



# IPQoS の管理

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Part No: 816-6182-10  
2002 年 9 月

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

Federal Acquisitions: Commercial Software-Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG 明朝 L、HG-MincyoL-Sun、HG ゴシック B、および HG-GothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。HG 平成明朝体 W3@X12 は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *IPQoS Administration Guide*

Part No: 816-4094-10

Revision A



020807@2851



# 目次

---

はじめに	13
<b>1 IPQoS の紹介 (概要)</b>	<b>17</b>
IPQoS の基本	17
差別化サービスとは	17
IPQoS の機能	18
サービス品質の詳細情報	18
IPQoS によるサービス品質の提供	20
サービスレベル契約の実装	20
一般の組織にとってのサービス品質の保証	20
サービス品質ポリシーの紹介	20
IPQoS によるネットワーク効率の向上	21
帯域幅がネットワークトラフィックに与える影響	21
サービスクラスを使ったトラフィックの優先順位付け	22
差別化サービスモデル	22
クラシファイア (ipgpc) の概要	23
メーター (tokenmt および tswtclmt) の概要	24
マーカー (dscpmk および dlcosmk) の概要	25
フローカウンティング (flowacct) の概要	25
例 — トラフィックが IPQoS モジュールをどのように通過するか	26
IPQoS 対応ネットワークでのトラフィック転送	27
DS コードポイント (DSCP)	27
ホップ単位動作	28

<b>2</b>	<b>IPQoS 対応ネットワークの計画 (手順)</b>	<b>31</b>
	一般的な IPQoS の構成計画 (作業マップ)	31
	diffserv ネットワークトポロジの計画	32
	diffserv ネットワークのハードウェア計画	32
	IPQoS ネットワークトポロジ	33
	サービス品質ポリシーの計画	36
	QoS ポリシー計画の手掛かり	36
	▼ IPQoS 用のネットワークを準備する方法	38
	▼ QoS ポリシーのクラスを定義する方法	39
	▼ QoS ポリシーにフィルタを定義する方法	41
	▼ フロー制御を計画する方法	43
	▼ 転送動作を計画する方法	46
	▼ フローアカウンティングを計画する方法	48
	IPQoS の構成例の紹介	49
	例 — IPQoS トポロジ	49
<b>3</b>	<b>IPQoS 構成ファイルの作成 (手順)</b>	<b>53</b>
	IPQoS 構成ファイル内での QoS ポリシーの定義 (作業マップ)	53
	QoS ポリシー作成用のツール	54
	IPQoS 構成ファイル	55
	Web サーバー用 IPQoS 構成ファイルの作成	55
	▼ IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法	58
	▼ IPQoS 構成ファイル内でフィルタを定義する方法	60
	▼ IPQoS 構成ファイル内でトラフィック転送を定義する方法	62
	▼ 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	65
	▼ 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	67
	アプリケーションサーバー用 IPQoS 構成ファイルの作成	70
	▼ アプリケーションサーバー用 IPQoS 構成ファイルの作成方法	72
	▼ 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	75
	▼ IPQoS 構成ファイル内でフロー制御を構成する方法	77
	ルーター上での差別化サービスの提供	81
	▼ IPQoS 対応ネットワーク上でルーターを構成する方法	81
<b>4</b>	<b>IPQoS の起動と保守 (手順)</b>	<b>83</b>
	IPQoS の管理 (作業マップ)	83

IPQoS 構成の有効化	84
▼ 新規構成の IPQoS カーネルモジュールへの適用方法	84
▼ リブート後にも IPQoS 構成を適用する方法	85
IPQoS メッセージの syslog によるログ記録の有効化	85
▼ ブート時に IPQoS メッセージを記録する方法	86
IPQoS エラーメッセージの使用	86
<b>5</b>	<b>フローカウンティングの使用と統計情報の収集 (手順) 91</b>
	フローカウンティングの設定 (作業マップ) 91
	フローに関する情報の記録 92
	▼ フローカウンティングデータ用のファイルの作成方法 92
	▼ フローカウンティングファイルを表示する方法 94
	統計情報の収集 95
	例 — IPQoS の kstat 統計情報 95
<b>6</b>	<b>IPQoS の詳細 (リファレンス) 97</b>
	IPQoS アーキテクチャと diffserv モデル 97
	クラシファイアモジュール 98
	メーターモジュール 100
	マーカーモジュール 103
	flowacct モジュール 107
	IPQoS 構成ファイル 110
	アクション文 111
	モジュール定義 112
	クラス句 113
	フィルタ句 113
	Params 句 114
	ipqosconf 構成ユーティリティ 114
	用語集 115
	索引 117



