QoS ポリシーでは、ドロップ率が低い優先度の PHB をすべての Diffserv 対応ルーターから与えられ、このクラスのパケットの帯域幅が保証される DSCP をトラフィックの 1 つのクラスに割り当てることができます。QoS ポリシーに別の DSCP を追加して、ほかのトラフィッククラスにさまざまなレベルの優先度を割り当てることもできます。優先度の低いパケットには、パケットの DSCP に示された優先順位に応じた帯域幅を、Diffserv システムが割り当てます。

IPQoS は、Diffserv アーキテクチャーに定義されている 2 種類の転送動作、完全優先転送 (EF) と相対的優先転送 (AF) をサポートしています。

- 完全優先転送

  このホップ単位動作は、EF 関連の DSCP を持つトラフィッククラスの優先順位がいちばん高いことを前提とします。EF DSCP のトラフィックは、キューに格納されません。EF では、低損失、低遅延、低ジッターのサービスを提供します。EF の推奨 DSCP は、101110 です。101110 が付加されたパケットは、着信先への途中で Diffserv 対応ネットワークを通過するときに、ドロップ率の低い優先度を与えられます。EF DSCP は、プレミアム SLA を持つ顧客またはアプリケーションに優先順位を割り当てるときに使用してください。

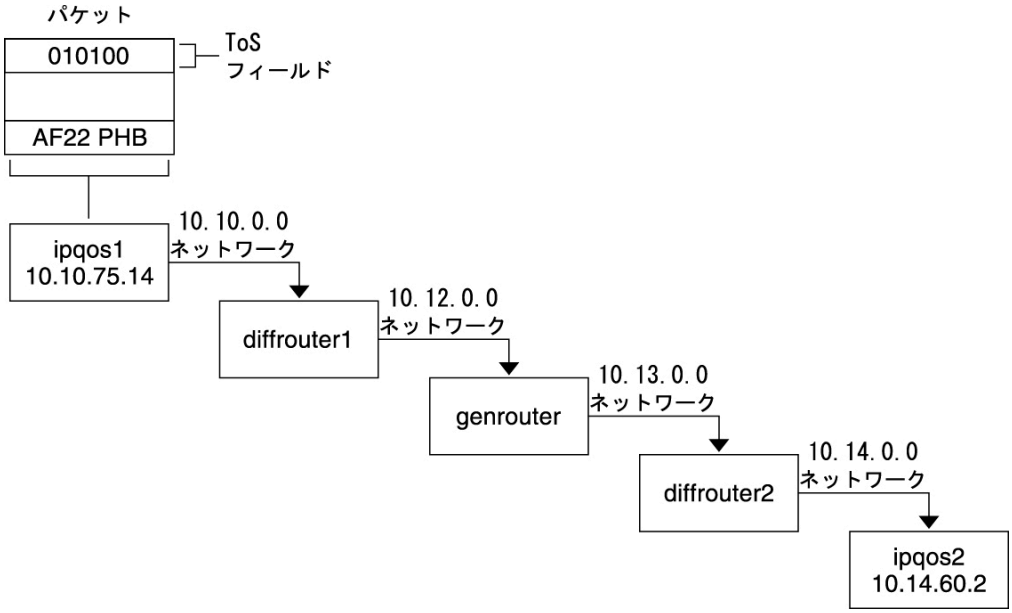
- 相対的優先転送

  このホップ単位動作には、パケットに割り当てられる転送クラスが 4 種類あります。すべての転送クラスは、低ドロップ、中ドロップ、および高ドロップの 3 つのドロップ優先度を提供します。詳細は、表6-2「相対的優先転送のコードポイント」を参照してください。AF コードポイントには、顧客やアプリケーションにさまざまなレベルのサービスを割り当てる機能があります。

diffserv 環境でのパケット転送

次の図は、Diffserv 対応環境を部分的に備えた企業のイントラネット部分を示しています。このシナリオでは、ネットワーク `10.10.0.0` および `10.14.0.0` 上のホストはすべて IPQoS に対応しており、ローカルルーターは Diffserv に対応しています。しかし、間にある 2 つのネットワークは Diffserv 用に構成されていません。

図 1-2 diffserv 対応ネットワークのホップ間のパケット転送



この図のパケットのフローは、ホスト `ipqos1` に起因するパケットの進行で始まります。手順は数回のホップでホスト `ipqos2` に続きます。

1. `ipqos1` のユーザーが `ftp` コマンドを実行して、3 ホップ離れたところにあるホスト `ipqos2` にアクセスしようとします。
2. `ipqos1` は、結果生じたパケットフローに QoS ポリシーを適用します。`ipqos1` は、`ftp` トラフィックを正常に分類します。

   システム管理者は、ローカルネットワーク 10.10.0.0 に起因するすべての発信 `ftp` トラフィックに対するクラスを作成済みです。`ftp` クラスのトラフィックには、AF22 ホップ単位動作 (クラス 2、中程度のドロップ優先度) が割り当てられます。`ftp` クラスには、2Mb/秒のトラフィックフロー速度が構成されます。
3. `ipqos-1` は、`ftp` フローを測定し、フローが 2M ビット/秒の認定速度を超過していないかどうかを判断します。
4. `ipqos1` 上のマーカーが、発信 `ftp` パケットの DS フィールドに 010100 DSCP (AF22 PHB に対応している) を付加します。

5. ルーター diffrouter1 は、ftp パケットを受信します。次に、DSCP をチェックします。diffrouter1 が輻輳している場合、AF22 でマークされているパケットは振り落とされます。
6. diffrouter1 のファイルで AF22 に対して構成されている PHB に合わせて、ftp トラフィックが次のホップに転送されます。
7. ftp トラフィックがネットワーク 10.12.0.0 を通って genrouter に進み、genrouter は Diffserv に対応していません。その結果、トラフィックはベストエフォートの転送動作を与えられます。
8. genrouter が ftp トラフィックをネットワーク 10.13.0.0 に渡し、diffrouter2 がそのトラフィックを受け取ります。
9. diffrouter2 は Diffserv に対応しています。したがって、ルーターのポリシーで AF22 パケットに対して定義されている PHB に合わせて、ftp パケットをネットワークに転送します。
10. ipqos2 は、ftp トラフィックを受信し、ipqos2 は、ipqos1 のユーザーにユーザー名とパスワードの入力を促します。

♦ ♦ ♦ 第 2 章

2 IPQoS 対応ネットワークの計画

Oracle Solaris を実行するシステムで IPQoS を構成できます。IPQoS システムは、Diffserv 対応ルーターと連携し、差別化サービスとイントラネットのトラフィック管理を提供します。

この章では、IPQoS 対応システムを Diffserv 対応ネットワークに追加するための計画情報について説明します。

- 25 ページの「一般的な IPQoS の構成計画のタスクマップ」
- 26 ページの「diffserv ネットワークトポロジの計画」
- 29 ページの「サービス品質ポリシーの計画」
- 30 ページの「QoS ポリシーの計画のタスクマップ」
- 39 ページの「IPQoS の構成例の紹介」

注記 - IPQoS 機能は、将来のリリースで削除される可能性があります。代わりに、同様の帯域幅リソース制御機能をサポートしている、`dladm`、`flowadm`、および関連コマンドを使用するようにしてください。詳細は、『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理 』を参照してください。

一般的な IPQoS の構成計画のタスクマップ

IPQoS などの差別化サービスをネットワーク上に実装する場合は、綿密な計画が必要です。各 IPQoS 対応システムの位置と機能だけではなく、各システムとローカルネットワーク上のルーターとの関係についても考慮してください。次のタスクマップに、ネットワーク上で IPQoS を実装するための主な計画タスクの一覧と、各タスクを完了するための手順へのリンクを示します。

タスク	説明	手順の参照先
1. IPQoS 対応システムを取り入れた Diffserv ネットワークトポロジを計画する	さまざまな Diffserv ネットワークトポロジから選択して、自分のサイトに最適なソリューションを考える。	26 ページの「diffserv ネットワークトポロジの計画」

タスク	説明	手順の参照先
2. IPQoS システムによって提供する各種サービスを計画する	ネットワークが提供するサービスの種類をいくつかの サービスレベル契約 (service-level agreement、SLA) に分類する	29 ページの「サービス品質ポリシーの計画」
3. IPQoS システムごとに QoS ポリシーを計画する	各 SLA の実装に必要なクラス、メータリング、およびアカウンティング機能を決める	29 ページの「サービス品質ポリシーの計画」
4. 必要であれば、Diffserv ルーターのポリシーを計画する	IPQoS システムで使用する Diffserv ルーターのスケジューリングポリシーおよびキューイングポリシーを決める	キューイングポリシーおよびスケジューリングポリシーについては、ルーターのドキュメントを参照

diffserv ネットワークトポロジの計画

ネットワークに差別化サービスを提供するには、IPQoS 対応システムが少なくとも 1 つと Diffserv 対応ルーターが 1 台必要です。このセクションで説明するように、この基本構成はさまざまな方法で拡張できます。

diffserv ネットワークのハードウェア計画

一般に、IPQoS はサーバーやサーバー統合上で実行します。一方、ネットワークのニーズに応じて、デスクトップシステムで IPQoS を実行することもできます。

次に、IPQoS 構成に使用できるシステムの例を示します。

- Web サーバーやデータベースサーバーなどの各種サービスを提供する Oracle Solaris システム
- アプリケーションサーバー。電子メール、FTP などのよく使われるネットワークアプリケーションを提供する
- Web キャッシュサーバーまたはプロキシサーバー
- IPQoS 対応サーバーファームのネットワーク。Diffserv 対応ロードバランサによって管理される
- ファイアウォール。単一の異機種システム混在ネットワークのトラフィックを管理する
- 仮想ローカルエリアネットワーク (LAN) の一部である IPQoS システム

IPQoS システムを導入しようとするネットワークトポロジでは、Diffserv 対応ルーターがすでに機能していることもありえます。ローカルルーターが Diffserv を実装していないと、パケットのマークは評価されずに次のホップへ渡されてしまいます。

IPQoS ネットワークトポロジ

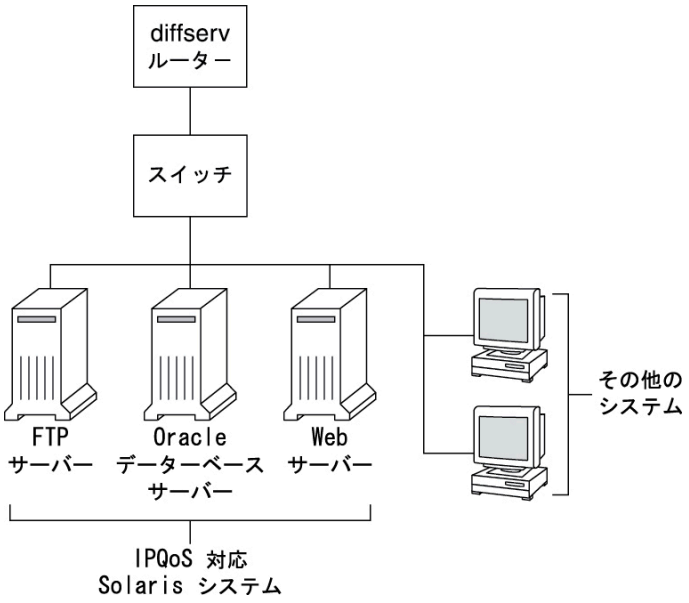
このセクションでは、ネットワーク上のさまざまなニーズを満たす IPQoS 計画を図で説明します。

個々のホストでの IPQoS

次の図は、IPQoS 対応システムの単一ネットワークを示しています。このネットワークは、ある企業のイントラネットの一部分です。アプリケーションサーバーや Web サーバーで IPQoS を有効にすると、各 IPQoS システムが発信トラフィックを送出する速度を制御できます。ルーターが Diffserv に対応していれば、着信トラフィックおよび発信トラフィックをさらに制御できます。

このガイドの例では、このシナリオを使用します。

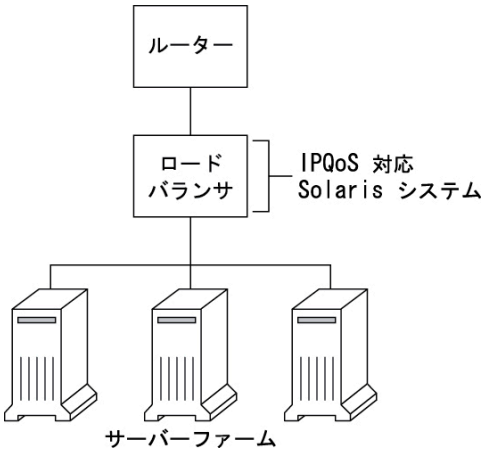
図 2-1 ネットワークセグメント上の IPQoS システム



サーバーファームのネットワークでの IPQoS

次の図には、複数の異種サーバーファームを備えるネットワークを示します。この手順では、ルーターが Diffserv に対応しているため、着信トラフィックと発信トラフィックの両方をキューに入れたり評価したりできます。また、ロードバランサも Diffserv 対応システムであるため、サーバーファームは IPQoS 対応となります。ロードバランサは、ユーザー ID やプロジェクト ID などのセレクタを使用することによって、ルーター以外で追加フィルタリングを提供できます。これらのセレクタは、アプリケーションデータに含まれます。

図 2-2 IPQoS 対応サーバーファームのネットワーク

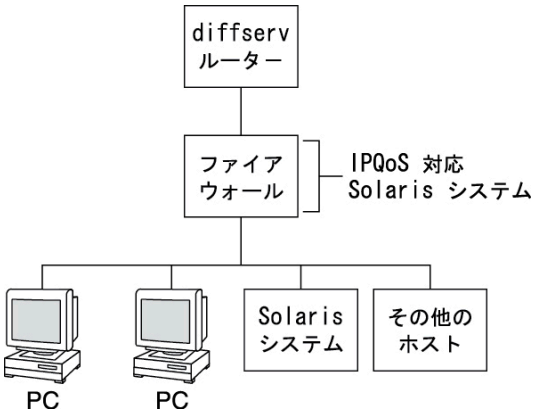


このシナリオでは、フロー制御とトラフィック転送を行って、ローカルネットワーク上の輻輳を管理しています。また、このシナリオでは、サーバーファームからの発信トラフィックが原因でイントラネットのほかの部分が過負荷状態になるのを防いでいます。

ファイアウォールでの IPQoS

次の図は、ファイアウォールによってほかのセグメントからセキュリティー保護されている、企業ネットワークのセグメントの 1 つを示しています。このシナリオでは、トラフィックはまず Diffserv 対応ルーターに入り、そこでフィルタにかけられ、キューに入れられます。次に、ルーターによって転送された着信トラフィックはすべて、IPQoS 対応のファイアウォールに進みます。IPQoS を使用するには、ファイアウォールで IP 転送スタックをバイパスしないでください。

図 2-3 IPQoS 対応のファイアウォールによって保護されているネットワーク



ファイアウォールのセキュリティーポリシーによって、着信トラフィックを内部ネットワークに入れて良いかどうかが決まります。QoS ポリシーは、ファイアウォールを通過した着信トラフィックのサービスレベルを制御します。QoS ポリシーによっては、発信トラフィックに転送動作を付けることもできます。

サービス品質ポリシーの計画

サービス品質 (QoS) ポリシーを計画するときは、ネットワークが提供するサービスの確認、分類、そして優先順位付けを行う必要があります。また、利用できる帯域幅の大きさを評価して、各トラフィッククラスがネットワークに送出される速度を決める必要もあります。

QoS ポリシー計画の手掛かり

IPQoS 構成ファイルで必要な情報を含む形式で QoS ポリシーの計画情報を収集します。たとえば、次のテンプレートを使って、IPQoS 構成ファイルで使用する主なカテゴリの情報を表にできます。

表 2-1 QoS 計画テンプレート

クラス	優先順位	フィルタ	セレクタ	レート	転送	アカウンティング
クラス 1	1	フィルタ 1 フィルタ 3	セレクタ 1 セレクタ 2	メーターの速度 (メーターの種類による)	マーカーのドロップ優先度	フローアカウンティング統計情報を必要とする
クラス 1	1	フィルタ 2	セレクタ 1 セレクタ 2	なし	なし	なし
クラス 2	2	フィルタ 1	セレクタ 1 セレクタ 2	メーターの速度 (メーターの種類による)	マーカーのドロップ優先度	フローアカウンティング統計情報を必要とする
クラス 2	2	フィルタ 2	セレクタ 1 セレクタ 2	なし	なし	なし

各カテゴリをいくつかに分けて、QoS ポリシーをさらに細かく定義できます。以降のセクションでは、テンプレートのカテゴリの情報を入手する方法について説明します。

QoS ポリシーの計画のタスクマップ

次のタスクマップでは、QoS ポリシーを計画するための主なタスクのリストと、各タスクの実行手順へのリンクを示します。

タスク	説明	手順の参照先
1. IPQoS をサポートするようネットワークトポロジを設計する	差別化サービスを提供するネットワーク上のホストとルーターを特定する	31 ページの「IPQOS のネットワークの準備」
2. ネットワーク上のサービスを分類するためのクラスを定義する	自分のサイトで提供するサービスの種類と SLA を調べ、これらのサービスを分類するためのトラフィッククラスを決める	31 ページの「QoS ポリシーのクラスの定義」
3. クラス用のフィルタを定義する	特定のトラフィッククラスをネットワークトラフィックフローから取り出すための最善の方法を決める	33 ページの「QoS ポリシーにフィルタを定義する方法」
4. IPQoS システムから送出されるトラフィックを測定するためのフロー制御速度を定義する	トラフィッククラスごとに容認できるフロー速度を決める	34 ページの「フロー制御を計画する方法」