# 表目次

---

表 2-1	IPQoS の構成計画 (作業マップ)	32
表 2-2	QoS 構成テンプレート	37
表 2-3	QoS ポリシーの計画 (作業マップ)	37
表 2-4	一般的な IPQoS セレクタ	42
表 2-5	メーターが定義されている QoS ポリシーの例	45
表 3-1	IPQoS 構成ファイルの作成 (作業マップ)	53
表 3-2	ネットワーク例用に構成されたホップ単位動作	82
表 4-1	IPQoS の構成と保守 (作業マップ)	83
表 4-2	IPQoS のエラーメッセージ	87
表 5-1	フローカウンティングの設定 (作業マップ)	91
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	99
表 6-2	相対的優先転送のコードポイント	104
表 6-3	801.D ユーザー優先順位値	106
表 6-4	flowacct レコードの属性	109
表 6-5	IPQoS モジュール	112





# 図目次

---

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	26
図 1-2	diffserv 対応ネットワークのホップ間のパケット転送	29
図 2-1	ネットワークセグメント上の IPQoS システム	33
図 2-2	IPQoS 対応サーバーファームのネットワーク	34
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	35
図 2-4	IPQoS のトポロジの例	49



# 例目次

---

例 3-1	プレミアム Web サーバー用 IPQoS 構成ファイルの例	55
例 3-2	ベストエフォート Web サーバー用の構成例	57
例 3-3	アプリケーションサーバー用の構成例	70
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	102
例 6-2	VLAN デバイスを持つシステムの IPQoS 構成ファイル	106
例 6-3	IPQoS 構成ファイルの構文	111



# はじめに

---

『IPQoS の管理』では、Solaris™ オペレーティング環境の IPQoS 機能を使用して、差別化サービスをネットワーク上で提供する方法について説明します。IPQoS を使用すると、さまざまなレベルのサービスをネットワークユーザーに提供できます。また、ネットワークトラフィックを管理することもできます。

---

## 対象読者

このマニュアルは、読者として、TCP/IP に関する詳細な知識を持つ熟練したシステム管理者を想定しています。主に、ネットワークのルーター管理を担当するか、その業務に精通している方を対象としています。また、サイトのネットワークトポロジ、ネットワーク使用に関する企業のポリシー、そしてできればネットワークセキュリティにも精通している必要があります。

---

## 内容の紹介

『IPQoS の管理』には、次の章が含まれます。

第 1 章では、IPQoS 機能および IPQoS のベースとなる diffserv アーキテクチャに関する基本的な情報を提供します。

第 2 章では、IPQoS 対応ネットワークトポロジの計画方法について説明します。この章では、予想される IPQoS システムを構築するためにサービス品質ポリシーを作成する際の計画手順についても説明します。

第 3 章では、サービス品質ポリシーに基づく IPQoS 構成ファイルの構築方法について説明します。

第4章では、IPQoSの保守方法および追跡方法について説明します。

第5章では、IPQoSフローカウンティングの構成方法、およびkstatコマンドを使用したIPQoS統計の表示方法について説明します。

第6章では、IPQoSの各モジュールおよびIPQoS構成ファイルに関する詳細な情報を提供します。

---

## 関連書籍

次の書籍は、差別化サービスアーキテクチャについて論じています。

- 『*Quality of Service*』、Paul Ferguson、Geoff Huston 共著、John Wiley & Sons, Inc. 発行、1998
- 『*Differentiated Services for the Internet*』、Kalevi Kilkki 著、Macmillan Technical Publishing 発行、1999

---

## Sun のオンラインマニュアル

docs.sun.com では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、<http://docs.sun.com> です。

---

## 表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。  system%
<b>AaBbCc123</b>	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% <b>su</b> password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
[ ]	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。  この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% <b>grep</b> `^#define \ XV_VERSION_STRING'

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```

■ C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

■ Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。



## 第 1 章

---

# IPQoS の紹介 (概要)

---

IP サービス品質 (IPQoS) 機能を使用すると、優先順位付け、管理、およびアカウントリング統計情報の収集を行うことができます。また、ネットワークのユーザーに一定レベルのサービスを提供したり、ネットワークの輻輳が発生しないようにトラフィックを管理したりできます。

この章では、以下の内容について説明します。

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

---

## 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) アーキテクチャに対応しています。Solaris 9/02 オペレーティング環境では、IPQoS が TCP/IP プロトコルスタックの IP レベルで実装されています。

### 差別化サービスとは

IPQoS を有効にすると、顧客やアプリケーションごとに、異なるレベルのネットワークサービスを提供できます。これらの「差別化サービス」は、企業が顧客に提供するサービスレベルの構造に基づく場合もあれば、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。

サービス品質の提供には、次の活動が含まれます。

- 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監視し、パフォーマンスの統計情報を提供する
- ネットワーク資源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。

- QoS ポリシーを設定するためのコマンド行ツール
- 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 ポリシーを設定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タリングモジュール
- パケットの IP ヘッダ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™ アーキテクチャのサポート
- IPv4 および IPv6 アドレス指定のサポート
- IPsec との相互運用性
- 仮想ローカルエリアネットワーク (VLAN) 用の 802.1 D ユーザー優先順位付けのサポート

## サービス品質の詳細情報

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。

## 関連書籍

サービス品質の理論や実践については、次の関連書籍を参照してください。

- 『*Quality of Service*』、Paul Ferguson、Geoff Huston 共著、John Wiley & Sons, Inc. 発行、1998
- 『*Differentiated Services for the Internet*』、Kalevi Kilkki 著、Macmillan Technical Publishing 発行、1999

## RFC (Requests for Comments)

IPQoS は、次の RFC およびインターネットドラフトの仕様に準拠しています。

- RFC 2474 (*Definition of the Differentiated Services Field*)。差別化サービスのサポートに必要な、IPv4 パケットヘッダーおよび IPv6 パケットヘッダーの ToS フィールドの機能拡張、すなわち DS フィールドについて解説している
- RFC 2475 (*An Architecture for Differentiated Services*)。diffserv アーキテクチャの構成とモジュールについて詳しく解説している
- RFC 2597 (*Assured Forwarding PHB Group*)。相対的優先転送 (AF) ホップ単位動作について解説している
- RFC 2598 (*An Expedited Forwarding PHB*)。完全優先転送 (EF) ホップ単位動作について解説している
- インターネットドラフト (*An Informal Management Model for Diffserv Routers*)。ルーター上に diffserv アーキテクチャを実装するためのモデルを紹介している

## Web サイト

IETF の Differentiated Services Working Group は、diffserv インターネットドラフトへのリンクを含む Web サイト (<http://www.ietf.org/html.charters/diffserv-charter.html>) を管理しています。

Cisco Systems 社や Juniper Networks 社などのルーター製造元は、それぞれ自社の Web サイトで、差別化サービスの製品への実装状況について解説しています。

## マニュアルページ

IPQoS ソフトウェアには、次のマニュアルページが含まれています。

- `ipqosconf(1M)`。IPQoS 構成ファイルを設定するためのコマンド
- `ipqos(7ipp)`。diffserv アーキテクチャモデルの IPQoS 実装について解説している
- `ipgpc(7ipp)`。diffserv クラシファイアの IPQoS 実装について解説している
- `tokenmt(7ipp)`。IPQoS の tokenmt メーターモジュールについて解説している
- `tswtclmt(7ipp)`。IPQoS の tswtclmt メーターモジュールについて解説している
- `dscpmk(7ipp)`。DSCP マーカーモジュールについて解説している
- `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解説している
- `flowacct(7ipp)`。IPQoS フローアカウンティングモジュールについて解説している
- `acctadm(1M)`。Solaris の拡張アカウンティング機能を構成し、現在では IPQoS 拡張機能も追加されているコマンド

---

## 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することができます。

### 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。これに対し、中間価格の SLA では、顧客は電子メールに対して営業時間中のみ高い優先順位が与えられ、他のすべてのトラフィックに対しては毎日 24 時間中程度の優先順位が与えられます。