タスク	説明	手順の参照先
5. QoS ポリシーで使用する DSCP すなわちユーザー優先順位の値を定義する	トラフィックフローがルーターまたはスイッチによって処理されるときに割り当てられる転送動作を決めるスキームを計画する	37 ページの「転送動作の計画」
6. 必要であれば、ネットワーク上のトラフィックフローに対する統計モニタリング計画を設定する	トラフィッククラスを評価して、アカウンティングまたは統計上の目的でモニターするトラフィックフローを決める	39 ページの「フローアカウンティングの計画」

注記 - このセクションでは、IPQoS 対応システムの QoS ポリシーを計画する方法について説明します。Diffserv ルーター向けの QoS ポリシーの計画を立てるには、ルーターのドキュメントおよびルーター製造元の Web サイトを参照してください。

IPQOS のネットワークの準備

QoS ポリシーを作成する前に行う一般的な計画タスクは、次のとおりです。

1. ネットワークトポロジを見直します。次に、IPQoS システムと Diffserv ルーターを使用する戦略の計画を作成します。トポロジの例は、26 ページの「diffserv ネットワークトポロジの計画」を参照してください。
2. IPQoS を必要とするホスト、または IPQoS サービスの有力な候補となるホストをトポロジで特定します。
3. IPQoS 対応後に同じ QoS ポリシーを共用できそうなシステムを調べます。

   たとえば、ネットワーク上のすべてのホストで IPQoS を有効にする場合は、同じ QoS ポリシーを使用できるホストをすべて特定します。各 IPQoS 対応システムにはローカル QoS ポリシーが必要で、そのポリシーはそれぞれの IPQoS 構成ファイルに実装されます。ただし、IPQoS 構成ファイルを 1 つだけ作成し、これを同じ QoS ポリシー要件を持つすべてのシステムにコピーして使うこともできます。
4. ネットワーク上の Diffserv ルーターに対して必要な計画タスクをすべて確認し、実行します。詳細については、ルーターのドキュメントとルーター製造元の Web サイトを参照してください。

QoS ポリシーのクラスの定義

QoS ポリシーを定義するための最初の手順は、トラフィックフローをクラスに整理することです。Diffserv ネットワーク上のトラフィックの種類ごとにクラスを作成する必要はありません。ま

た、計画したネットワークトポロジによっては、IPQoS 対応システムごとに異なる QoS ポリシーを作成する必要があります。クラスの概要は、17 ページの「IPQoS クラス」を参照してください。

クラスを定義する前に、31 ページの「IPQOS のネットワークの準備」の説明に従って、ネットワークのどのシステムを IPQoS 対応にするかを決定してください。

1. 表2-1「QoS 計画テンプレート」に示すように、QoS ポリシー情報を構成するための QoS 計画表を作成します。
2. ネットワーク上にあるすべての QoS ポリシーについて、残りの手順を実行します。
3. QoS ポリシーで使用するクラスを定義します。

   可能なクラス定義のネットワークトラフィックを分析するためのガイドラインについては、13 ページの「サービスクラスを使ったトラフィックの優先順位付け」を参照してください。
4. QoS 計画表にクラスを一覧表示します。
5. 優先レベルを各クラスに割り当てます。

   たとえば、優先レベル 1 がもっとも高い優先順位のクラスを表すようにし、それ以降の優先レベルを残りのクラスに割り当てます。この優先レベルの目的は、クラスを整理することだけです。QoS ポリシーによっては、同じ優先順位を複数のクラスに割り当てることもできます。

   PHB をクラスに割り当てるほかに、そのクラスのフィルタに優先順位セレクタを定義することもできます。優先順位セレクタは、IPQoS 対応ホストでのみ有効です。たとえば、同じ速度と同じ DSCP を持ついくつかのクラスが、IPQoS システムから送出されるときに帯域幅をめぐって競合することがあるとします。このような場合、各クラスの優先順位セレクタによって、ほかの点ではまったく同じ評価のクラスに割り当てられるサービスレベルをさらに細かく順序付けることができます。

   クラスの定義の次は、33 ページの「QoS ポリシーにフィルタを定義する方法」の説明に従って、各クラスのフィルタを定義します。

フィルタの定義

パケットフローを特定のクラスのメンバーとして識別するには、フィルタを作成します。各フィルタには、パケットフローの評価基準を定義するセレクタがいくつか含まれています。IPQoS 対応システムは、次にセレクタの基準を使用して、トラフィックフローからパケットを抽出します。そして、IPQoS システムは、パケットとクラスを関連付けます。フィルタの概要については、16 ページの17 ページの「IPQoS フィルタ」を参照してください。

また、クラスのパケットを取り出すのに必要なものだけを使用してください。定義するセレクタが多いほど、IPQoS パフォーマンスに与える影響も大きくなります。

次の表に、もっとも一般的に使用されるセレクタを示します。最初の 5 つのセレクタは、IPQoS 5 タプルを表し、IPQoS システムがパケットをフローのメンバーとして識別するときに使用します。セレクタの完全なリストについては、表6-1「IPQoS クラシファイアで利用可能なフィルタセレクタ」を参照してください。

表 2-2 一般的な IPQoS セレクタ

名前	定義
saddr	発信元アドレス
daddr	着信先アドレス
sport	発信元ポート番号。/etc/services に定義されている既知のポート番号、またはユーザー定義のポート番号を使用できる
dport	着信先ポート番号
protocol	IP プロトコル番号またはプロトコル名。/etc/protocols のトラフィックフロータイプに割り当てられる
ip_version	IPv4 (デフォルト) または IPv6 のいずれかを使用するアドレス指定スタイル。
dsfield	DS フィールドの内容、つまり DSCP。このセレクタは、特定の DSCP が付いている着信パケットを取り出すために使用する
priority	クラスに割り当てられている優先レベル。詳細は、31 ページの「QoS ポリシーのクラスの定義」を参照してください。
user	上位アプリケーションの実行時に使用される UNIX のユーザー ID またはユーザー名。
projid	プロジェクト ID。上位アプリケーションの実行時に使用される
direction	トラフィックフローの方向。有効な値は、LOCAL_IN、LOCAL_OUT、FWD_IN、FWD_OUT のいずれかです。

▼ QoS ポリシーにフィルタを定義する方法

始める前に フィルタを定義する前に、QoS ポリシーのクラスを定義する必要があります。詳細は、29 ページの31 ページの「QoS ポリシーのクラスの定義」を参照してください。

1. **クラスごとに最低 1 つのフィルタを QoS 計画表に作成します。**

   必要であれば、1 つのクラスの着信トラフィックと発信トラフィックに対して、フィルタを別々に作成することを検討してください。たとえば、ftp-in フィルタと ftp-out フィルタを IPQoS 対

応の FTP サーバーの QoS ポリシーに追加します。そうすれば、基本セレクタに加えて、該当する方向セレクタも定義できます。

2. **クラスのフィルタごとにセレクタを最低 1 つ定義します。**
QoS 計画表を使用して、定義したクラスのフィルタを追跡します。

例 2-1 FTP トラフィック向けのフィルタの定義

次の例は、発信 FTP トラフィック向けにフィルタを定義する方法を示しています。

クラス	優先順位	フィルタ	セレクタ
ftp-traffic	4	ftp-out	saddr 10.190.17.44 daddr 10.100.10.53 sport 21 direction LOCAL_OUT

フロー制御の計画

フロー制御では、クラスのトラフィックフローを測定し、定義された速度でネットワークにパケットを送出します。フロー制御を計画するときは、IPQoS メータリングモジュールが使用するパラメータを定義します。メーターは、トラフィックがネットワークに送出される速度を決定します。メータリングモジュールの紹介は、17 ページの「メーター (tokenmt および tswtclmt) の概要」を参照してください。

トラフィックの計測は、一般に次の理由で行います。

- SLA は、ネットワークの使用率が高いときでも、このクラスのパケットにある程度のサービスを保証する。
- 低い優先順位のクラスがネットワークをあふれさせる傾向がある

マーカーをメーターと組み合わせて使用すると、これらのクラスに対して差別化サービスを提供したり帯域幅の管理を行ったりできます。

▼ フロー制御を計画する方法

始める前に フロー制御の計画前に、33 ページの「QoS ポリシーにフィルタを定義する方法」の説明に従って、フィルタまたはセレクタを定義しておくようにしてください。

1. **ネットワークの最大帯域幅を調べます。**

2. **ネットワークでサポートされている SLA を確認し、顧客と、各顧客に保証されているサービスの種類を特定します。**

   一定レベルのサービスを保証するには、顧客によって生成される特定のトラフィッククラスを計測する必要があります。

3. **クラスのリストを確認し、SLA に対応付けられているクラス以外に計測する必要があるクラスがあるかどうか判断します。**

   たとえば、IPQoS システムが大量のトラフィックを生成するアプリケーションを実行するとします。この場合は、そのアプリケーションのトラフィックを分類したあと、フローのパケットがネットワークに戻される速度を制御して、フローを計測します。

   注記 - すべてのクラスを計測する必要があるとは限りません。

4. **各クラスのどのフィルタがフロー制御に必要なトラフィックを選択するのかを判断します。次に、メータリングを必要とするクラスのリストを精緻化します。**

   複数のフィルタを持つクラスでも 1 つのフィルタに対してだけ計測を必要とする場合もあります。たとえば、特定のクラスの着信および発信トラフィックのフィルタを定義する場合、1 方向のトラフィックのみがフロー制御を必要とする結論になることがあります。

5. **フロー制御するクラスごとにメーターモジュールを選択し、QoS 計画表のメーター欄にモジュール名を追加します。**

6. **計測するクラスごとの速度を計画表に追加します。**

   `tokenmt` モジュールを使用する場合は、次の速度をビット/秒で定義する必要があります。

   - 認定速度
   - 最大速度

   特定のクラスの計測に十分な場合は、認定速度と認定バーストを `tokenmt` に定義するだけでも構いません。

   必要に応じて、次の速度も定義できます。

   - 認定バースト
   - 最大バースト

tokenmt 速度の詳しい説明については、91 ページの「tokenmt をツーレートメーターとして構成する」を参照してください。tokenmt(7ipp) のマニュアルページでも詳しく説明しています。

tswtclmt モジュールを使用する場合は、次の速度を bps で定義する必要があります。

- 認定速度
- 最大速度

また、ウィンドウサイズをミリ秒単位で定義することもできます。これらの速度は、93 ページの「tswtclmt メータリングモジュール」および tswtclmt(7ipp) のマニュアルページで定義されています。

7. **計測するトラフィックのトラフィック適合結果 (outcome) を計画表に追加します。**

   どちらのメータリングモジュールでも、結果は緑、赤、黄です。メーターの結果については、90 ページの「メーターモジュール」で詳しく説明されています。

   認定速度に適合したトラフィックまたは適合しなかったトラフィックに対して取るべきアクションを決める必要があります。常にではありませんが多くの場合、このアクションは、ホップ単位の動作でパケットヘッダーをマークすることです。緑レベルのトラフィックに対して取りうるアクションの 1 つとして、トラフィックフローが認定速度を超えないかぎり処理を続行することもあります。あるいは、フローが最大速度を超えた場合にそのクラスのパケットをドロップすることもできます。

例 2-2 メーターの定義

次の表では、電子メールトラフィックのクラスに対するメーターの記入例を示します。IPQoS システムを持つネットワークの総帯域幅は 100 Mbits/秒、つまり毎秒 10000000 ビットです。QoS ポリシーは、電子メールクラスに低い優先順位を割り当てます。このクラスには、ベストエフォート型の転送動作も割り当てられます。

クラス	優先順位	フィルタ	セレクタ	レート
email	8	mail_in	daddr10.50.50.5 dport imap direction LOCAL_IN	
email	8	mail_out	saddr10.50.50.5	メーター =tokenmt 認定速度 =5000000

クラス	優先順位	フィルタ	セレクタ	レート
			sport imap direction LOCAL_OUT	認定バースト =5000000 最大速度 =10000000 最大バースト =1000000 緑の優先度 = 処理続行 黄の優先度 = 黄の PHB を付加 赤の優先度 = ドロップ

転送動作の計画

転送動作によって、ネットワークに転送されるトラフィックフローの優先度とドロップ優先順位が決まります。選択できる主な転送動作は 2 つあり、1 つはほかのトラフィッククラスとの関連でクラスのフローに優先順位を付けることで、もう 1 つはフローを完全にドロップすることです。

Diffserv モデルでは、マーカーを使用して選択した転送動作をトラフィックフローに割り当てます。IPQoS には、次のマーカーモジュールが用意されています。

このセクションでは、IP パケットに限定して説明します。IPQoS システムに VLAN デバイスが含まれている場合は、`dlcosmk` マーカーを使って、データグラムの転送動作をマークできます。詳細は、96 ページの「VLAN デバイスでの `dlcosmk` マーカーの使用」を参照してください。

IP トラフィックに優先順位を付けるには、各パケットに DSCP を割り当てる必要があります。`dscpmk` マーカーは、パケットの DS フィールドに DSCP のマークを設定します。クラスの DSCP は、転送動作の種類に対応付けられている既知のコードポイントのグループから選択します。既知のコードポイントには、EF PHB 用の 46 (101110) や AF PHB 用のいくつかのコードポイントがあります。DSCP と転送の概要情報については、21 ページの「IPQoS 対応ネットワークでのトラフィック転送」を参照してください。

▼ 転送動作を計画する方法

始める前に 転送動作を確認する前に、QoS ポリシーのクラスとフィルタを定義しておいてください。トラフィックを制御する場合は、メーターをマーカーと組み合わせて使用しますが、転送動作を定義するだけであれば、マーカーを単独で使用できます。

1. **これまでに作成したクラスとそれらに割り当てた優先順位を確認します。**

すべてのトラフィッククラスをマークする必要があるとは限りません。

2. **もっとも高い優先順位のクラスに EF PHB を割り当てます。**

   EF PHB は、EF DSCP 46 (101110) を持つパケットが、AF PHB を割り当てたパケットよりも先にネットワーク上に送出されることを保証します。このため、もっとも高い優先順位のトラフィックには EF PHB を使用します。EF については、94 ページの「完全優先転送 (Expedited Forwarding、EF) PHB」を参照してください。

3. **トラフィックを計測するクラスに転送動作を割り当てます。**

4. **残りのクラスに、すでに割り当てた優先順位に応じた DS コードポイントを割り当てます。**

例 2-3 ゲームアプリケーション向け QoS ポリシー

次の表は、QoS ポリシーの一部を示しています。このポリシーは、高いトラフィックレベルを生成する人気のあるゲームアプリケーション向けのクラスを定義します。

クラス	優先順位	フィルタ	セレクタ	レート	転送動作をどうするか
games_app	9	games_in	sport 6080	なし	なし
games_app	9	games_out	dport 6081	メーター =tokenmt 認定速度 = 5000000 認定バースト = 5000000 最大速度 = 10000000 最大バースト = 15000000 緑の優先度 = 処理続行 黄の優先度 = 黄の PHB を付加 赤の優先度 = ドロップ	緑 = AF31 黄 = AF42 赤 = ドロップ

これらの転送動作では、認定速度に適合しているか、最大速度を下回っている games_app トラフィックには、低い優先順位の DSCP を割り当てます。games_app トラフィックが最大速度を上回ると、QoS ポリシーは games_app のパケットをドロップするよう指示します。すべての AF コードポイントについては、表6-2「相対的優先転送のコードポイント」を参照してください。

フローアカウンティングの計画

IPQoS flowacct モジュールを使用して、請求またはネットワーク管理のためにトラフィックフローを追跡します。次の状況では、QoS ポリシーにフローアカウンティングを含めようにしてください。

- 会社が SLA を顧客に提供している。

  まず、SLA を確認して、企業が顧客に使用料を請求するネットワークトラフィックの種類を調べます。次に、QoS ポリシーを確認して、課金の対象となるトラフィックが含まれるクラスを調べます。
- ネットワークの問題を防ぐためにモニタリングまたはテストを必要とするアプリケーションはあるか。

  フローアカウンティングを使ってそれらのアプリケーションの動作を監視することを検討します。QoS ポリシーを確認して、モニタリングを必要とするトラフィックに割り当てたクラスを調べます。

QoS 計画表で、フローアカウンティングを必要とする各クラスのフローアカウンティング欄に Y と記入します。

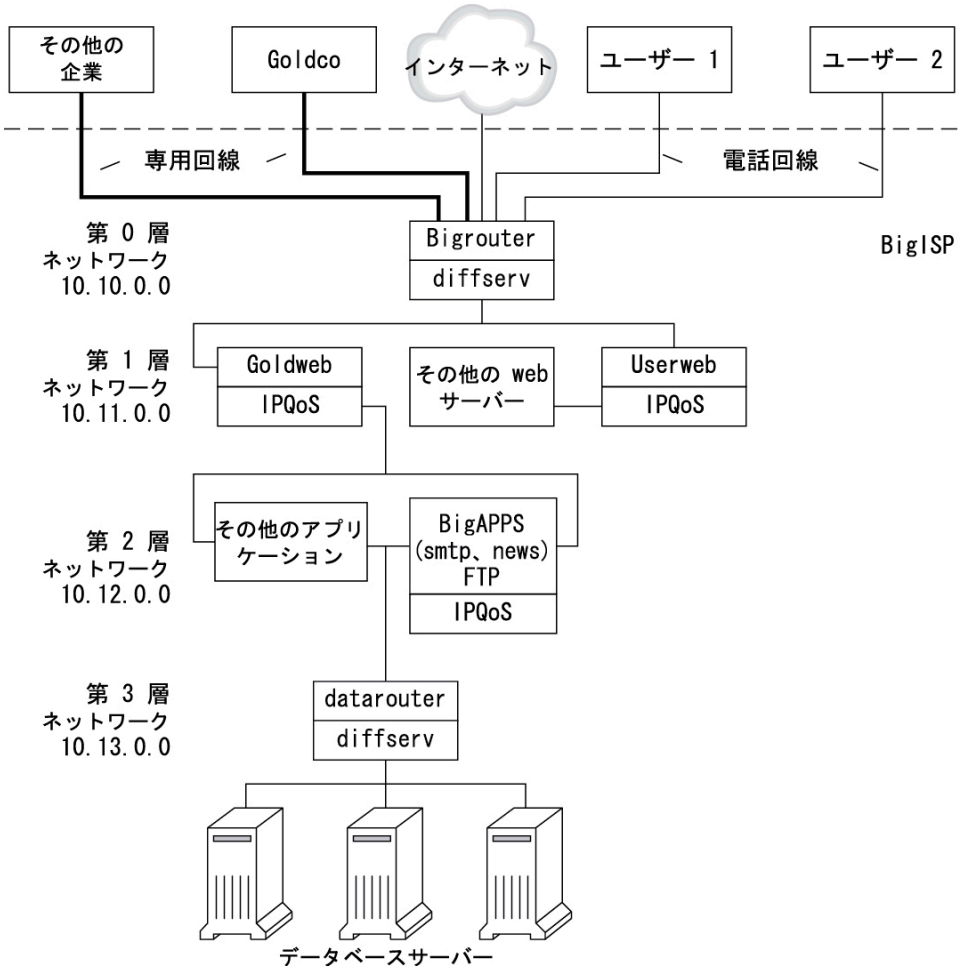
IPQoS の構成例の紹介

このセクションでは、ガイドの残りの章のタスクで使用する IPQOS の構成例について説明します。この例は、架空のサービスプロバイダである BigISP の公共イントラネットでの差別化サービスソリューションを示しています。BigISP は、専用回線によって BigISP にアクセスする大企業向けにサービスを提供しています。モデムによるダイアルインを行う個人顧客も BigISP からサービスを購入しています。