### 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することができます。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することができます。

### 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、Solaris 9/02 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」とみなされません。

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
- サービスクラスに関して必要に応じて収集する統計情報。たとえば、顧客または特定のアプリケーションによって生成されるトラフィックなど

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 ポリシーの作成作業については、36 ページの「サービス品質ポリシーの計画」で説明します。

---

## 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いくつかの機能があります。コンピュータネットワークが拡大すると、ユーザーや高機能なプロセッサの数が増加して、生成されるネットワークトラフィックを管理する必要性も増大します。ネットワークを過剰に使用すると、データが失われたり、トラフィックの輻輳が生じて応答時間が遅くなったりします。

従来、システム管理者は帯域幅を拡張する方法でネットワークトラフィックの問題に対処してきました。しかし同時に、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、またどこに必要なかを評価できます。

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があります。それに加えて、ISP や ASP の場合は、ネットワークパフォーマンスを管理して、顧客が支払いに応じたネットワークサービスを受けられるようにする必要があります。

## 帯域幅がネットワークトラフィックに与える影響

IPQoS を使用すると、ネットワークの「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調整できます。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することができます。

ネットワーク上のトラフィックを効果的に管理するには、帯域幅の使用量に関して、次の点を明らかにしておく必要があります。

- 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
- 優先順位を落としてもよい、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

サービス品質を実装するには、まずネットワークトラフィックを解析して、トラフィックを分類するためのおおまかなグループを決めます。次に、各種グループを、それぞれ特性や優先順位を持つサービスクラス別にまとめます。これらのクラスが基本的なカテゴリとなり、それに基づいて組織の QoS ポリシーを作成します。サービスクラスは、管理の対象となるトラフィックグループを代表します。

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。このように、ある顧客の Web サイト宛ての着信トラフィックを 1 つのトラフィッククラスとしてまとめることができます。

企業の場合は、部署の要件に基づいて、またはネットワークトラフィックにおける特定のアプリケーションの優越に基づいてサービスクラスを作成できます。次に、企業向けのトラフィッククラスの例をいくつか示します。

- 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る。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 ポリシーでは少量の帯域幅と低い優先順位を割り当てる
- 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る
- 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る
- 企業の外部 Web サイトへの来訪。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 には次のモジュールがあります。これらのモジュールは、RFC 2475 に定義されている「差別化サービス (diffserv)」アーキテクチャの一部です。

- 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

この節では、IPQoS で使用する diffserv モジュールについて簡単に説明します。QoS ポリシーを設定するには、これらのモジュール、その名前、およびその使用目的を認識しておく必要があります。各モジュールの詳細については、97 ページの「IPQoS アーキテクチャと 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、クラシファイアは、ネットワークトラフィックの「フロー」からいくつかの packets、つまり次の IP ヘッダーフィールドに同じ情報が入っている packets のグループを選択します。

- 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 のクラシファイアモジュールの名前は ipgpc です。ipgpc は、トラフィックフローを、IPQoS 構成ファイルに設定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、98 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。

## クラス

「クラス」とは、似たような特性を共有するネットワークフローのグループのことです。たとえば、ISP は、顧客に提供するさまざまなサービスレベルを表すクラスを定義することができます。一方、ASP は、アプリケーションごとに異なるサービスレベルを提供する SLA を定義することができます。ASP の QoS ポリシーでは、特定の着信先 IP アドレス宛ての発信 FTP トラフィックを 1 つのクラスにまとめることができます。ある企業の外部 Web サイトからの発信トラフィックも、1 つのクラスとして定義できます。

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。

クラスの定義方法については、39 ページの「QoS ポリシーのクラスを定義する方法」を参照してください。

## フィルタ

「フィルタ」とは、「セレクタ」と呼ばれるパラメータを含む、規則セットのことです。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
- 発信元および着信先のポート番号
- 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、単純なフィルタで 80 という値の着信先ポートを指定するとします。ipgpc クラシファイアは、着信先ポート 80 (HTTP) 宛てのすべてのパケットを選択し、QoS ポリシーの指示に従ってそれらのパケットを処理します。

フィルタの作成については、41 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設定された速度にどれだけ適合しているかを評価します。そして、トラフィックフローの結果に基づいて、次のアクションを選択します。次のアクションでは、パケットを別のアクションに送信したり、それ以上処理しないでネットワークに戻したりできます。