IPQoS トポロジ

次の図は、BigISP の公共イントラネットで使用するネットワークトポロジを示しています。

図 2-4 IPQoS のトポロジの例



BigISP には、公共イントラネットにこれらの 4 つの層があります。

- **第 0 層** – ネットワーク `10.10.0.0` には、`Bigrouter` という大規模な Diffserv ルーターがあり、外部インタフェースと内部インタフェースの両方を備えています。Goldco 社という大企

業をはじめとする数社の企業が Bigrouter で終端する専用回線サービスを借りています。第 0 層では、電話回線または ISDN を介して接続する個人客も管理しています。

- **第 1 層** – ネットワーク 10.11.0.0 では、Web サービスを提供しています。Goldweb サーバーは、Goldco 社が BigISP から購入したプレミアムサービスの一部である Web サイトのホストとして動作します。Userweb サーバーは、個人客が購入した小規模の Web サイトのホストして動作します。Goldweb と Userweb はどちらも IPQoS に対応しています。
- **第 2 層** – ネットワーク 10.12.0.0 では、すべての顧客が使用するアプリケーションを提供します。アプリケーションサーバーの 1 つである BigAPPS は IPQoS 対応サーバーです。BigAPPS は、SMTP、ニュース、および FTP サービスを提供します。
- **第 3 層** – ネットワーク 10.13.0.0 には、大規模データベースサーバーがいくつか格納されています。第 3 層へのアクセスは datarouter という Diffserv ルーターによって制御されます。

♦ ♦ ♦ 第 3 章

3 IPQoS 構成ファイルの作成のタスク

この章では、IPQoS 構成ファイル `/etc/inet/ipqosinit.conf` の作成方法について説明します。内容は次のとおりです。

- 43 ページの「QoS ポリシーの定義のタスクマップ」
- 44 ページの「QoS ポリシー作成用のツール」
- 45 ページの「Web サーバー用 IPQoS 構成ファイルの作成」
- 61 ページの「アプリケーションサーバー用 IPQoS 構成ファイルの作成」
- 71 ページの「ルーター上での差別化サービスの提供」

この章では、完全な QoS ポリシーを定義していること、このポリシーを IPQoS 構成ファイルのベースとしてすぐに使用できることを前提としています。QoS ポリシーを計画するには、29 ページの「サービス品質ポリシーの計画」を参照してください。

注記 - IPQoS 機能は、将来のリリースで削除される可能性があります。代わりに、同様の帯域幅リソース制御機能をサポートしている、`dladm`、`flowadm`、および関連コマンドを使用してください。詳細は、『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理 』を参照してください。

QoS ポリシーの定義のタスクマップ

このタスクマップでは、IPQoS 構成ファイルを作成するための一般的なタスクのリストと、各タスクの実行手順を説明したセクションへのリンクを示します。

タスク	説明	手順の参照先
1. IPQoS 対応のネットワーク構成を計画する	ローカルネットワーク上でどのシステムを IPQoS 対応にするかを決定する	31 ページの「IPQOS のネットワークの準備」

タスク	説明	手順の参照先
2. ネットワーク上の IPQoS システム用 QoS ポリシーを計画する	トラフィックフローを区別できるサービスクラスとして識別する。次に、トラフィック管理が必要なフローを決定する	29 ページの「サービス品質ポリシーの計画」
3. IPQoS 構成ファイルを作成し、その最初のアクションを定義する	IPQoS ファイルを作成し、IP クラシファイアを呼び出し、処理を実行させるクラスを定義する	48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」
4. トラフィックの選択とクラス分けとを規定するフィルタを追加する。	クラス用のフィルタを作成する。	50 ページの「IPQoS 構成ファイル内でフィルタを定義する方法」
5. IP クラシファイアに処理させるクラスおよびフィルタをさらに作成する。	IPQoS 構成ファイルにクラスおよびフィルタをさらに追加する。	58 ページの「ベストエフォート Web サーバー用の IPQoS 構成ファイルを作成する方法」
6. QoS ポリシーがフロー制御を必要とする場合、フロー制御速度および適合レベルをメーターに割り当てる。	メータリングモジュールを構成するパラメータを含む `action` 文を追加する。	68 ページの「IPQoS 構成ファイル内でフロー制御を構成する方法」
7. QoS ポリシーが差別化転送動作を必要とする場合、トラフィッククラスの転送方法を定義する。	マーカーを構成するパラメータを含む `action` 文を追加する。	52 ページの「IPQoS 構成ファイル内でトラフィック転送を定義する方法」
8. QoS ポリシーがトラフィックフローに関する統計の取得を必要とする場合、これらのアカウンティング統計の収集方法を定義する。	フローアカウンティングモジュールを構成するパラメータを含む `action` 文を追加する。	56 ページの「IPQoS 構成ファイル内でクラスのアカウンティングを有効にする方法」
9. 指定した IPQoS 構成ファイルの内容を、適切なカーネルモジュールに追加する。	IPQoS 構成ファイルを適用する。	74 ページの「`ipqos` サービスを開始する方法」
10. ネットワーク上のいずれかの IPQoS 構成ファイルで転送動作が定義されている場合、結果として得られる DSCP を、ルーターの適切なスケジューリングファイルに追加する。	転送動作をルーターファイル内で構成する。	71 ページの「ルーター上での差別化サービスの提供」

QoS ポリシー作成用のツール

ネットワークの QoS ポリシーは、IPQoS 構成ファイル `/etc/inet/ipqosinit.conf` にあります。テキストエディタでこの構成ファイルを作成します。次に、`ipqosconf` コマンドを実行する `svc:/network/ipqos` サービスを開始します。ポリシーはカーネル IPQoS システムに書き込ま

れます。ipqosconf コマンドの詳細は、ipqosconf(1M) のマニュアルページを参照してください。例6-3「IPQoS 構成ファイルの構文」にも、IPQoS 構成ファイルの完全な構文が示されています。

基本 IPQoS 構成ファイル

IPQoS 構成ファイルは、29 ページの「サービス品質ポリシーの計画」で定義した QoS ポリシーを実装する action 文のツリーで構成されています。IPQoS 構成ファイルは、IPQoS モジュールの構成を行います。各アクション文には、アクション文の中で呼び出されるモジュールが処理する「*クラス*」、「*フィルタ*」、または「*パラメータ*」のセットが含まれます。

この章のタスクでは、3 つの IPQoS 対応システム用の IPQoS 構成ファイルを作成する方法を示します。これらのシステムは、図2-4「IPQoS のトポロジの例」で紹介した BigISP 社のネットワークトポロジの一部です。

- Goldweb – プレミアムレベル SLA を購入した顧客用 Web サイトのホストとして機能する Web サーバー
- Userweb – ベストエフォート SLA を購入したホームユーザー向けの個人用 Web サイトのホストとして機能する、やや能力の劣る Web サーバー
- BigAPPS – ゴールドレベルとベストエフォートの両方の顧客に、メール、ネットワークニュース、および FTP を提供するアプリケーションサーバー

3 つのサンプル構成ファイルを通して、もっとも一般的な IPQoS 構成を示します。IPQoS を実装するために、次のセクションのサンプルファイルをテンプレートとして使用することもできます。

Web サーバー用 IPQoS 構成ファイルの作成

このセクションでは、プレミアム Web サーバーの構成の作成方法を示し、IPQoS 構成ファイルについて説明します。次に、個人用 Web サイトのホストとして機能するサーバー用の別の構成ファイルで、まったく異なるサービスレベルを構成する方法を示します。両方のサーバーは、図2-4「IPQoS のトポロジの例」で示したネットワーク例の一部です。

次の構成ファイルは、Goldweb サーバーの IPQoS アクティビティーを定義します。このサーバーは、プレミアム SLA を購入した Goldco 社の Web サイトのホストです。

例 3-1 プレミアム Web サーバー用 IPQoS 構成ファイルの例

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name goldweb
        next_action markAF11
        enable_stats FALSE
    }
    class {
        name video
        next_action markEF
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class goldweb
    }
    filter {
        name videoout
        sport videosrv
        direction LOCAL_OUT
        class video
    }
}
action {
    module dscpmk
    name markAF11
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}
action {
    module dscpmk
    name markEF
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}
action {
    module flowacct
    name acct
```

```
    params {
        enable_stats TRUE
        timer 10000
        timeout 10000
        max_limit 2048
    }
}
```

次の構成ファイルは、Userweb の IPQoS アクティビティーを定義します。このサーバーは、低価格または「ベストエフォート型」の SLA を購入した個人の Web サイトのホストです。このサービスレベルでは、IPQoS システムがより高額の SLA を利用する顧客からのトラフィックを処理したあとに、できるかぎり最良のサービスをベストエフォートの顧客に保証します。

例 3-2 ベストエフォート Web サーバー用の構成例

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name Userweb
        next_action markAF12
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class Userweb
   }
}
action {
    module dscpmk
    name markAF12
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}
```

▼ IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法

IPQoS 構成ファイルは、使用の準備ができたら /etc/inet/ipqosinit.conf にコピーする必要があります。新しいインストールで開始する場合は、ドラフト構成ファイルが使用される場所で、このファイルを編集した方が簡単です。次の手順では、例3-1「プレミアム Web サーバー用 IPQoS 構成ファイルの例」で紹介された IPQoS 構成ファイルの最初のセグメントを構築します。

注記 - IPQoS 構成ファイルを作成する際、各 action 文および句を必ず中括弧 ({ }) で囲んでください。中括弧の使用例については、例3-1「プレミアム Web サーバー用 IPQoS 構成ファイルの例」を参照してください。

---

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **プレミアム Web サーバーにログインします。**

3. **/etc/inet/ipqosinit.conf を編集します。**

4. **最初の非コメント行として、バージョン番号 fmt_version 1.0 を挿入します。**

   すべての IPQoS 構成ファイルは、この行から開始する必要があります。

5. **冒頭の action 文を挿入し、汎用の IP クラシファイア ipgpc を構成します。**

   IPQoS 構成ファイルを成す action 文のツリーは、この初期アクションから始まります。たとえば、構成ファイルは、ipgpc クラシファイアを呼び出す冒頭の action 文で始まります。

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
```

fmt_version 1.0	IPQoS 構成ファイルを開始する
action {	action 文を開始する
module ipgpc	構成ファイル内の最初のアクションとして ipgpc クラシファイアを構成する

name ipgpc.classify	クラシファイアの action 文の名前を定義する。名前は常に ipgpc.classify でなければならない

action 文の詳しい構文については、102 ページの「action 文」および ipqosconf(1M) のマニュアルページを参照してください。

6. **統計パラメータ global_stats を含む params 句を追加します。**

```
params {
         global_stats TRUE
   }
```

ipgpc.classify 文のパラメータ global_stats TRUE は、そのアクションに関する統計の収集を可能にします。また、global_stats TRUE を使用し、かつクラス句定義に enable_stats TRUE を指定すれば、そのクラスの統計の収集が可能になります。

統計を有効にすると、パフォーマンスに影響を与えます。新規 IPQoS 構成ファイルを作成したときには、IPQoS が適正に動作するか検証するために、統計収集を有効にしてもかまいません。あとで global_stats の引数を FALSE に変更すれば、統計収集を無効にできます。

グローバル統計は、params 句で定義可能なパラメータの 1 種類に過ぎません。params 句の構文などについては、104 ページの「params 句」および ipqosconf(1M) のマニュアルページを参照してください。

7. **プレミアムサーバーに向かうトラフィックを特定するクラスを定義します。**

```
class {
        name goldweb
        next_action markAF11
        enable_stats FALSE
    }
```

この文は、*class 句*と呼ばれます。この class 句には次の内容が含まれます。

name goldweb	goldweb クラスを作成して、Goldweb サーバーに向かうトラフィックを特定する
next_action markAF11	ipgpc モジュールに対し、goldweb クラスのパケットをアクション文 markAF11 に渡すよう指示する。アクション文 markAF11 は、dscpmk マーカーを呼び出す。
enable_stats FALSE	goldweb クラスの統計取得を可能にする。ただし、enable_stats の値が FALSE であるため、このクラスの統計収集は無効になる。

class 句の構文の詳細については、103 ページの「class 句」および ipqosconf(1M) のマニュアルページを参照してください。

8. **もっとも高い優先順位の転送を必要とするアプリケーションを特定するクラスを定義します。**

```
class {
        name video
        next_action markEF
        enable_stats FALSE
    }
```

name video	video クラスを作成して、Goldweb サーバーから発信されるストリーミングビデオのトラフィックを特定する
next_action markEF	ipgpc モジュールに対し、ipgpc による処理が完了した video クラスのパケットを、markEF 文に渡すよう指示する。markEF 文は、dscpmk マーカーを呼び出す
enable_stats FALSE	video クラスの統計収集を可能にする。ただし、enable_stats の値が FALSE であるため、このクラスの統計収集は無効になる。

9. **変更を /etc/inet/ipqosinit.conf ファイルに保存します。**

- **変更する場合は、ipqos サービスを開始します。**

  サービスの開始または再開の具体的な手順については、74 ページの「ipqos サービスを開始する方法」を参照してください。

- **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

  必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

▼ IPQoS 構成ファイル内でフィルタを定義する方法

始める前に 次の手順の前に、構成ファイルの作成を開始しており、クラスを定義してあるものとします。48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」で作成した IPQoS 構成ファイルの構築を続行します。

注記 - IPQoS 構成ファイルを作成する際、各 `class` 句および `filter` 句を必ず中括弧 ({ }) で囲んでください。中括弧の使用例については、例3-1「プレミアム Web サーバー用 IPQoS 構成ファイルの例」を参照してください。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **IPQoS 構成ファイルが開かれていない場合は、開きます。**

3. **定義した最後のクラスの終わりを特定します。**

   たとえば、IPQoS 対応サーバー `Goldweb` では、次の `class` 句のあとから作業を始めます。

```
class {
        name video
        next_action markEF
        enable_stats FALSE
    }
```

4. **`filter` 句を定義し、IPQoS システムからの発信トラフィックを選択します。**

```
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class goldweb
    }
```

`name webout`	フィルタに `webout` という名前を割り当てる。
`sport 80`	ソースポート 80 (既知の HTTP (Web) トラフィック用ポート) のトラフィックを選択する。
`direction LOCAL_OUT`	ローカルシステムから発信されるトラフィックを選択する
`class goldweb`	フィルタが所属するクラス (このインスタンスでは `goldweb` クラス) を特定する

   IPQoS 構成ファイル内の `filter` 句の構文の詳細は、104 ページの「`filter` 句」を参照してください。

5. **IPQoS システムのストリーミングビデオトラフィックを選択する `filter` 句を定義します。**

```
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
```

`name videoout`	フィルタに `videoout` という名前を割り当てる。
`sport videosrv`	ソースポート `videosrv` のトラフィックを選択する。これは、以前にこのシステムのストリーミングビデオアプリケーション用に定義したポート
`direction LOCAL_OUT`	ローカルシステムから発信されるトラフィックを選択する
`class video`	フィルタが所属するクラス (このインスタンスでは `video` クラス) を特定する

6. **変更を `/etc/inet/ipqosinit.conf` ファイルに保存します。**

   - **変更する場合は、`ipqos` サービスを開始します。**
     サービスの開始または再開の具体的な手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。

   - **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**
     必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

▼ IPQoS 構成ファイル内でトラフィック転送を定義する方法

この手順では、IPQoS 構成ファイルにクラスのホップ単位の動作を追加して、トラフィック転送を定義する方法を示します。

注記 - 次の手順では、`dscpmk` マーカーモジュールを使用してトラフィック転送を構成する方法を示します。`dlclosmk` マーカーを使用した VLAN システムのトラフィック転送については、96 ページの「VLAN デバイスでの `dlcosmk` マーカーの使用」を参照してください。

始める前に 次の手順では、既存の IPQoS 構成ファイルにクラスおよびフィルタを定義してあるものとします。例3-1「プレミアム Web サーバー用 IPQoS 構成ファイルの例」からの IPQoS 構成ファイルの構築を続行します。

1. **管理者になります。**
   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **IPQoS 構成ファイルがまだ開かれていない場合は、開きます。**

3. **定義した最後のフィルタの終わりを特定します。**
   たとえば、IPQoS 対応サーバー `Goldweb` では、構成ファイルの `filter` 句のあとから作業を始めます。

```
filter {
        name videoout
        sport videosrv
        direction LOCAL_OUT
        class video
    }
}
```

   この `filter` 句は `ipgpc` クラシファイア `action` 文の終わりにあるため、filter を終了するには閉じ中括弧、`action` 文を終了するには 2 番目の閉じ中括弧が必要です。

4. **`action` 文でマーカーを呼び出します。**

```
action {
    module dscpmk
    name markAF11
```

`module dscpmk`	`dscpmk` マーカーモジュールを呼び出す
`name markAF11`	`action` 文に `markAF11` という名前を割り当てる。

   以前に定義した `goldweb` クラスには `next_action markAF11` という文が含まれています。この文は、クラシファイアによる処理が完了したトラフィックフローを、アクション文 `markAF11` に送信します。

5. **トラックフローに対してマーカーが取るアクションを定義します。**

```
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}
```

`global_stats FALSE`	マーカー `action` 文 `markAF11` の統計収集を可能にする。ただし、`enable_stats` の値が `FALSE` であるため、統計は収集されない。

`dscp_map{0–63:10}`	DSCP `10` を、マーカーにより処理中の `goldweb` クラスのパケットヘッダーに割り当てる
`next_action continue`	`userweb` クラスのパケットに対しこれ以上処理を行う必要がないこと、およびこれらのパケットをネットワークストリームに戻してもよいことを示す

DSCP `10` は、マーカーに対し、`dscp` マップ内のすべてのエントリを 10 進数値の 10 (バイナリ値 001010) に設定するよう指示します。このコードポイントは、`goldweb` トラフィッククラスのパケットが AF11 ホップ単位動作 (PHB) に従うことを示します。AF11 は、DSCP `10` を持つすべてのパケットが、低ドロップ、および高い優先順位のサービスを受けることを保証します。このため、`Goldweb` 上のプレミアム顧客用の発信トラフィックには、AF (相対的優先転送) PHB で指定可能なもっとも高い優先順位が与えられます。AF に設定可能な DSCP の表については、表6-2「相対的優先転送のコードポイント」を参照してください。

6. **別のマーカー `action` 文を開始します。**

```
action {
    module dscpmk
    name markEF
```

`module dscpmk`	`dscpmk` マーカーモジュールを呼び出す
`name markEF`	`action` 文に `markEF` という名前を割り当てる。

7. **トラフィックフローに対してマーカーが取るアクションを定義します。**

```
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}
```

`global_stats TRUE`	`video` クラスの統計収集を有効にする。このクラスはストリーミングビデオのパケットを選択する
`dscp_map{0–63:46}`	DSCP `46` を、マーカーにより処理中の `video` クラスのパケットヘッダーに割り当てる
`next_action acct`	`dscpmk` モジュールに対し、`dscpmk` による処理が完了した `video` クラスのパケットを、`action` 文 `acct` に渡すよう指示する。`action` 文 `acct` は `flowacct` モジュールを呼び出す。

DSCP 46 は、dscpmk モジュールに対し、dscp マップのすべてのエントリを DS フィールドの 10 進数の 46 (バイナリ 101110) に設定するよう指示します。このコードポイントは、video トラフィッククラスのパケットが完全優先転送ホップ単位動作 (PHB) に従うことを示します。

注記 - EF のコードポイントは 46 (バイナリ値 101110) にすることをお勧めします。その他の DSCP は、AF PHB をパケットに割り当てるときに使用します。

EF PHB は、DSCP 46 を持つパケットが IPQoS および Diffserv 対応システムによりもっとも高い優先度を与えられることを保証します。ストリーミングアプリケーションは、もっとも高い優先順位のサービスを必要とします。これが、QoS ポリシーでこれらのアプリケーションに EF PHB を割り当てる理由です。PHB の完全優先転送の詳細については、94 ページの「完全優先転送 (Expedited Forwarding、EF) PHB」を参照してください。

8. **作成したばかりの DSCP を Diffserv ルーターの適切なファイルに追加します。**

   詳細は、71 ページの「ルーター上での差別化サービスの提供」を参照してください。

9. **変更を /etc/inet/ipqosinit.conf ファイルに保存します。**

   - **変更する場合は、ipqos サービスを開始します。**

     サービスの開始または再開の具体的な手順については、74 ページの「ipqos サービスを開始する方法」を参照してください。

   - **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

     必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

次の手順

- トラフィックフローのフローアカウンティング統計の収集を開始するには、56 ページの「IPQoS 構成ファイル内でクラスのアカウンティングを有効にする方法」を参照してください。
- マーカーモジュールの転送動作を定義するには、52 ページの「IPQoS 構成ファイル内でトラフィック転送を定義する方法」を参照してください。
- メータリングモジュールのフロー制御パラメータを定義するには、68 ページの「IPQoS 構成ファイル内でフロー制御を構成する方法」を参照してください。
- IPQoS 構成ファイルをアクティブ化するには、74 ページの「ipqos サービスを開始する方法」を参照してください。

- さらにフィルタを定義するには、50 ページの「IPQoS 構成ファイル内でフィルタを定義する方法」を参照してください。
- アプリケーションからのトラフィックフロー向けのクラスを作成するには、63 ページの「アプリケーションサーバー用 IPQoS 構成ファイルを作成する方法」を参照してください。

▼ IPQoS 構成ファイル内でクラスのアカウンティングを有効にする方法

この手順では、IPQoS 構成ファイル内でトラフィッククラスのアカウンティングを有効にする方法を示します。48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」で紹介した `video` クラスのフローアカウンティングを定義します。このクラスは、プレミアム SLA の一部として課金されるストリーミングビデオのトラフィックを選択します。

始める前に 次の手順では、既存の IPQoS 構成ファイルにクラス、フィルタ、メーターのアクション (必要な場合だけ)、およびマーカーのアクション (必要な場合だけ) を定義してあるものとします。例 3-1「プレミアム Web サーバー用 IPQoS 構成ファイルの例」からの IPQoS 構成ファイルの構築を続行します。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **IPQoS 構成ファイルがまだ開かれていない場合は、開きます。**

3. **定義した最後の `action` 文の終わりを特定します。**

   たとえば、IPQoS 対応サーバー `Goldweb`, では、構成ファイル `/etc/inet/ipqosinit.conf` の `markEF action` 文のあとから作業を始めます。

   ```
   action {
       module dscpmk
       name markEF
       params {
           global_stats TRUE
           dscp_map{0-63:46}
           next_action acct
       }
   }
   ```

4. **フローアカウンティングを呼び出す `action` 文を開始します。**

```
action {
    module flowacct
    name acct
```

`module flowacct`	`flowacct` フローアカウンティングモジュールを呼び出す
`name acct`	`action` 文に `acct` という名前を割り当てる。

5. **トラフィッククラスに関するアカウンティングを制御する `params` 句を定義します。**

```
params {
        global_stats TRUE
        timer 10000
        timeout 10000
        max_limit 2048
        next_action continue
    }
}
```

`global_stats TRUE`	`video` クラスの統計収集を有効にする。このクラスはストリーミングビデオのパケットを選択する
`timer 10000`	フローテーブル内で、タイムアウトしたフローが走査される間隔を、ミリ秒単位で指定する。このパラメータでは、間隔は 10000 ミリ秒
`timeout 10000`	最小の間隔タイムアウト値を指定する。フローのパケットがタイムアウト値で指定された時間検出されないと、フローは「タイムアウト」する。このパラメータでは、パケットは 10000 ミリ秒後にタイムアウトする
`max_limit 2048`	このアクションインスタンスのフローテーブル内でアクティブなフローレコードの最大数を設定する
`next_action continue`	`video` クラスのパケットに対しこれ以上処理を行う必要がないこと、およびこれらのパケットをネットワークストリームに戻してもよいことを示す

`flowacct` モジュールは、指定された `timeout` 値に達するまで、特定のクラスのパケットフローに関する統計情報を収集します。

6. **変更を `/etc/inet/ipqosinit.conf` ファイルに保存します。**

■ **変更する場合は、`ipqos` サービスを開始します。**

サービスの開始または再開の具体的な手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。

- **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

  必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

▼ ベストエフォート Web サーバー用の IPQoS 構成ファイルを作成する方法

ベストエフォート Web サーバー用の IPQoS 構成ファイルは、プレミアム Web サーバー用の IPQoS 構成ファイルとは少し違います。この手順では、例3-2「ベストエフォート Web サーバー用の構成例」からの構成ファイルを使用します。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **ベストエフォート Web サーバーにログインします。**

3. **新規 IPQoS 構成ファイルを拡張子 `.qos` を付けて作成します。**

```
fmt_version 1.0
action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
   }
```

   ファイルは、`ipgpc` クラシファイアを呼び出す部分 `action` 文から始める必要があります。また、`action` 文には、統計収集を有効にする `params` 句も含めています。この `action` 文については、48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」を参照してください。

4. **ベストエフォート Web サーバーに向かうトラフィックを特定するクラスを定義します。**

```
class {
        name userweb
        next_action markAF12
        enable_stats FALSE
    }
```

`name userweb`	`userweb` クラスを作成して、ユーザーから Userweb サーバーに向かうトラフィックを特定する
`next_action markAF1`	`ipgpc` モジュールに対し、`ipgpc` による処理が完了した `userweb` クラスのパケットを、`action` 文 `markAF12` に渡すよう指示する。`action` 文 `markAF12` は、`dscpmk` マーカーを呼び出す
`enable_stats FALSE`	`userweb` クラスの統計収集を可能にする。ただし、`enable_stats` の値が `FALSE` であるため、このクラスの統計は収集されない

`class` 句の処理については、48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」を参照してください。

5. **`userweb` クラスのトラフィックフローを選択する `filter` 句を定義します。**

```
    filter {
       name webout
       sport 80
       direction LOCAL_OUT
       class userweb
   }
}
```

`name webout`	フィルタに `webout` という名前を割り当てる。
`sport 80`	ソースポート 80 (既知の HTTP (Web) トラフィック用ポート) のトラフィックを選択する。
`direction LOCAL_OUT`	ローカルシステムから発信されるトラフィックを選択する
class userweb	フィルタが所属するクラス (このインスタンスでは `userweb` クラス) を特定する

`filter` 句の処理については、50 ページの「IPQoS 構成ファイル内でフィルタを定義する方法」を参照してください。

6. **`dscpmk` マーカーを呼び出す `action` 文を開始します。**

```
action {
    module dscpmk
    name markAF12
```

`module dscpmk`	`dscpmk` マーカーモジュールを呼び出す

`name markAF12`	`action` 文に `markAF12` という名前を割り当てる。

以前に定義した `userweb` クラスには `next_action markAF12` という文が含まれています。この文は、クラシファイアによる処理が完了したトラフィックフローを、 `action` 文 `markAF12` に送信します。

7. **トラフィックフローの処理に使用する、マーカーのパラメータを定義します。**

```
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}
```

`global_stats FALSE`	マーカー `action` 文 `markAF12` の統計収集を可能にする。ただし、`enable_stats` の値が `FALSE` であるため、統計は収集されない。
`dscp_map{0–63:12}`	DSCP 12 を、マーカーにより処理中の `userweb` クラスのパケットヘッダーに割り当てる
`next_action continue`	`userweb` クラスのパケットに対しこれ以上処理を行う必要がないこと、およびこれらのパケットをネットワークストリームに戻してもよいことを示す

DSCP 12 は、マーカーに対し、`dscp` マップ内のすべてのエントリを 10 進数値の 12 (バイナリ値 001100) に設定するよう指示します。このコードポイントは、`userweb` トラフィッククラスのパケットが AF12 ホップ単位動作 (PHB) に従うことを示します。AF12 は、DS フィールド内に DSCP 12 を持つすべてのパケットが、中程度のドロップ、および高い優先順位のサービスを受けることを保証します。

8. **変更を `/etc/inet/ipqosinit.conf` ファイルに保存します。**

- **変更する場合は、`ipqos` サービスを開始します。**

  サービスの開始または再開の具体的な手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。

- **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

  必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

アプリケーションサーバー用 IPQoS 構成ファイルの作成

このセクションでは、顧客に主要アプリケーションを提供するアプリケーションサーバー用の構成ファイルを作成する方法について説明します。この手順では、例として図2-4「IPQoS のトポロジの例」の `BigAPPS` サーバーを使用します。

次の構成ファイルは、`BigAPPS` サーバーの IPQoS アクティビティーを定義します。このサーバーは、顧客向けの FTP、電子メール (SMTP)、およびネットワークニュース (NNTP) のホストです。

例 3-3 アプリケーションサーバー用サンプル IPQoS 構成ファイル

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name smtp
        enable_stats FALSE
        next_action markAF13
    }
    class {
        name news
        next_action markAF21
    }
    class {
        name ftp
        next_action meterftp
    }
    filter {
        name smtpout
        sport smtp
        class smtp
    }
    filter {
        name newsout
        sport nntp
        class news
    }
    filter {
        name ftpout
        sport ftp
        class ftp
    }
   filter {
```

```
        name ftpdata
        sport ftp-data
        class ftp
    }
}
action {
    module dscpmk
    name markAF13
    params {
        global_stats FALSE
        dscp_map{0-63:14}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
action {
    module tokenmt
    name meterftp
    params {
        committed_rate 50000000
        committed_burst 50000000
        red_action_name AF31
        green_action_name markAF22
        global_stats TRUE
    }
}
action {
    module dscpmk
    name markAF31
    params {
        global_stats TRUE
        dscp_map{0-63:26}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF22
    params {
        global_stats TRUE
        dscp_map{0-63:20}
        next_action continue
    }
}
```

▼ アプリケーションサーバー用 IPQoS 構成ファイルを作成する方法

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **IPQoS 対応アプリケーションサーバーにログインします。**

3. **新規 IPQoS 構成ファイルを拡張子 `.qos` を付けて作成します。**

4. **`action` 文の最初に、`ipgpc` クラシファイアを呼び出す次の必須の句を挿入します。**

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
```

   冒頭の `action` 文については、48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」を参照してください。

5. **BigAPPS サーバー上の 3 つのアプリケーションからのトラフィックを選択するクラス定義を追加します。**

```
    class {
        name smtp
        enable_stats FALSE
        next_action markAF13
    }
    class {
        name news
        next_action markAF21
    }
    class {
        name ftp
        enable_stats TRUE
        next_action meterftp
    }
```

`name smtp` SMTP アプリケーションが扱う電子メールのトラフィックフローが含まれる、`smtp` という名前のクラスを作成する。

`enable_stats FALSE`	`smtp` クラスの統計収集を可能にする。ただし、`enable_stats` の値が `FALSE` であるため、このクラスの統計は取得されない
`next_action markAF13`	`ipgpc` モジュールに対し、`ipgpc` による処理が完了した `smtp` クラスのパケットを、`action` 文 `markAF13` に渡すよう指示する。
`name news`	NNTP アプリケーションが扱うネットワークニュースのトラフィックフローが含まれる `news` という名前のクラスを作成する。
`next_action markAF21`	`ipgpc` モジュールに対し、`ipgpc` による処理が完了した `news` クラスのパケットを、アクション文 `markAF21` に渡すよう指示する
`name ftp`	FTP アプリケーションが扱う発信トラフィックを処理する `ftp` という名前のクラスを作成する。
`enable_stats TRUE`	`ftp` クラスの統計収集を可能にする
`next_action meterftp`	`ipgpc` モジュールに対し、`ipgpc` による処理が完了した `ftp` クラスのパケットを、`action` 文 `meterftp` に渡すよう指示する。

クラスの定義の詳細については、48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」を参照してください。

6. **手順 2 で定義したクラスのトラフィックを選択する `filter` 句を定義します。**

```
    filter {
        name smtpout
        sport smtp
        class smtp
    }
    filter {
        name newsout
        sport nntp
        class news
    }
        filter {
        name ftpout
        sport ftp
        class ftp
    }
        filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
```

`name smtpout`	フィルタに `smtpout` という名前を割り当てる。
`sport smtp`	ソースポート 25 (既知の `sendmail` (SMTP) アプリケーション用ポート) のトラフィックを選択する。
`class smtp`	フィルタが所属するクラス (このインスタンスでは `smtp` クラス) を特定する
`name newsout`	フィルタに `newsout` という名前を割り当てる。
`sport nntp`	ソースポート名 `nntp` (既知のネットワークニュース (NNTP) アプリケーション用ポート) のトラフィックを選択する。
`class news`	フィルタが所属するクラス (このインスタンスでは `news` クラス) を特定する
`name ftpout`	フィルタに `ftpout` という名前を割り当てる。
`sport ftp`	ソースポート 21 (既知の FTP トラフィック用ポート番号) の制御データを選択する。
`name ftpdata`	フィルタに `ftpdata` という名前を割り当てる。
`sport ftp-data`	ソースポート 20 (既知の FTP データトラフィック用ポート番号) のトラフィックを選択する。
`class ftp`	`ftpout` および `ftpdata` フィルタが所属するクラス (このインスタンスでは `ftp`) を特定する。

7. **変更を `/etc/inet/ipqosinit.conf` ファイルに保存します。**

■ **変更する場合は、`ipqos` サービスを開始します。**

サービスの開始または再開の具体的な手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。

■ **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

▼ IPQoS 構成ファイル内でアプリケーショントラフィックの転送を構成する方法

次の手順では、アプリケーショントラフィックの転送の構成方法について示します。次の手順では、アプリケーショントラフィッククラスのホップ単位動作を定義します。これらのクラスは、ネットワーク上のほかのトラフィックよりも優先度を低くする場合があります。この手順では、例3-3「アプリケーションサーバー用サンプル IPQoS 構成ファイル」の IPQoS 構成ファイルを引き続き構築します。

始める前に この手順では、マークしたアプリケーションに対してクラスとフィルタを定義した既存の IPQoS 構成ファイルがあることを前提にしています。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **`/etc/inet/ipqosinit.conf` を開き、最後の `filter` 句の終わりを特定します。**

   `/etc/inet/ipqosinit.conf` で、最後のフィルタは次のとおりです。

   ```
    filter {
           name ftpdata
           sport ftp-data
           class ftp
       }
   }
   ```

3. **マーカーを呼び出します。**

   ```
   action {
       module dscpmk
       name markAF13
   ```

   `module dscpmk` `dscpmk` マーカーモジュールを呼び出す

   `name markAF13` `action` 文に `markAF13` という名前を割り当てる。

4. **電子メールのトラフィックフローにマークされるホップ単位動作を定義します。**

   ```
       params {
           global_stats FALSE
           dscp_map{0-63:14}
           next_action continue
       }
   ```