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つのメーター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ーリング方式を使用
- tswtclmt。タイムスライディングウィンドウメーターリング方式を使用

どちらのメーター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、および設定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 を使ってパケットを緑、黄、または赤の結果にマップします。

メーター用のパラメータの定義については、43 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あります。

- `dscpmk`。IP パケットヘッダーの DS フィールドに「DS コードポイント (DSCP)」と呼ばれる数値を付ける。`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る
- `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る。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)」のことである

`dlcosmk` は、IPQoS の追加機能として設計されたものであり、`diffserv` モデルの一部ではない

QoS ポリシーのマーカー計画の実装については、46 ページの「転送動作を計画する方法」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` モデルに追加されています。`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することができます。フローアカウンティングは、容量計画やシステムの監視にも役立ちます。

`flowacct` 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できます。基本的なログには、次に示すように、IPQoS 5 タプルの他に 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

上記以外の属性に関する統計情報を収集することもできます。  
これについては、92 ページの「フローに関する情報の記録」、および flowacct (7ipp) と acctadm(1M) のマニュアルページを参照してください。

フローアカウント計画の実装については、48 ページの「フローアカウント計画する方法」を参照してください。

## 例 — 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。

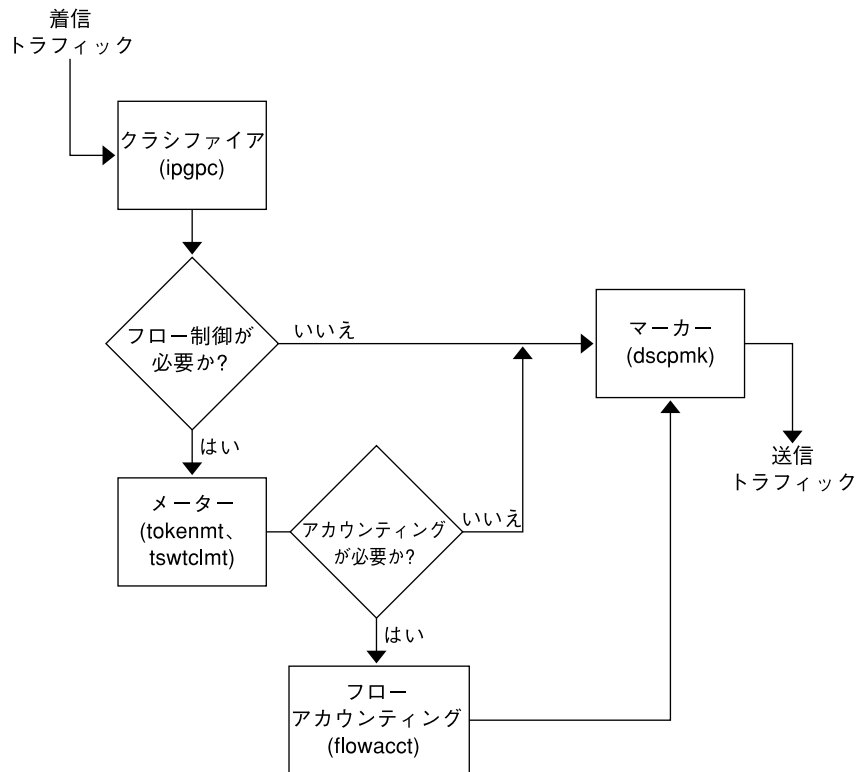


図 1-1 diffserv モデルの IPQoS 実装を通過するトラフィックフロー

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. メーターが、設定された速度を強制するため、トラフィック適合度の値をフロー制御され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

この節では、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 (DSCP)

DS コードポイントは、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 アーキテクチャでは、IPQoS 対応のシステムおよび使用するルーターに対して、1 組の DS コードポイントと、それに対応するアクションすなわち「転送動作」を定義します。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。ルーターは、DSCP 値を持つパケットを受信すると、パケットをネットワーク上に送出するときに、その DSCP に対応付けられた転送動作をパケットに適用します。