```
}
```

`global_stats FALSE`	マーカー `action` 文 `markAF13` の統計収集を可能にする。ただし、`enable_stats` の値が `FALSE` であるため、統計は収集されない。
`dscp_map{0–63:14}`	DSCP `14` を、マーカーにより処理中の `smtp` クラスのパケットヘッダーに割り当てる
`next_action continue`	`smtp` クラスのパケットに対しこれ以上処理を行う必要がないことを示す。よって、これらのパケットはネットワークストリームに戻すことができる

DSCP `14` は、マーカーに対し、`dscp` マップ内のすべてのエントリを 10 進数値の 14 (バイナリ値 001110) に設定するよう指示します。DSCP `14` は、AF13 のホップ単位の動作を設定します。マーカーは、`smtp` トラフィッククラスのパケットの DS フィールドに DSCP `14` を付けます。AF13 は、DSCP `14` を持つすべてのパケットに高いドロップ優先度を割り当てますが、それと同時に Class 1 の優先順位も保証するため、ルーターは電子メールの発信トラフィックに対し、キューの中で高い優先順位を与えます。設定可能な AF コードポイントのリストについては、表 6-2「相対的優先転送のコードポイント」を参照してください。

5. **マーカー `action` 文を追加して、ネットワークニュースのトラフィック用のホップ単位動作を定義します。**

```
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
```

`name markAF21`	`action` 文に `markAF21` という名前を割り当てる。
`dscp_map{0–63:18}`	DSCP `18` を、マーカーにより処理中の `nntp` クラスのパケットヘッダーに割り当てる

DSCP `18` は、マーカーに対し、`dscp` マップ内のすべてのエントリを 10 進数値の 18 (バイナリ値 010010) に設定するよう指示します。DSCP `18` は、AF21 のホップ単位の動作を設定します。マーカーは、`news` トラフィッククラスのパケットの DS フィールドに DSCP `18` を付けます。AF21 は DSCP `18` を持つすべてのパケットに低いドロップ優先度を保証しますが、優先順位は Class 2 にとどまります。よって、ネットワークニューストラフィックが振り落とされる可能性は低くなります。

6. **変更を `/etc/inet/ipqosinit.conf` ファイルに保存します。**

   - **変更する場合は、`ipqos` サービスを開始します。**

     サービスの開始または再開の具体的な手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。

   - **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

     必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

▼ IPQoS 構成ファイル内でフロー制御を構成する方法

ネットワークに送出される特定のトラフィックフローの速度を制御するには、メーターのパラメータを定義しなければなりません。IPQoS 構成ファイル内で、2 つのメーター `tokenmt` と `tswtclmt` とのどちらかを使用できます。

この手順では、例3-3「アプリケーションサーバー用サンプル IPQoS 構成ファイル」のアプリケーションサーバーの IPQoS 構成ファイルを引き続き構築します。次の手順では、メーターを構成し、メーター `action` 文の内部で呼び出される 2 つのマーカーアクションを構成します。

始める前に この手順では、フローを制御するアプリケーション用のクラスおよびフィルタを定義してあるものとします。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **`/etc/inet/ipqosinit.conf` を開きます。**

   次のマーカーアクションのあとに、変更を開始します。

```
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
```

3. **ftp クラスのトラフィックをフロー制御するメーター action 文を作成します。**

```
action {
    module tokenmt
    name meterftp
```

module tokenmt	tokenmt メーターを呼び出す
name meterftp	action 文に meterftp という名前を割り当てる。

4. **メーターの速度を構成するパラメータを追加します。**

```
params {
       committed_rate 50000000
       committed_burst 50000000
```

committed_rate 50000000	ftp クラスのトラフィックに 50,000,000 bps の転送速度を割り当てる
committed_burst 50000000	ftp クラスのトラフィックに 50,000,000 ビットのバーストサイズを割り当てる。

tokenmt パラメータについては、91 ページの「tokenmt をツーレートメーターとして構成する」を参照してください。

5. **次のようにパラメータを追加して、トラフィック適合優先順位を構成します。**

```
    red_action markAF31
    green_action_name markAF22
    global_stats TRUE
    }
}
```

red_action_name markAF31	ftp クラスのトラフィックフローが認定速度を超過した場合、パケットは、マーカー action 文 markAF31 に送信されることを示す。
green_action_name markAF22	ftp クラスのトラフィックフローが認定速度に適合する場合、パケットがアクション文 markAF22 に送られることを示す
global_stats TRUE	ftp クラスのメータリング統計取得を有効にする

トラフィックの適合性については、90 ページの「メーターモジュール」を参照してください。

6. **ホップ単位動作を ftp クラスの不適合トラフィックフローに割り当てるマーカー action 文を追加します。**

```
action {
    module dscpmk
    name markAF31
    params {
        global_stats TRUE
        dscp_map{0-63:26}
        next_action continue
    }
}
```

`module dscpmk`	`dscpmk` マーカーモジュールを呼び出す
`name markAF31`	`action` 文に `markAF31` という名前を割り当てる。
`global_stats TRUE`	`ftp` クラスの統計取得を有効にする
`dscp_map{0-63:26}`	`ftp` クラスのトラフィックが認定速度を超過した場合は常に、DSCP 26 を ftp クラスのパケットヘッダーに割り当てる
`next_action continue`	トラフィッククラス `ftp` のパケットに対しこれ以上処理を行う必要がないこと、およびこれらのパケットをネットワークストリームに戻してもよいことを示す。

DSCP 26 は、マーカーに対し、`dscp` マップ内のすべてのエントリを 10 進数値の 26 (バイナリ値 011010) に設定するよう指示します。DSCP 26 は、AF31 のホップ単位の動作を設定します。マーカーは、DS フィールドの DSCP 26 で `ftp` トラフィッククラスのパケットをマークします。

AF31 は DSCP 26 を持つすべてのパケットに低いドロップ優先度を保証しますが、優先順位は Class 3 にとどまります。このため、速度不適合の FTP トラフィックがドロップされる可能性は低くなりますが、表6-2「相対的優先転送のコードポイント」に、設定可能な AF コンポーネントを示します。

7. **認定速度に適合する `ftp` トラフィックフローにホップ単位動作を割り当てるマーカー `action` 文を追加します。**

```
action {
    module dscpmk
    name markAF22
    params {
        global_stats TRUE
        dscp_map{0-63:20}
        next_action continue
    }
}
```

`name markAF22`	`marker` アクションに `markAF22` という名前を割り当てる。
`dscp_map{0–63:20}`	`ftp` クラスのトラフィックが認定速度に適合する場合は常に、DSCP `20` をパケットヘッダーに割り当てる

DSCP 20 は、マーカーに対し、`dscp` マップ内のすべてのエントリを 10 進数値の 20 (バイナリ値 010100) に設定するよう指示します。DSCP `20` は、AF22 のホップ単位の動作を設定します。マーカーは、DS フィールドの DSCP `20` で `ftp` トラフィッククラスのパケットをマークします。

AF22 は、DSCP `20` を持つすべてのパケットに中程度のドロップ優先度と Class 2 の優先順位を保証します。このため、速度適合の FTP トラフィックは、IPQoS システムから同時に送出されるフロー内で中程度のドロップ優先度を保証されます。ただし、ルーターは、Class 1 で中程度のドロップ優先度以上を持つトラフィッククラスの転送を優先します。表6-2「相対的優先転送のコードポイント」 に、設定可能な AF コンポーネントを示します。

8. **アプリケーションサーバー用に作成した DSCP を、Diffserv ルーターの適切なファイルに追加します。**

9. **変更を `/etc/inet/ipqosinit.conf` ファイルに保存します。**

   - **変更する場合は、`ipqos` サービスを開始します。**

     サービスの開始または再開の具体的な手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。

   - **IPQoS 構成ファイルで変更を続行する場合は、別のタスクを選択します。**

     必要になる可能性のある追加の変更のリストについては、25 ページの「一般的な IPQoS の構成計画のタスクマップ」を参照してください。

ルーター上での差別化サービスの提供

差別化サービスを提供するには、26 ページの「diffserv ネットワークのハードウェア計画」の説明に従って、ネットワークトポロジに Diffserv 対応ルーターを含める必要があります。ルーター上で Diffserv を構成し、ルーターのファイルを更新する実際の手順は、このガイドの扱う範囲ではありません。

このセクションでは、ネットワーク上のさまざまな IPQoS 対応システムおよび Diffserv ルーター間で、転送情報を調整する一般的な手順を説明します。

最初に、ネットワークのすべての IPQoS 対応システムの構成ファイルを確認し、さまざまな QoS ポリシーで使用されているコードポイントを確認します。

コードポイント、およびコードポイントを適用するシステムとクラスを一覧表示します。異なる領域で同じコードポイントを使用できますが、同じマークが付けられたクラス間の優先度を決めるには、IPQoS 構成ファイル内に `precedence` セレクタなどほかの条件を指定してください。

たとえば、この章の手順で使用するネットワーク例の場合、次のコードポイント表を作成できます。

システム	クラス	PHB	DS コードポイント
`Goldweb`	`video`	EF	46 (101110)
`Goldweb`	`goldweb`	AF11	10 (001010)
`Userweb`	`webout`	AF12	12 (001100)
`BigAPPS`	`smtp`	AF13	14 (001110)
`BigAPPS`	`news`	AF18	18 (010010)
`BigAPPS`	`ftp` 適合トラフィック	AF22	20 (010100)
`BigAPPS`	`ftp` 不適合トラフィック	AF31	26 (011010)

ネットワークの IPQoS 構成ファイルから得たコードポイントを特定したら、Diffserv ルーターの適切なファイルに追加します。これらのコードポイントは、ルーターの Diffserv スケジューリングメカニズムの構成に役立ちます。詳しくは、ルーターの製造元のドキュメントおよび Web サイトを参照してください。

♦ ♦ ♦ 第 4 章

4 IPQoS の起動と保守のタスク

この章では、IPQoS 構成ファイルを有効化する方法および IPQoS 関連のイベントを記録する方法について説明します。内容は次のとおりです。

- 73 ページの「IPQoS の管理」
- 76 ページの「IPQoS エラーメッセージのトラブルシューティング」

注記 - IPQoS 機能は、将来のリリースで削除される可能性があります。代わりに、同様の帯域幅リソース制御機能をサポートしている、`dladm`、`flowadm`、および関連コマンドを使用してください。詳細は、『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理 』を参照してください。

IPQoS の管理

このセクションでは、Oracle Solaris システムで IPQoS を起動および保守する方法について説明します。これらのタスクを開始する前に、43 ページの「QoS ポリシーの定義のタスクマップ」の説明に従って、IPQoS 構成ファイルを完成しておく必要があります。このセクションでは、次のタスクについて説明します。

- 74 ページの「`ipqos` パッケージを追加する方法」
- 74 ページの「`ipqos` サービスを開始する方法」
- 75 ページの「ブート時に IPQoS メッセージを記録する方法」

▼ ipqos パッケージを追加する方法

ipqos パッケージは、デフォルトではすべての構成で追加されません。この手順では、このパッケージを追加する方法について説明します。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **IPQoS パッケージがまだインストールされていないことを確認します。**

```
# pkg list ipqos
pkg list: no packages matching 'ipqos' installed
```

3. **IPS パッケージリポジトリに AI パッケージが含まれていることを確認します。**

```
# pkg list -a ipqos
NAME (PUBLISHER)                          VERSION                    IFO
system/network/ipqos                      0.5.11-0.175.2.0.0.26.2    i--
```

4. **AI パッケージをインストールします。**

```
# pkg install system/network/ipqos
           Packages to install:  1
       Create boot environment: No
Create backup boot environment: No
            Services to change:  1

DOWNLOAD                        PKGS       FILES    XFER (MB)   SPEED
Completed                        1/1       32/32      0.1/0.1  546k/s
```

▼ ipqos サービスを開始する方法

IPQoS 構成ファイルを変更したら、ipqos SMF サービスを開始する必要があります。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **/etc/inet/ipqosinit.conf ファイルが適切に変更されていることを確認します。**

3. **ipqos サービスが実行中かどうかを判断します。**

```
# svcs svc:/network/ipqos
STATE          STIME    FMRI
disabled       Mar_11   svc:/network/ipqos:default
```

4. **ipqos サービスを有効化または再開します。**

   - **ipqos サービスが無効な場合は、開始します。**

     ```
     # svcadm enable svc:/network/ipqos
     ```

   - **ipqos サービスが有効な場合は、無効化または再有効化します。**

     これらの手順では、サービスの開始時に新しい構成ファイルを使用します。

     ```
     # svcadm disable svc:/network/ipqos
     # svcadm enable svc:/network/ipqos
     ```

5. **新規 IPQoS 構成のテストおよびデバッグを行います。**

   UNIX ユーティリティーを使用して、IPQoS の動作を追跡し、IPQoS 実装に関する統計を収集します。この情報は、構成が予想どおりに機能するかを判断するのに役立ちます。

参照
- IPQoS モジュールがどのように機能するかに関する統計は、84 ページの「統計情報の収集」を参照してください。
- `ipqosconf` メッセージを記録するには、75 ページの「ブート時に IPQoS メッセージを記録する方法」を参照してください。

▼ ブート時に IPQoS メッセージを記録する方法

IPQoS ブート時のメッセージを記録するには、`/etc/syslog.conf` ファイルを変更する必要があります。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **`/etc/syslog.conf` ファイルの最後エントリとして、次のテキストを追加します。**

   ```
   user.info                 /var/adm/messages
   ```

   列の区切りは、空白ではなくタブを使用してください。

このエントリでは、IPQoS により生成されたブート時のメッセージがすべて /var/adm/messages ファイルに記録されます。

3. **システムをリブートして設定を適用します。**

例 4-1 /var/adm/messages からの IPQoS 出力

システムのリブート後に /var/adm/messages を表示すると、出力に次の例のような IPQoS ロギングメッセージが含まれることがあります。

```
May 14 10:44:33 ipqos-14 ipqosconf: [ID 815575 user.info]
 New configuration applied.
May 14 10:44:46 ipqos-14 ipqosconf: [ID 469457 user.info]
Current configuration saved to init file.
May 14 10:44:55 ipqos-14 ipqosconf: [ID 435810 user.info]
Configuration flushed.
```

次の例のような IPQoS エラーメッセージが表示されることもあります。

```
May 14 10:56:47 ipqos-14 ipqosconf: [ID 123217 user.error]
 Missing/Invalid config file fmt_version.
May 14 10:58:19 ipqos-14 ipqosconf: [ID 671991 user.error]
No ipgpc action defined.
```

これらのエラーメッセージについては、表4-1「IPQoS のエラーメッセージ」を参照してください。

IPQoS エラーメッセージのトラブルシューティング

次の表に、IPQoS によって生成されるエラーメッセージと、考えられる解決方法を示します。

表 4-1 IPQoS のエラーメッセージ

エラーメッセージ	説明	解決方法
Undefined action in parameter *parameter-name's* action *action-name*	*parameter-name* に指定したアクション名が IPQoS 構成ファイル内に存在しません。	アクションを作成するか、パラメータ内の別の既存のアクションを参照します。
Action *action-name* involved in cycle	IPQoS 構成ファイル内の *action-name* はアクション循環の一部です。これは IPQoS では許可されません。	アクション循環を決定します。次に、IPQoS 構成ファイルから循環参照の 1 つを削除します。
Action *action-name* isn't referenced by any other actions	ipgpc アクション以外で、ほかの定義済みアクションにより参照されないアクション定義が IPQoS 構成	参照されていないアクションを削除します。または別のアクションに現在参照されていないアクションを参照させます。

エラーメッセージ	説明	解決方法
	内にあります。これは IPQoS では許可されません。	
`Missing/Invalid config file fmt_version`	構成ファイルの書式がファイルの最初のエントリに指定されていません。これは IPQoS では必須です。	48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」の説明に従って、書式のバージョンを追加します。
`Unsupported config file format version`	IPQoS がサポートしない書式のバージョンが、構成ファイル内で指定されています。	書式のバージョンを、Solaris 9 9/02 以降のリリースの IPQoS で必要な `fmt_version 1.0` に変更します。
`No ipgpc action defined.`	構成ファイル内で、`ipgpc` クラシファイアのアクションが定義されていません。これは IPQoS では必須です。	`How to Create the IPQoS Configuration File and Define Traffic Classes`のように48 ページの「IPQoS 構成ファイルを作成し、トラフィッククラスを定義する方法」のアクションを定義します。
`Can't commit a null configuration`	`ipqosconf -c` を実行して空の構成をコミットしようとしました。IPQoS は空の構成を許可しません。	構成ファイルを確実に適用してから構成をコミットします。手順については、74 ページの「`ipqos` サービスを開始する方法」を参照してください。
`Invalid CIDR mask on line` *line-number*	構成ファイル内で、CIDR マスクの IP アドレスとして無効なアドレスを使用しました。	マスク値を 1–32 (IPv4 の場合) および 1–128 (IPv6 の場合) の範囲内の値に変更します。
`Address masks aren't allowed for host names line` *line-number*	構成ファイル内で、ホスト名の CIDR マスク値を定義しました。これは IPQoS では許可されません。	マスクを削除するか、あるいはホスト名を IP アドレスに変更します。
`Invalid module name line` *line-number*	構成ファイル内のアクション文に無効なモジュール名を指定しました。	モジュール名のスペルに入力ミスがないか確認します。IPQoS モジュールのリストについては、表6-5「IPQoS モジュール」を参照してください。
`ipgpc action has incorrect name line` *line-number*	構成ファイル内で `ipgpc` アクションの名前が、必須の `ipgpc.classify` ではありません。	アクション名を `ipgpc.classify` に変更します。
`Second parameter clause not supported line` *line-number*	構成ファイル内で、単一のアクションに対し 2 つのパラメータ句を指定しました。これは IPQoS では許可されません。	アクションのパラメータすべてを結合して、単一のパラメータ句にします。
`Duplicate named action`	構成ファイル内で、2 つのアクションの名前が同じです。	どちらかのアクションの名前を変更するか、あるいは削除します。
`Duplicate named filter/class in action` *action-name*	同じアクション内の 2 つのフィルタまたは 2 つのクラスの名前が同じです。これは IPQoS 構成ファイルでは許可されません。	どちらかのフィルタまたはクラスの名前を変更するか、あるいは削除します。

エラーメッセージ	説明	解決方法
`Undefined class in filter` *filter-name* `in action` *action-name*	フィルタが、構成ファイル内のアクションで定義されていないクラスを参照します。	クラスを作成するか、あるいは既存のクラスへの参照に変更します。
`Undefined action in class` *class-name* `action` *action-name*	クラスが、構成ファイル内で定義されていないアクションを参照します。	アクションを作成するか、あるいは既存のアクションへの参照に変更します。
`Invalid parameters for action` *action-name*	構成ファイル内のパラメータのどれかが無効です。	指名されたアクションが呼び出すモジュールについては、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」のモジュールエントリを参照してください。または、`ipqosconf`(1M) のマニュアルページを参照してください。
`Mandatory parameter missing for action` *action-name*	アクションの必須パラメータが構成ファイル内に定義されていません。	指名されたアクションが呼び出すモジュールについては、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」のモジュールエントリを参照してください。または、`ipqosconf`(1M) のマニュアルページを参照してください。
`Max number of classes reached in ipgpc`	IPQoS 構成ファイルの `ipgpc` アクションに、許可される数を超えたクラスを指定しました。最大数は 10007 です。	構成ファイルを確認して、不要なクラスを削除します。または、`/etc/system` ファイルにエントリ `ipgpc_max_classes` *class-number* を追加して最大クラス数を増やすこともできます。
`Max number of filters reached in action ipgpc`	IPQoS 構成ファイルの `ipgpc` アクションに、許可される数を超えたフィルタを指定しました。最大数は 10007 です。	構成ファイルを確認して、不要なフィルタを削除します。または、`/etc/system` ファイルにエントリ `ipgpc_max_filters` *filter-number* を追加して最大フィルタ数を増やすこともできます。
`Invalid/missing parameters for filter` *filter-name* `in action ipgpc`	構成ファイル内で、フィルタ *filter-name* に無効なパラメータが指定されているか、あるいはパラメータが不足しています。	有効なパラメータのリストについては、`ipqosconf`(1M) のマニュアルページを参照してください。
`Name not allowed to start with '!', line` *line-number*	アクション、フィルタ、またはクラス名の最初に感嘆符 (!) を記述しました。IPQoS ファイルでは感嘆符は許可されていません。	感嘆符を削除するか、あるいは、アクション、クラス、またはフィルタの名前を変更します。
`Name exceeds the maximum name length line` *line-number*	構成ファイル内のアクション、クラス、またはフィルタの名前が、最大長の 23 文字を超えています。	アクション、クラス、またはフィルタの名前を短くします。
`Array declaration line` *line-number* `is invalid`	構成ファイル内で、*line-number* 行のパラメータの配列宣言が無効です。	無効な配列を持つ `action` 文が呼び出す配列宣言の正しい構文については、87 ページの「IPQoS アーキテクチャー

エラーメッセージ	説明	解決方法
		と Diffserv モデル」を参照してください。または、**ipqosconf**(1M) のマニュアルページを参照してください。
`Quoted string exceeds line,` *line-number*	文字列の最後の閉じ引用符が同一行に存在しません。これは構成ファイルでは必須です。	引用符で囲まれた文字列を、構成ファイルの同一行内に収めます。
`Invalid value, line` *line-number*	構成ファイルの *line-number* に、パラメータとしてサポートされない値が指定されています。	`action` 文が呼び出すモジュールの許容値については、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」のモジュールの説明を参照してください。または、**ipqosconf**(1M) のマニュアルページを参照してください。
`Unrecognized value, line` *line-number*	構成ファイルの *line-number* に、パラメータとしてサポートされない列挙値が指定されています。	パラメータの列挙値が適正であるかどうかを確認します。認識されない行番号を持つ `action` 文が呼び出すモジュールについては、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」を参照してください。または、**ipqosconf**(1M) のマニュアルページを参照してください。
`Malformed value list line` *line-number*	構成ファイルの *line-number* で指定された列挙が、仕様構文に適合しません。	間違った形式の値リストを持つ `action` 文が呼び出すモジュールの正しい構文については、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」のモジュールの説明を参照してください。または、**ipqosconf**(1M) のマニュアルページを参照してください。
`Duplicate parameter line` *line-number*	重複したパラメータが *line-number* に指定されています。これは構成ファイルでは許可されません。	重複したパラメータのどちらかを削除します。
`Invalid action name line` *line-number*	構成ファイルの *line-number* のアクションの名前で、定義済みの名前「continue」または「drop」を使用しています。	定義済みの名前を使用しないよう、アクションの名前を変更します。
`Failed to resolve src/dst host name for filter at line` *line-number*`, ignoring filter`	構成ファイル内で、あるフィルタ用に定義された発信元または着信先アドレスを、`ipqosconf` が解釈処理できません。このため、このフィルタは無視されます。	フィルタが重要な場合、あとで構成の適用を試みます。
`Incompatible address version line` *line-number*	*line-number* 上の IP アドレスのバージョンが、すでに指定済みの IP アドレスのバージョンまたは `ip_`	競合する 2 つのエントリを変更して、互換性を持たせます。

エラーメッセージ	説明	解決方法
	version パラメータと互換性がありません。	
Action at line *line-number* has the same name as currently installed action, but is for a different module	システムの IPQoS 構成内にすでに存在するアクションのモジュールを変更しようとしました。これは許可されません。	現行の構成をフラッシュしてから、新しい構成を適用します。

♦ ♦ ♦ 第 5 章

5 フローアカウンティングの使用と統計情報の収集のタスク

この章では、IPQoS システムによって処理されるトラフィックに関して、アカウンティング情報と統計情報を取得する方法について説明します。内容は次のとおりです。

- 81 ページの「トラフィックフローに関する情報の記録」
- 84 ページの「統計情報の収集」

注記 - IPQoS 機能は、将来のリリースで削除される可能性があります。代わりに、同様の帯域幅リソース制御機能をサポートしている、`dladm`、`flowadm`、および関連コマンドを使用してください。詳細は、『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理 』を参照してください。

トラフィックフローに関する情報の記録

IPQoS `flowacct` モジュールを使用して、トラフィックフロー (発信元および着信先のアドレスなど) 、フローのパケットの数、および同様のデータの情報を収集します。フローに関する情報を蓄積して記録するプロセスのことを「*フローアカウンティング*」と呼びます。

特定のクラスのトラフィックに関するフローアカウンティングの結果は、*フローレコード*というテーブルに記録されます。各フローレコードは、一連の属性から構成されます。これらの属性には、特定のクラスの一定時間のトラフィックフローに関するデータが格納されます。`flowacct` 属性のリストについては、表6-4「`flowacct` レコードの属性」 を参照してください。

フローアカウンティングは、サービスレベル契約 (SLA) に定義されているとおりに顧客に課金する場合に、特に役立ちます。また、フローアカウンティングを使って、重要なアプリケーションのフロー統計情報を取得することもできます。このセクションでは、`flowacct` を Oracle Solaris

拡張アカウンティング機能と組み合わせて、トラフィックフローに関するデータを取得するためのタスクについて説明します。

詳細は、次のソースを参照してください。

- IPQoS 構成ファイルで `flowacct` パラメータを割り当てる方法については、56 ページの「IPQoS 構成ファイル内でクラスのアカウンティングを有効にする方法」を参照してください。
- IPQoS 構成ファイル内の `flowacct` のアクション文の作成手順については、68 ページの「IPQoS 構成ファイル内でフロー制御を構成する方法」を参照してください。
- `flowacct` がどのように機能するかについては、88 ページの「クラシファイアモジュール」を参照してください。
- 技術的な情報については、`flowacct`(7ipp) のマニュアルページを参照してください。

▼ フローアカウンティングデータ用のファイルを作成する方法

`flowacct` アクションを IPQoS 構成ファイルに追加する前に、`flowacct` モジュールからフローレコードのファイルを作成する必要があります。`acctadm` では、基本属性および拡張属性のいずれかをファイルに記録できます。すべての `flowacct` 属性のリストについては、表6-4「`flowacct` レコードの属性」を参照してください。詳細は、`acctadm`(1M) のマニュアルページを参照してください。

1. **管理者になります。**

   詳細は、『Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティー保護 』の「割り当てられている管理権利の使用」を参照してください。

2. **基本フローアカウンティングファイルを作成します。**

   次の例で、例3-1「プレミアム Web サーバー用 IPQoS 構成ファイルの例」で構成されるプレミアム Web サーバー用の基本的なフローアカウンティングファイルを作成する方法を示します。

   ```
   # /usr/sbin/acctadm -e basic -f /var/ipqos/goldweb/account.info flow
   ```

`acctadm -e`	`acctadm` を `-e` オプションを指定して呼び出します。`-e` オプションによって、あとに続く引数が有効になる
`basic`	`flowacct` の 8 つの基本属性のデータだけがファイルに記録されることを示す

`/var/ipqos/goldweb/account.info`	`flowacct` から得られるフローレコードを格納するファイルの絶対パス名を示す
`flow`	`acctadm` にフローアカウンティングを有効にするよう指示する

3. **引数を指定しないで `acctadm` と入力し、IPQoS システムのフローアカウンティングに関する情報を表示します。**

   `acctadm` 出力は、次の例のようになります。

```
Task accounting: inactive
       Task accounting file: none
     Tracked task resources: none
   Untracked task resources: extended
         Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
Untracked process resources: extended,host,mstate
            Flow accounting: active
       Flow accounting file: /var/ipqos/goldweb/account.info
     Tracked flow resources: basic
   Untracked flow resources: dsfield,ctime,lseen,projid,uid
```

   最後の 4 つのエントリ以外はすべて、Oracle Solaris のリソースマネージャー機能で使用されます。IPQoS に固有のエントリは次のとおりです。

`Flow accounting: active`	フローアカウンティングが有効になっていることを示す
`Flow accounting file: /var/ipqos/goldweb/account.info`	現在のフローアカウンティングファイルの名前を示す。
`Tracked flow resources: basic`	基本フロー属性だけが記録されることを示す
`Untracked flow resources: dsfield,ctime,lseen,projid,uid`	基本フロー属性だけが記録されることを示す

4. **(オプション) 拡張属性をアカウンティングファイルに追加します。**

```
# acctadm -e extended -f /var/ipqos/goldweb/account.info flow
```

5. **(オプション) 基本属性だけがアカウンティングファイルに記録されるような設定に戻します。**

```
# acctadm -d extended -e basic -f /var/ipqos/goldweb/account.info
```

`-d` オプションによって拡張アカウンティングが無効になります。

6. **フローアカウンティングファイルの内容を参照します。**

   フローアカウンティングファイルの内容を表示する手順については、『Oracle Solaris 11.2 でのリソースの管理 』の「libexacct に対する Perl インタフェース」を参照してください。

参照 拡張アカウンティング機能の詳細は、『Oracle Solaris 11.2 でのリソースの管理 』の第 4 章「拡張アカウンティングについて」を参照してください。

次の手順

- IPQoS 構成ファイル内に `flowacct` パラメータを定義するには、56 ページの「IPQoS 構成ファイル内でクラスのアカウンティングを有効にする方法」を参照してください。
- `acctadm` で作成されたファイル内のデータを出力するには、『Oracle Solaris 11.2 でのリソースの管理 』の「libexacct に対する Perl インタフェース」を参照してください。

統計情報の収集

`kstat` コマンドを使用すると、IPQoS モジュールから統計情報を生成できます。

`/bin/kstat -m` *ipqos-module-name*

表6-5「IPQoS モジュール」 に示されている、有効な IPQoS モジュール名であればどれでも指定できます。たとえば、`dscpmk` マーカーによって生成される統計情報を表示するには、次のコマンドを使用します。

```
/bin/kstat -m dscpmk
```

技術的な情報については、`kstat`(1M) のマニュアルページを参照してください。

例 5-1 IPQoS 用の `kstat` 統計

次の例では、`kstat` を実行して `flowacct` モジュールに関する統計情報を取得した場合に予想される結果の一例について説明します。

```
# kstat -m flowacct
module: flowacct                        instance: 3
name:   Flowacct statistics             class:    flacct
        bytes_in_tbl                    84
        crtime                          345728.504106363
```