---

注 - d1cosmk マーカーは、DSCP を使用しません。代わりに、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上で IPQoS を構成する場合は、103 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他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、同じ PHB をそのパケットに適用します。ただし、別の diffserv システムによって DSCP が変更された場合は例外です。PHB の詳細については、103 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資源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 ポリシーでこの目的を達成するには、トラフィッククラスが IPQoS 対応システムから送出される時に、その優先度レベルを示す DS コードポイントを定義します。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 ポリシーによってあるトラフィッククラスにドロップ率が低い PHB を保証する DSCP 割り当てることができます。このトラフィッククラスは、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ます。QoS ポリシーに別の DSCP を追加して、他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 (Expedited Forwarding、EF)

「完全優先転送 (Expedited Forwarding、EF)」PHB は、EF 関連の DSCP を持つトラフィッククラスが一番高い優先順位を割り当てられ、キューに格納されないことを保証します。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 DS コードポイントは、101110 にすることをお勧めします。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 (Assured Forwarding、AF)

「相対的優先転送 (Assured Forwarding、AF)」PHB 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。各転送クラスには、3 種類のドロップ優先度があります (表 6-2 を参照)。

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。QoS ポリシーでは、QoS ポリシーを計画するときに、ネットワーク上のトラフィックやサービスに優先順位を付けておきます。そうすれば、優先順位の付いたトラフィックにそれぞれ異なる AF レベルを割り当てることができます。

## 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 に対応していません。

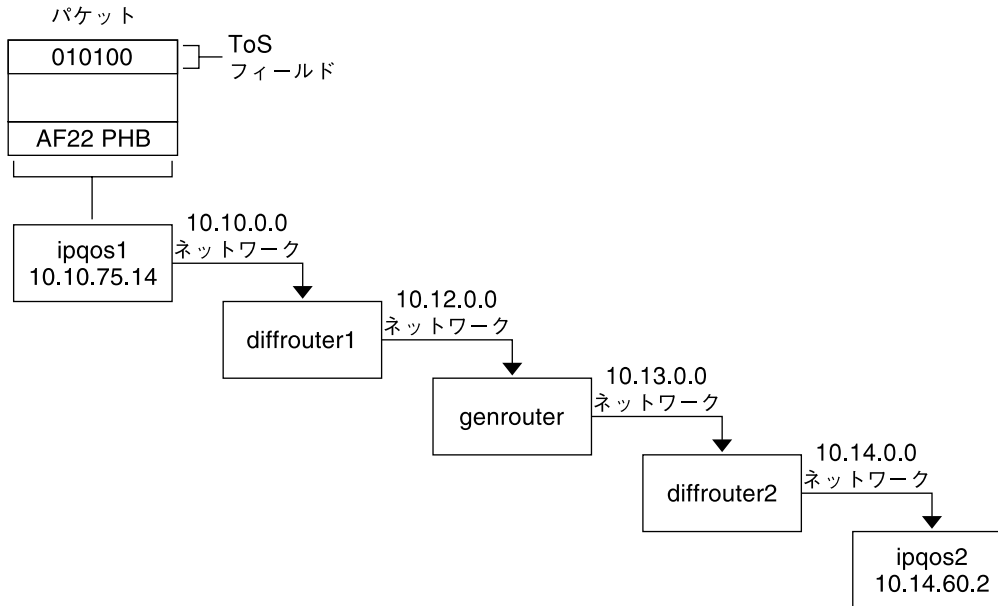


図 1-2 diffserv 対応ネットワークのホップ間のパケット転送

次の手順は、前の図に示すパケットの流れをたどっています。この手順では、パケットがホスト ipqos1 から送出され、いくつかのホップを通過して ipqos2 に進みます。

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 が、発生したパケットフローに QoS ポリシーを適用し、ftp トラフィックの分類に成功します。  
システム管理者は、ローカルネットワーク 10.10.0.0 から送出されるすべての発信 ftp トラフィックに対して、あらかじめ 1 つのクラスを作成しています。この ftp クラスのトラフィックには、AF22 PHB (クラス 2、中程度のドロップ優先度) が割り当てられており、2 Mbps のトラフィックフロー速度が設定されています。
3. ipqos1 が ftp フローを計測して、2 Mbps の認定速度を超えていないかどうか調べます。
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 が ftp パケットを受け取り、DSCP をチェックします。diffrouter1 に輻輳があった場合、AF22 クラスのパケットはドロップされます。
6. diffrouter1 のファイルで AF22 に対して設定されている 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 に進みます。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 が ftp トラフィックを受け取り、ipqos1 上のユーザーにユーザー名とパスワードの入力を求めるプロンプトを表示します。