```
        epackets                        0
        flows_in_tbl                    1
        nbytes                          84
        npackets                        1
        snaptime                        345774.031843301
        usedmem                         256
```

`class: flacct` トラフィックフローが属するクラスの名前 (この例では `flacct`) を示す。

`bytes_in_tbl` フローテーブルの総バイト数。フローテーブルの総バイト数とは、フローテーブルに現在格納されているすべてのフローレコードの合計バイト数。このフローテーブルの総バイト数は 84 である。テーブルにフローがない場合、`bytes_in_tbl` の値は 0 になる

`crtime` この `kstat` 出力が作成された最も最近の時間

`epackets` 処理中にエラーが発生したパケットの数 (この例では 0)

`flows_in_tbl` フローテーブルのフローレコード数 (この例では 1)。テーブルにレコードがない場合、`flows_in_tbl` の値は 0 になる

`nbytes` この `flowacct` アクションのインスタンスで表示される合計バイト数 (この例では 84)。フローテーブルに現在格納されているバイトを含む値。この値には、タイムアウトになり、フローテーブルに現在は含まれていない値も含まれる

`npackets` この `flowacct` アクションのインスタンスで表示される合計パケット数 (この例では 1)。`npackets` には、フローテーブルに現在あるパケットが含まれる。`npackets` には、タイムアウトになり、フローテーブルに現在は含まれていないパケットも含まれる。

`usedmem` この `flowacct` インスタンスで保持されているフローテーブルが使用しているメモリーのバイト数。この例では、`usedmem` の値は 256。フローテーブルにフローレコードがまったく存在しない場合、`usedmem` の値は 0 になる

♦ ♦ ♦ 第 6 章

6 IPQoS の詳細のリファレンス

この章では、次の IPQoS の内容について詳しく説明します。

- 87 ページの「IPQoS アーキテクチャーと Diffserv モデル」
- 101 ページの「IPQoS 構成ファイル」

詳細は、次のリソースを参照してください。

- 概要は、第1章「IPQoS の概要」を参照してください。
- 計画情報については、第2章「IPQoS 対応ネットワークの計画」を参照してください。
- IPQoS の構成手順については、第3章「IPQoS 構成ファイルの作成のタスク」を参照してください。

注記 - IPQoS 機能は、将来のリリースで削除される可能性があります。代わりに、同様の帯域幅リソース制御機能をサポートしている、`dladm`、`flowadm`、および関連コマンドを使用してください。詳細は、『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理 』を参照してください。

IPQoS アーキテクチャーと Diffserv モデル

このセクションでは、IPQoS アーキテクチャーとこのアーキテクチャーが RFC 2475, An Architecture for Differentiated Services (http://www.ietf.org/rfc/rfc2475.txt?number=2475) で定義された差別化サービス (Diffserv) モデルを実装する方法について説明します。次に示す Diffserv モデルの要素が、IPQoS に含まれます。

- クラシファイア
- メーター
- マーカー

さらに、IPQoS には、仮想ローカルエリアネットワーク (VLAN) デバイスで使用されるフローアカウンティングモジュールと `dlcosmk` マーカーが含まれています。

クラシファイアモジュール

Diffserv モデルでは、「*クラシファイア*」は、トラフィックフローを選択して、それぞれに異なるサービスレベルを適用するためのグループに分類する作業を担当します。RFC 2475 で定義されたクラシファイアは、当初、境界ルーター用に設計されました。それとは対照的に、IPQoS クラシファイア `ipgpc` は、内部ホストからローカルネットワークへのトラフィックフローを処理するために設計されています。このため、IPQoS システムと Diffserv ルーターの両方を備えたネットワークは、より広範囲な差別化サービスを提供できます。技術情報については、`ipgpc`(7ipp) のマニュアルページを参照してください。

`ipgpc` クラシファイアは、次の機能を実行します。

1. IPQoS 対応システムの IPQoS 構成ファイルに指定された条件を満たすトラフィックフローを選択します。

   QoS ポリシーは、パケットヘッダーに存在する必要のあるさまざまな条件を定義します。これらの条件は、「*セレクタ*」と呼ばれます。 `ipgpc` クラシファイアは、これらのセレクタを、IPQoS システムから受信したパケットのヘッダーと比較して、一致するパケットをすべて選択します。
2. パケットフローを、IPQoS 構成ファイルの定義に従い、同じ特性を持つネットワークトラフィックである「*クラス*」に分類します。
3. パケットの差別化サービス (DS) フィールドの値を調べ、差別化サービスコードポイント (DSCP) の存在を確認します

   DSCP は、受信したトラフィックに送信側によって転送動作のマークが付けられているかどうかを示します。
4. 特定クラスのパケットに関して、IPQoS 構成ファイル内で次に指定されているアクションを調べます。
5. パケットを、IPQoS 構成ファイルで指定された次の IPQoS モジュールに渡すか、あるいはネットワークストリームに戻します。

クラシファイアの概要は、16 ページの「クラシファイア (ipgpc) の概要」を参照してください。IPQoS 構成ファイルでクラシファイアを呼び出すには、101 ページの「IPQoS 構成ファイル」を参照してください。

IPQoS セレクタ

`ipgpc` クラシファイアは、IPQoS 構成ファイルの `filter` 句で使用可能なさまざまなセレクタをサポートします。フィルタを定義するときには、特定クラスのトラフィック取得に必要な最小限の

セレクタを使用してください。定義するフィルタの数が、IPQoS のパフォーマンスに影響を与える可能性があります。

次の表に、ipgpc で使用できるセレクタを示します。

表 6-1 IPQoS クラシファイアで利用可能なフィルタセレクタ

セレクタ	引数	選択される情報
saddr	IP アドレス番号。	発信元アドレス
daddr	IP アドレス番号。	着信先アドレス
sport	ポート番号またはサービス名。/etc/services の定義に従う。	トラフィッククラスの発信元ポート
dport	ポート番号またはサービス名。/etc/services の定義に従う。	トラフィッククラスの着信先ポート
protocol	プロトコル番号またはプロトコル名。/etc/protocols の定義に従う。	このトラフィッククラスが使用するプロトコル
dsfield	0 - 63 の値を持つ DS コードポイント (DSCP)	DSCP。パケットに適用される転送動作を定義する。このパラメータを指定した場合は、dsfield_mask パラメータも指定すること
dsfield_mask	0 - 255 の値を持つビットマスク	dsfield セレクタと組み合わせて使用。dsfield_mask は、dsfield セレクタに適用して、ビットのどれが一致するかを決定する。
if_name	インタフェース名。	特定クラスの着信トラフィックまたは発信トラフィックで使用されるインタフェース
user	選択する UNIX ユーザー ID の番号またはユーザー名。パケットにユーザー ID またはユーザー名が存在しない場合、デフォルトの –1 が使用される	アプリケーションに指定されるユーザー ID
projid	選択するプロジェクト ID の番号	アプリケーションに付加されるプロジェクト ID
priority	優先順位の番号。もっとも低い優先順位は 0。	このクラスのパケットに与えられる優先順位。優先順位は、同じクラスに複数存在するフィルタの重要度の順位付けに使用される
direction	指定可能な値:	IPQoS マシン上のパケットフローの方向
	LOCAL_IN	ローカルシステムから IPQoS システムへの入力トラフィック
	LOCAL_OUT	ローカルシステムから IPQoS システムへの出力トラフィック
	FWD_IN	転送される入力トラフィック
	FWD_OUT	転送される出力トラフィック

セレクタ	引数	選択される情報
precedence	優先度の値。もっとも高い優先度は 0。	同一優先順位のフィルタの順序付けに使用される。
ip_version	V4 または V6	パケットにより使用されるアドレス指定スキーム (IPv4 または IPv6)

メーターモジュール

「*メーター*」はフローの転送速度をパケット単位で追跡します。このメーターは、構成されているパラメータにパケットが一致するかどうかを決定します。メーターモジュールは、パケットサイズ、構成されたパラメータ、およびフロー速度に基づき、パケットの次のアクションをアクションセットの中から決定します。

メーターには 2 つのメータリングモジュール、すなわち tokenmt および tswtclmt があります。モジュールの構成は、IPQoS 構成ファイルで行います。モジュールのどちらか一方または両方をクラスに構成できます。
メータリングモジュールを構成する際、速度に関する 2 つのパラメータを定義できます。

- committed-rate – 特定クラスのパケットに容認可能な転送速度を bps で定義する
- peak-rate – 特定クラスのパケットに最大限容認可能な転送速度を bps で定義する

パケットに対するメータリングアクションの結果 (outcome) は、次の 3 つのどれかになります。

- green – パケットの生成するフローは認定速度内である
- yellow – パケットの生成するフローは認定速度を超過しているが、最大速度内である
- red – パケットの生成するフローは最大速度を超過している

IPQoS 構成ファイル内で、結果ごとに異なるアクションを構成できます。

tokenmt メータリングモジュール

tokenmt モジュールは、「*トークンバケット*」を使用してフローの転送速度を測定します。tokenmt は、シングルレートメーターまたはツーレートメーターとして機能するように構成できます。tokenmt アクションインスタンスは、2 つのトークンバケットを管理します。これらのトークンバケットは、トラフィックフローが構成されたパラメータに適合するかどうかを調べます。

tokenmt(7ipp) のマニュアルページでは、IPQoS がどのようにトークンメーターパラダイムを実装するかが説明されています。

tokenmt の構成パラメータは次のとおりです。

- committed_rate – フローの認定速度を bps で指定する
- committed_burst – 認定バーストサイズをビット単位で指定する。committed_burst パラメータは、認定速度でネットワークに渡すことのできる、特定クラスの発信パケット数を定義する
- peak_rate – 最大速度を bps で指定する
- peak_burst – 最大バーストサイズまたは超過バーストサイズをビット単位で指定する。peak_burst パラメータは、トラフィッククラスに、認定速度を超過する最大バーストサイズを付与する
- color_aware – tokenmt のカラーアウェアモードを有効にする。
- color_map – DSCP 値を緑、黄、または赤にマッピングする整数配列を定義する

tokenmt をシングルレートメーターとして構成する

tokenmt をシングルレートメーターとして構成するには、IPQoS 構成ファイル内で tokenmt に peak_rate パラメータを指定しないでください。赤、緑、または黄の結果 (outcome) を識別するようにシングルレートの tokenmt インスタンスを構成するには、peak_burst パラメータを指定する必要があります。peak_burst パラメータを使用しないことによって、tokenmt が赤または緑の結果だけを識別するように構成できます。2 つの出力を持つシングルレート tokenmt の例については、例3-3「アプリケーションサーバー用サンプル IPQoS 構成ファイル」を参照してください。

tokenmt がシングルレートメーターとして機能する場合、peak_burst パラメータは実質的に超過バーストサイズです。committed_burst と peak_burst のどちらかと committed_rate は、ゼロ以外の正の整数にする必要があります。

tokenmt をツーレートメーターとして構成する

tokenmt をツーレートメーターとして構成するには、IPQoS 構成ファイル内で tokenmt アクションに peak_rate パラメータを指定します。ツーレートの tokenmt は、必ず赤、黄、および緑の 3 つの結果 (outcome) を識別します。committed_rate、committed_burst、および peak_burst パラメータは、ゼロ以外の正の整数にする必要があります。

tokenmt をカラーアウェアとして構成する

ツーレートの tokenmt をカラーアウェアとして構成するには、「カラーアウェアネス」を特に追加するパラメータを追加します。tokenmt をカラーアウェアとして構成する action 文の例を次に示します。

例 6-1 IPQoS 構成ファイル用のカラーアウェア tokenmt アクション

```
action {
    module tokenmt
    name meter1
    params {
       committed_rate 4000000
       peak_rate 8000000
       committed_burst 4000000
       peak_burst 8000000
       global_stats true
       red_action_name continue
       yellow_action_name continue
       green_action_name continue
       color_aware true
       color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
    }
}
```

color_aware パラメータを true に設定することによって、カラーアウェアを有効にできます。カラーアウェアにした tokenmt メーターは、以前の tokenmt アクションによってパケットが赤、黄、または緑にマーキング済みであるものと見なします。カラーアウェアの tokenmt は、ツーレートメーター用のパラメータに加え、パケットヘッダー内の DSCP も使用してパケットを評価します。

color_map パラメータは、パケットヘッダーの DSCP がマッピングされる配列を含みます。次の color_map 配列について説明します。

```
color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
```

DSCP が 0~20 および 22 のパケットは緑にマッピングされます。DSCP が 21 および 23~42 のパケットは赤にマッピングされます。DSCP が 43~63 のパケットは黄にマッピングされます。tokenmt は、デフォルトのカラーマップを格納します。ただし、このデフォルトは必要に応じて color_map パラメータを使用して変更できます。

*color*_action_name パラメータでは、continue を指定するとパケットの処理を完了できます。また、たとえば yellow_action_name mark22 のように、引数を指定してパケットをマーカーアクションに送信することもできます。

tswtclmt メータリングモジュール

tswtclmt メータリングモジュールは、時間ベースの「*速度エスティメータ*」を使用して、トラフィッククラスの平均帯域幅を見積もります。tswtclmt は必ず 3 つの結果 (outcome) を識別するメーターとして機能します。速度エスティメータは、フローの到着速度の見積もりを提供します。この速度は、一定期間すなわち「*時間ウィンドウ*」内の、トラフィックストリームの実行帯域幅の平均を見積もります。

tswtclmt を構成するには、次のパラメータを使用します。

- committed_rate – 認定速度を bps で指定する
- peak_rate – 最大速度を bps で指定する
- window – 時間ウィンドウをミリ秒で定義する。この時間ウィンドウに対して平均帯域幅の履歴が記録される

tswtclmt の技術的な詳細については、tswtclmt(7ipp) のマニュアルページを参照してください。

マーカーモジュール

IPQoS には 2 つのマーカーモジュール、すなわち dscpmk および dlcosmk が含まれます。このセクションでは、両方のマーカーの使用方法を説明します。dlcosmk は VLAN デバイスを使用する IPQoS システムでだけ利用可能であるため、通常は dscpmk を使用してください。

これらのモジュールの技術的情報については、dscpmk(7ipp) および dlcosmk(7ipp) のマニュアルページを参照してください。

パケット転送での dscpmk マーカーの使用

マーカーは、クラシファイアモジュールまたはメータリングモジュールによって処理されたあとのトラフィックフローを受け取ります。マーカーは、転送動作をトラフィックにマークします。転送動作とは、フローが IPQoS システムから送出されたあと、フローに対して行われるアクションです。トラフィッククラスに対して実行される転送動作は、「*ホップ単位動作 (PHB)*」に定義されます。PHB はトラフィッククラスに優先順位を割り当てます。これは、そのクラスのフローに割り当てられる、ほかのトラフィッククラスに対する相対的な優先度です。PHB は、IPQoS システムの隣接する

ネットワーク上での転送動作だけを制御します。詳細は、21 ページの「ホップ単位動作」を参照してください。

「*パケット転送*」とは、特定クラスのトラフィックを、ネットワーク上の次の宛先へ送信するプロセスを指します。IPQoS システムなどのホストの場合、パケットはホストからローカルネットワークストリームへ転送されます。Diffserv ルーターの場合、パケットはローカルネットワークからルーターの次のホップへ転送されます。

マーカーは、パケットヘッダー内の DS フィールドに、IPQoS 構成ファイル内で定義された転送動作のマークを付けます。以後、IPQoS システムおよびあとに続く Diffserv 対応システムは、マークが変更されないかぎり、DS フィールド内の指示に従ってトラフィックを転送します。PHB を割り当てるため、IPQoS システムは、パケットヘッダーの DS フィールドの値をマークします。この値は、DSCP (Differentiated Services Codepoint) と呼ばれます。Diffserv アーキテクチャーは、2 種類の転送動作、すなわち EF および AF を定義しており、各転送動作はそれぞれ異なる DSCP を使用します。DSCP の概要については、21 ページの「DS コードポイント」を参照してください。

IPQoS システムは、トラフィックフローの DSCP を読み取り、ほかの送信トラフィックフローに対する相対的な優先度を評価します。次に IPQoS システムは、並行するトラフィックフローすべての優先順位を定め、各フローを優先順位に従ってネットワーク上に送出します。

Diffserv ルーターは、送信トラフィックフローを受け取り、パケットヘッダー内の DS フィールドを読み取ります。DSCP を使用すると、ルーターで現在のトラフィックフローに優先順位を付け、スケジュールを設定できます。ルーターは、PHB で指示された優先順位に従って各フローを転送します。あとに続くホップ上の Diffserv 対応システムも同じ PHB を認識する場合を除いて、ネットワークの境界ルーターを越えて PHB を適用することはできません。

完全優先転送 (Expedited Forwarding、EF) PHB

完全優先転送 (EF) は、推奨される EF コードポイント 46 (101110) の付いたパケットが、ネットワークに送出されるときに、可能なかぎり最良の扱いを受けることを保証します。完全優先転送は、しばしば専用回線に例えられます。コードポイント 46 (101110) を持つパケットには、宛先に向かう途中、すべての Diffserv ルーターによる優先待遇が保証されます。

相対的優先転送 (Assured Forwarding、AF) PHB

相対的優先転送 (AF) では、4 つの異なるクラスの転送動作をマーカーに指定できます。次の表に、クラス、各クラスに指定できる 3 つのドロップ優先度、および各優先度に対応する推奨 DSCP を示します。各 DSCP は、AF 値 (10 進数値およびバイナリ値) で表されます。

表 6-2 相対的優先転送のコードポイント

	クラス 1	クラス 2	クラス 3	クラス 4
低ドロップ優先度	AF11 = 10 (001010)	AF21 = 18 (010010)	AF31 = 26 (011010)	AF41 = 34 (100010)
中ドロップ優先度	AF12 = 12 (001100)	AF22 = 20 (010100)	AF32 = 28 (011100)	AF42 = 36 (100100)
高ドロップ優先度	AF13 = 14 (001110)	AF23 = 22 (010110)	AF33 = 30 (011110)	AF43 = 38 (100110)

AF コードポイントは、各トラフィッククラスに差別化転送動作を提供する際のガイドとして、すべての Diffserv 対応システム上で使用できます。

これらのパケットが Diffserv ルーターに達すると、ルーターはパケットのコードポイントを、キュー内のほかのトラフィックの DSCP とともに評価します。次にルーターは、利用可能な帯域幅、およびパケットの DSCP により割り当てられた優先順位に応じて、パケットを転送またはドロップします。EF PHB の付いたパケットは、どの AF PHB の付いたパケットよりも広い帯域幅の使用が保証されます。

ネットワーク上の IPQoS システムと Diffserv ルーターとの間でパケットのマーキングを合致させて、パケットが意図したとおりに転送されるようにしてください。たとえば、ネットワーク上の IPQoS システムがパケットにコードポイント AF21 (010010)、AF13 (001110)、AF43 (100110)、および EF (101110) を付けるとします。この場合、AF21、AF13、AF43、および EF DSCP を、Diffserv ルーターの適切なファイルに追加する必要があります。

AF PHB の設定および使用する機器での DS コードポイントの設定手順については、ルーター製造元のドキュメントを参照してください。

マーカーへの DSCP の設定

DSCP の長さは 6 ビットです。DS フィールドの長さは 1 バイトです。DSCP を定義すると、マーカーは、DS コードポイントでパケットヘッダーの最初の 6 つの重みビットをマークします。残りの 2 ビットは、使用されません。

DSCP を定義するには、マーカーアクション文の中で次のパラメータを使用します。

`dscp_map{0-63:`*DS-name tcodepoint*`}`

dscp_map パラメータは、(DSCP) 値を使用して生成する 64 要素の配列です。dscp_map は、dscpmk マーカーによって着信 DSCP を発信 DSCP にマップするために使用されます。

DSCP 値は、10 進表記で dscp_map に指定する必要があります。たとえば、EF コードポイント 101110 は 10 進数値 46 に変換する必要があり、その結果 dscp_map{0-63:46} になります。AF コードポイントの場合、表6-2「相対的優先転送のコードポイント」で示されるさまざまなコードポイントを、dscp_map で使用するために 10 進数表記に変換する必要があります。

VLAN デバイスでの dlcosmk マーカーの使用

dlcosmk マーカーモジュールは、データグラムの MAC ヘッダー内に転送動作をマークします。VLAN インタフェースを持つ IPQoS システムでだけ、dlcosmk を使用できます。

dlcosmk は、*VLAN タグ*と呼ばれる 4 バイトを MAC ヘッダーに追加します。VLAN タグには、IEEE 801.D 標準に定義されている 3 ビットのユーザー優先順位値が含まれます。VLAN を認識する Diffserv 対応スイッチは、データグラム内のユーザー優先順位フィールドを読み取ることができます。801.D ユーザー優先順位値は、商用スイッチと互換性のあるサービスクラス (CoS) マークを実装します。

次の表のサービスマークのクラスを定義することによって、dlcosmk マーカーアクションのユーザー優先順位値を使用できます。

表 6-3 801.D ユーザー優先順位値

サービスクラス	定義
0	ベストエフォート
1	背景
2	予備
3	エクセレントエフォート
4	制御された負荷
5	応答時間 100ms 未満のビデオ
6	応答時間 10ms 未満のビデオ
7	ネットワーク制御

詳細は、dlcosmk(7ipp) のマニュアルページを参照してください。

VLAN デバイスを持つシステムでの IPQoS 構成

このセクションでは、VLAN デバイスを持つシステムでの IPQoS の実装方法を示す、単純なネットワークのシナリオを紹介します。このシナリオには、スイッチで接続された 2 つの IPQoS システム、すなわち machine1 および machine2 が含まれます。machine1 上の VLAN デバイスの IP アドレスは 10.10.8.1、machine2 上の VLAN デバイスの IP アドレスは 10.10.8.3 です。

machine1 向けの次の IPQoS 構成ファイルは、machine2 への切り替えによる、トラフィックのマーキングの簡単な解決策を示しています。

例 6-2 VLAN デバイスを持つシステムの IPQoS 構成ファイル