## 第 2 章

---

# IPQoS 対応ネットワークの計画 (手順)

---

IPQoS は、Solaris 9 9/02 オペレーティング環境を実行する任意のシステム上で構成できます。そして、IPQoS システムを diffserv 対応ルーターと組み合わせると、イントラネット上で差別化サービスを提供したり、トラフィック管理を行うことができます。

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作業について説明します。

この章では、以下の内容について説明します。

- 31 ページの「一般的な IPQoS の構成計画 (作業マップ)」
- 32 ページの「diffserv ネットワークトポロジの計画」
- 36 ページの「サービス品質ポリシーの計画」
- 49 ページの「IPQoS の構成例の紹介」

---

## 一般的な IPQoS の構成計画 (作業マップ)

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や機能だけでなく、IPQoS 対応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も考慮する必要があります。次の表に、IPQoS をネットワーク上に実装するための主な計画作業を示します。

表 2-1 IPQoS の構成計画 (作業マップ)

作業	説明	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を考慮して、自分のサイトにもっとも適したソリューションを考える	32 ページの「diffserv ネットワークトポロジの計画」
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	36 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、およびアカウントリング機能を決める	36 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、ルーターのマニュアルを参照

## 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 対応システムが最低 1 つと diffserv 対応ルーターが 1 台必要です。この節で説明するように、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 (Sun Enterprise™ 10000 サーバーなど) 上で実行します。一方、ネットワークのニーズに応じて、UltraSPARC システムなどの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。

- Solaris システム。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
- 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
- IPQoS システム。仮想 LAN の一部である

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。もし、現在使用しているルーターが diffserv に対応していない場合は、Cisco Systems 社や Juniper Networks 社などのルーター製造元が提供する diffserv のソリューションを検討してください。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

以下では、ネットワーク上のさまざまなニーズを満たす IPQoS 計画を、いくつかの図で説明します。

### 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。

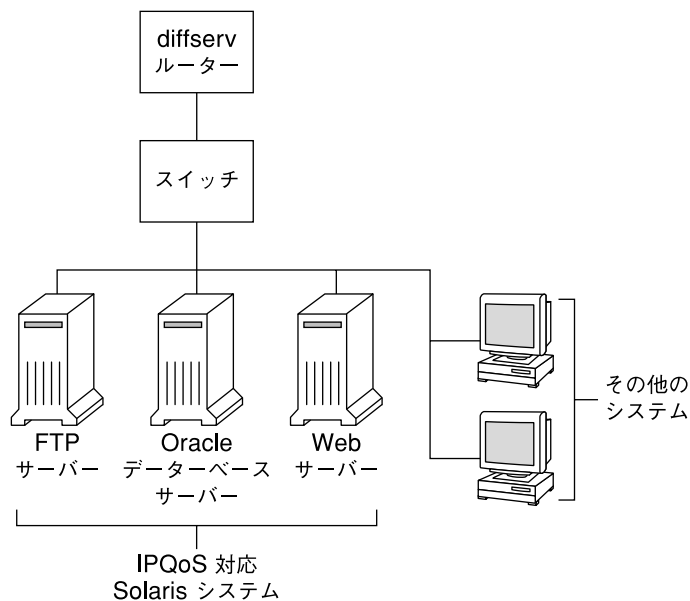


図 2-1 ネットワークセグメント上の IPQoS システム

前の図に示すネットワークは、企業のイントラネットの 1 つのセグメントにすぎません。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ネットワークストリームに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。

このマニュアルで取り上げる例では、個々のホストでの IPQoS をシナリオとして使用します。このマニュアルを通じて使用するトポロジ例については、図 2-4 を参照してください。

## サーバーファームのネットワークでの IPQoS

次の図は、いくつかの異機種サーバーファームを備えたネットワークを示しています。

このようなトポロジでは、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い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、アプリケーションデータに含まれているユーザー ID やプロジェクト ID などのセレクトクを使って、ルーターの向こう側へ追加のフィルタ処理を実行することもできます。