```
fmt_version 1.0
action {
        module ipgpc
       name ipgpc.classify

        filter {
                name myfilter2
                daddr 10.10.8.3
                class myclass
        }

        class {
                name myclass
                next_action mark4
        }
}

action {
        name mark4
        module dlcosmk
        params {
                cos 4
                next_action continue
  global_stats true
        }
}
```

この構成では、machine2 上の VLAN デバイスを着信先とする machine1 からのすべてのトラフィックが、dlcosmk マーカーに渡されます。mark4 マーカーアクションは、CoS が 4 でクラスが myclass のデータグラムに VLAN マークを追加するように dlcosmk に指示します。ユーザー優先順位値 4 は、2 台のマシン間の切り替えによって、machine1 からの myclass トラフィックフローへの制御された負荷転送を指定しなければならないことを示します。

flowacct モジュール

IPQoS の flowacct モジュールは、トラフィックフローに関する情報を記録し、このプロセスは、*フローアカウンティング*と呼ばれます。フローアカウンティングは、顧客への課金や特定クラスへのトラフィック量の評価に使用できるデータを作成します。

フローアカウンティングは、オプションです。通常、flowacct は、メーターまたはマーカーに処理されたトラフィックフローが、ネットワークストリームへ送出される前に通る、最後のモジュールです。Diffserv モデルでの flowacct の位置の図については、図1-1「diffserv モデルの IPQoS 実装を通過するトラフィックフロー」を参照してください。技術的な情報については、flowacct(7ipp) のマニュアルページを参照してください。

フローアカウンティングを有効にするには、flowacct に加えて、Oracle Solaris の exacct アカウンティング機能および acctadm コマンドを使用する必要があります。フローアカウンティングの詳細は、第5章「フローアカウンティングの使用と統計情報の収集のタスク」を参照してください。

flowacct パラメータ

flowacct モジュールは、*フローレコード*で構成された*フローテーブル*内に、フローに関する情報を収集します。テーブル内の各エントリには、1 つのフローレコードが含まれます。フローテーブルは、表示できません。

フローレコードを測定してフローテーブルに書き込むには、IPQoS 構成ファイル内で次の flowacct パラメータを定義します。

- timer – タイムアウトしたフローをフローテーブルから削除し、acctadm により作成されたファイルに書き込む間隔を、ミリ秒単位で定義する
- timeout – パケットフローがタイムアウトするまでの非アクティブな時間を、ミリ秒単位で定義する

注記 - timer と timeout には異なる値を指定できます。

- max_limit – フローテーブルに格納可能なフローレコードの数に上限を設定する

flowacct パラメータの IPQoS 構成ファイルでの使用例については、68 ページの「IPQoS 構成ファイル内でフロー制御を構成する方法」を参照してください。

フローテーブル

flowacct モジュールは、flowacct インスタンスが認識するすべてのパケットフローを記録するフローテーブルを管理します。

フローは、flowacct の 8 タプルと呼ばれる、次のパラメータによって特定されます。

- 発信元アドレス
- 着信先アドレス
- 発信元ポート
- 着信先ポート
- DSCP
- ユーザー ID
- プロジェクト ID
- プロトコル番号

フローの 8 タプルのパラメータが変化しないかぎり、フローテーブルには 1 つのエントリだけが含まれます。max_limit パラメータにより、フローテーブルに含めることのできるエントリ数が決定されます。

フローテーブルは、IPQoS 構成ファイル内の timer パラメータに指定された間隔でスキャンされます。デフォルトは 15 秒です。IPQoS 構成ファイル内の timeout 間隔に指定された時間以上、IPQoS システムがパケットを認識しない場合、フローは「タイムアウト」します。デフォルトのタイムアウト時間は、60 秒です。タイムアウトしたエントリは、acctadm コマンドを使用して作成されたアカウンティングファイルに書き込まれます。

flowacct レコード

flowacct レコードには、次の表に示される属性が含まれています。

表 6-4 flowacct レコードの属性

属性名	属性の内容	タイプ
src-addr-*address-type*	オリジネータの発信元アドレス。*address-type* は、IPQoS 構成ファイルの指定に従い、v4 (IPv4 の場合) または v6 (IPv6 の場合) になる	Basic
dest-addr-*address-type*	パケットの着信先アドレス。*address-type* は、IPQoS 構成ファイルの指定に従い、v4 (IPv4 の場合) または v6 (IPv6 の場合) になる	Basic

属性名	属性の内容	タイプ
src-port	フローの起点となる発信元ポート	Basic
dest-port	フローの宛先となる着信先ポート番号	Basic
protocol	フローのプロトコル番号	Basic
total-packets	フロー内のパケット数	Basic
total-bytes	フロー内のバイト数	Basic
action-name	このフローを記録した flowacct アクションの名前	Basic
creation-time	flowacct がそのフローのパケットを最初に認識した時間	Extended のみ
last-seen	そのフローのパケットを最後に認識した時間	Extended のみ
diffserv-field	フローの発信パケットヘッダー内の DSCP	Extended のみ
user	アプリケーションから取得される UNIX ユーザー ID またはユーザー名	Extended のみ
projid	アプリケーションから取得されるプロジェクト ID	Extended のみ

flowacct モジュールでの acctadm の使用

acctadm コマンドを使用して、flowacct により生成されるさまざまなフローレコードを格納するファイルを作成します。acctadm は、拡張アカウンティング機能と連動して動作します。技術的な情報については、acctadm(1M) のマニュアルページを参照してください。

flowacct モジュールは、フローを観察し、フローテーブルにフローレコードを入力します。次に flowacct は、timer に指定された間隔でパラメータと属性を評価します。last_seen 値に timeout 値を加えた時間以上パケットが検出されない場合、パケットはタイムアウトします。タイムアウトしたエントリはすべて、フローテーブルから削除されます。削除されたタイムアウトエントリは、timer パラメータに指定された時間が経過するたびに、アカウンティングファイルに書き込まれます。

acctadm を呼び出して flowacct モジュールで使用するには、次の構文を使用します。

```
acctadm -e file-type -f filename flow
```

acctadm -e　　acctadm を -e オプションを指定して呼び出します。-e は、直後にタイプを指定することを示します。

file-type	収集する属性を指定します (`basic` または `extended`)。各ファイルタイプの属性の一覧については、表6-4「`flowacct` レコードの属性」を参照してください。
-ffile-name	フローレコードを格納するファイル *file-name* を作成します。
`flow`	`acctadm` を IPQoS 上で実行することを示します。

IPQoS 構成ファイル

このセクションでは、IPQoS 構成ファイル各部の詳細を説明します。IPQoS のブート時にアクティブになるポリシーは、`/etc/inet/ipqosinit.conf` ファイルに格納されています。このファイルは編集可能ですが、新しい IPQoS システムの場合、別の名前で構成ファイルを作成するのが最善の方法です。IPQoS 構成の適用とデバッグに関するタスクについては、第3章「IPQoS 構成ファイルの作成のタスク」を参照してください。

IPQoS 構成ファイルの構文については、例6-3「IPQoS 構成ファイルの構文」を参照してください。

例 6-3 IPQoS 構成ファイルの構文

```
file_format_version ::= fmt_version version

action_clause ::= action {
     name action-name
     module module-name
     params-clause |  ""
     cf-clauses
}
action_name ::= string
module_name ::= ipgpc | dlcosmk | dscpmk | tswtclmt | tokenmt | flowacct

params_clause ::= params {
     parameters
     params-stats |   ""
     }
parameters ::=    prm-name-value parameters |  ""
prm_name_value ::= param-name param-value

params_stats ::= global-stats Boolean

cf_clauses ::= class-clause cf-clauses |
               filter-clause cf-clauses | ""

class_clause ::= class {
```

```
      name class-name
      next_action next-action-name
      class-stats | ""
                  }
class_name  ::= string
next_action_name  ::= string
class_stats ::= enable_stats Boolean
boolean ::= TRUE | FALSE

filter_clause ::= filter {
                  name filter-name
                  class class-name
                  parameters
                  }
filter_name ::= string
```

action 文

action 文を使用して、87 ページの「IPQoS アーキテクチャーと Diffserv モデル」で説明されているさまざまな IPQoS モジュールを呼び出します。

IPQoS 構成ファイルを新規作成する場合、必ずバージョン番号から始める必要があります。ついで、次の action 文を追加して、クラシファイアを呼び出す必要があります。

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
}
```

クラシファイア action 文の次に、params 句または class 句を記述します。

ほかのすべての action 文には次の構文を使用します。

```
action {
name action-name
module module-name
params-clause | ""
cf-clauses
}
```

name *action-name*	アクションに名前を付ける
module *module-name*	呼び出し予定の IPQoS モジュールを識別します。表6-5「IPQoS モジュール」に記載のモジュールの 1 つでなければなりません。

params-clause クラシファイアが処理するパラメータ (グローバル統計、次に処理するアクションなど) を指定する

cf-clauses `class` 句または `filter` 句のゼロ以上のセット

モジュール定義

モジュールの定義によって、`action` 文のパラメータを処理するモジュールが示されます。IPQoS 構成ファイルには、次の表のモジュールを含めることができます。

表 6-5 IPQoS モジュール

モジュール名	定義
`ipgpc`	IP クラシファイア
`dscpmk`	IP パケット内で DSCP 作成に使用するマーカー
`dlcosmk`	VLAN デバイスで使用するマーカー
`tokenmt`	トークンバケットメーター
`tswtclmt`	タイムスライディングウィンドウメーター
`flowacct`	フローアカウンティングモジュール

`class` 句

トラフィックのクラスごとに `class` 句を定義します。

IPQoS 構成内の残りのクラスを定義する構文は、次のとおりです。

```
class {

      name class-name
      next_action next-action-name
}
```

特定のクラスに関する統計情報収集を有効にするには、最初に `ipgpc.classify` アクション文でグローバル統計を有効にする必要があります。詳細は、102 ページの「`action` 文」を参照してください。

クラスに関する統計の収集を有効にするには、`enable_stats TRUE` 文を使用します。クラスの統計を収集する必要がない場合は、`enable_stats FALSE` を指定します。あるいは、`enable_stats` 文を削除してもかまいません。

IPQoS 対応ネットワーク上のトラフィックは、特に定義しなければ「*デフォルトクラス*」になります。

filter 句

*フィルタ*は、トラフィックフローをクラスに分類するセレクタで構成されます。これらのセレクタは、クラス句で作成されたクラスのトラフィックへ適用する条件を、明確に定義します。パケットがもっとも高い優先順位のフィルタのセレクタすべてに一致する場合、パケットはそのフィルタのクラスのメンバーと見なされます。ipgpc クラシファイアと使用できるセレクタの完全なリストについては、表6-1「IPQoS クラシファイアで利用可能なフィルタセレクタ」 を参照してください。

次の構文を持つ「*filter 句*」を使用して IPQoS 構成ファイル内にフィルタを定義します。

```
filter {
        name filter-name
        class class-name
        parameters (selectors)
        }
```

params 句

params 句には、アクション文で定義されたモジュールの処理方法が含まれます。params 句の構文を次に示します。

```
params {
            parameters
            params-stats | ""
        }
```

params 句では、モジュールに適用するパラメータを使用します。

params 句の params-stats 値は、global_stats TRUE または global_stats FALSE になります。global_stats TRUE 命令は、グローバル統計を呼び出した action 文に関する UNIX スタイルの統計を有効にします。kstat コマンドを使用して、統計情報を表示できます。クラス単位の統計を有効にする前に、action 文の統計を有効にする必要があります。

索引

数字・記号

/etc/inet/ipqosinit.conf ファイルコマンド
  概要, 44

, 44

あ

アプリケーションサーバー
  IPQoS 用の構成, 61

か

カラーアウェアネス, 18, 92
完全優先転送 (EF), 22, 94
  定義
    IPQoS 構成ファイル, 55
クラシファイアモジュール, 16
  action 文, 48
  クラシファイアの機能, 88
クラス
  class 句の構文, 103
  セレクタ、リスト, 89
  定義
    IPQoS 構成ファイル, 58, 63

さ

サービスクラス *参照* クラス
サービス品質 (QoS)
  QoS ポリシー, 12
  タスク, 10
サービスレベル契約 (SLA), 11
  顧客への課金、フローアカウンティングに基づく, 81
  さまざまなサービスクラスの提供, 15
差別化サービス, 10
  差別化サービスモデル, 16
  さまざまなサービスクラスの提供, 15
  ネットワークトポロジ, 26
サンプル IPQoS 構成ファイル
  VLAN デバイスの構成, 97
  アプリケーションサーバー, 61
  プレミアム Web サーバー, 45
  ベストエフォート型 Web サーバー, 47
セレクタ, 17
  IPQoS 5 タプル, 16
  計画, QoS ポリシーでの, 33
  セレクタ、リスト, 89
相対的優先転送 (AF), 22, 94
  マーカー action 文, 54
  AF コードポイント表, 94

た

帯域幅の調整, 13
  計画、QoS ポリシーでの, 14
タスクマップ
  IPQoS
    QoS ポリシー計画, 30
    構成計画, 25
    構成ファイルの作成, 43
統計情報, IPQoS の
  グローバル統計の有効化, 103
トラフィック管理
  帯域幅の調整, 13
  転送トラフィック, 23
  トラフィック転送, 21, 22, 22
  トラフィックフローの優先順位付け, 13
  ネットワークトポロジの計画, 27
  フローの制御, 17
トラフィック適合

計画
QoS ポリシーでの速度, 35
QoS ポリシーの結果, 36
結果 (outcome), 18, 90
速度パラメータ, 90, 91
定義, 69
トラフィック転送
Diffserv ネットワークを介したトラフィックフロー, 22
IP パケットの転送、DSCP を使用した, 21
データグラムの転送, 96
パケット転送での PHB の影響, 93
トラフィックの転送
計画、QoS ポリシー, 14

な

ネットワークトポロジ, IPQoS の
IPQoS 対応サーバーファームを備えた LAN, 27

は

フィルタ, 17
`filter` 句の構文, 104
計画、QoS ポリシーでの, 32
作成
IPQoS 構成ファイル, 59, 64
セレクタ、リスト, 89
フィルタ 句
IPQoS 構成ファイル, 104
負荷分散
IPQoS 対応ネットワーク, 28
フローアカウンティング, 81, 98
フローレコードの表, 99
フロー制御
メータリングモジュールによる, 18
ホップ単位動作 (PHB), 21
AF 転送, 22
使用、`dscpmk` マーカーでの, 93
EF 転送, 22
定義
IPQoS 構成ファイル, 70

ま

マーカーモジュール, 18, 18, 18, 93, 93
参照 `dlcosmk` マーカー
参照 `dscpmk` マーカー
DS コードポイントの指定, 95
PHB、IP パケット転送での, 21
VLAN デバイスのサポート, 96
メータリングモジュール, 18, 18, 90, 90
参照 `tokenmt` メーター
参照 `tswtclmt` メーター
概要, 17
メータリングの結果, 18, 90
呼び出し
IPQoS 構成ファイル, 69

や

ユーザー優先順位の値, 18

A

`acctadm` コマンド, フローアカウンティングでの使用, 83
`acctadm` コマンド、フローアカウンティング用, 19, 100
`action` 文, 102

C

`class` 句
IPQoS 構成ファイル, 49, 103
CoS (サービスクラス) マーク, 18

D

Diffserv 対応ルーター
DS コードポイントの評価, 95
計画, 31
Diffserv モデル
IPQoS 実装, 16, 17, 19
IPQoS での実装, 19
クラシファイアモジュール, 16
フローの例, 19
マーカーモジュール, 18
メーターモジュール, 17
`dlcosmk` マーカー, 18

VLAN タグ, 96
データグラム転送の計画, 37
ユーザー優先値、表, 96
DS コードポイント (DSCP), 18, 21
AF 転送のコードポイント, 22, 94
dscp_map パラメータ, 95
EF 転送のコードポイント, 22, 94
PHB および DSCP, 21
カラーアウェアネス構成, 92
計画、QoS ポリシーでの, 38
構成, diffserv ルーターでの, 94
構成、Diffserv ルーターでの, 72
定義
IPQoS 構成ファイル, 53
dscpmk マーカー, 18
パケット転送での PHB, 93
呼び出し
マーカー action 文, 53, 60, 66, 69

F
filter 句
IPQoS 構成ファイル, 51
flowacct モジュール, 19, 98
acctadm コマンド、フローアカウンティングファイルの作成用, 100
flowacct の action 文, 56
パラメータ, 98
フローレコード, 81
フローレコードの属性, 99
フローレコードの表, 99

I
ipgpc クラシファイア *参照* クラシファイアモジュール
IPQoS, 9
Diffserv モデルの実装, 16
IPQoS ネットワークのルーター, 71
QoS ポリシーの計画, 29
VLAN デバイスのサポート, 96
エラーメッセージ, 76
関連する RFC, 10
機能, 10
構成計画, 25
構成ファイル, 45, 101
マーカー action 文, 53
冒頭の action 文, 102
action 文の構文, 102
class 句, 49
filter 句, 51
IPQoS モジュールのリスト, 103
冒頭の action 文, 48
構成例, 39, 40
サポートするネットワークトポロジ, 26, 27, 28, 28
統計情報の生成, 84
トラフィック管理機能, 13, 15
ネットワーク例, 45
メッセージのロギング, 75
IPQoS 構成ファイルの例
カラーアウェアネスセグメント, 92
IPQoS 対応ネットワークのハードウェア, 26
IPQoS ネットワーク上の仮想 LAN (VLAN) デバイス, 96
IPQoS のエラーメッセージ, 76
IPQoS の統計
グローバル統計の有効化, 49
IPQoS の統計情報
生成、kstat コマンドによる, 84
クラスベースの統計の有効化, 103
IPQoS のネットワークトポロジ, 26
IPQoS 対応のファイアウォールを備えた LAN, 28
IPQoS 対応ホストを備えた LAN, 27
構成例, 40
IPQoS のネットワーク例, 45
ipqosconf コマンド
構成の適用, 74

K
kstat コマンド、IPQoS での使用, 84

P
params 句
マーカー action, 53
メータリング action, 69
flowacct action での使用, 57
グローバル統計の定義, 49, 104
構文, 104

Q

QoS ポリシー, 12
　計画タスクマップ, 30
　実装
　　IPQoS 構成ファイル, 43
　フィルタの作成, 32
　ポリシー組織のテンプレート, 29

R

RFC (Request for Comments)
　IPQoS, 10

S

`svc:/network/ipqos` サービス
　概要, 44
IPQoS の `syslog.conf` ファイルのロギング, 75

T

`tokenmt` メーター, 18
　カラーアウェアとして構成, 18
　カラーアウェアネス構成, 92
　シングルレートメーターとして構成, 91
　速度の計測, 90
　速度パラメータ, 91
　ツーレートメーター, 91
`tswtclmt` メーター, 18, 93
　速度の計測, 93

W

Web サーバー
　IPQoS の構成, 47, 59
　IPQoS 用の構成, 45, 57