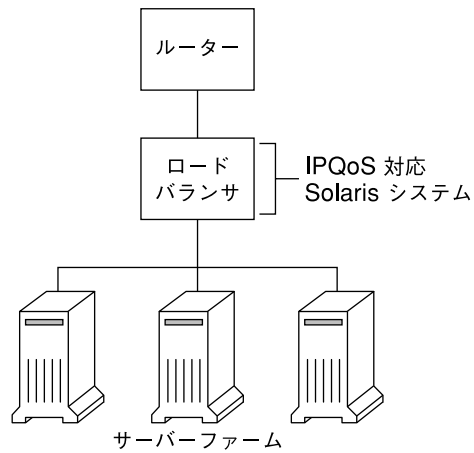


図 2-2 IPQoS 対応サーバーファームのネットワーク

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、このトポロジでは、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他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他のセグメントから保護されている、企業ネットワークのセグメントの 1 つを示しています。

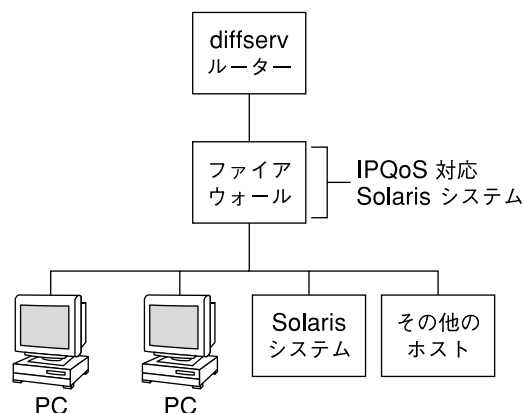


図 2-3 IPQoS 対応のファイアウォールによって保護されているネットワーク

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てはなりません。

ファイアウォールのセキュリティポリシーによって、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

QoS ポリシーを計画するための情報を IPQoS 構成ファイルに必要な情報を含む形式で収集します。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情報を表にすることができます。

表 2-2 QoS 構成テンプレート

クラス	優先順位	フィルタ	セレクタ	速度	転送動作をどうするか	アカウント ングが必要か
クラス 1	1	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 ウンティ ング統計情報 を必要とす る
		フィルタ 2	セレクタ 1 セレクタ 2			
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 ウンティ ング統計情報 を必要とす る
		フィルタ 2	セレクタ 1 セレクタ 2			

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することができます。以下では、テンプレートに示されているカテゴリの情報を入手する方法について説明します。

次の作業マップは、QoS ポリシーを計画するための主な作業を示しています。

表 2-3 QoS ポリシーの計画 (作業マップ)

作業	説明	参照先
1. IPQoS をサポートするようネットワークポロジを設計する	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	38 ページの「IPQoS 用のネットワークを準備する方法」
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	39 ページの「QoS ポリシーのクラスを定義する方法」

表 2-3 QoS ポリシーの計画 (作業マップ) (続き)

作業	説明	参照先
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	41 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	43 ページの「フロー制御を計画する方法」
5. QoS ポリシーで使用する DS コードポイント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方法を計画する	46 ページの「転送動作を計画する方法」
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監視計画を設定する	トラフィッククラスを評価して、アカウントリングまたは統計上の目的で監視するトラフィックフローを決める	48 ページの「フローアカウントリングを計画する方法」

注 - この節の残りの部分では、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 ルーターの QoS ポリシーを計画する方法については、ルーターのマニュアル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 用のネットワークを準備する方法

次の手順は、QoS ポリシーを作成する前に行う一般的な計画作業を示しています。

1. ネットワークトポロジを確認し、IPQoS システムと **diffserv** ルーターを使用するトポロジを計画します。  
トポロジの例については、32 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 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作業をすべて確認し、実行します。  
詳細については、ルーターのマニュアルとルーター製造元の Web サイトを参照してください。

## ▼ QoS ポリシーのクラスを定義する方法

QoS ポリシーを定義する最初の手順は、トラフィックフローをいくつかのクラスに分類することです。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。

---

注 - クラスの概要については、23 ページの「クラス」を参照してください。

---

次の手順の前に、38 ページの「IPQoS 用のネットワークを準備する方法」で特定したとおり、ネットワーク上のどのシステムを IPQoS 対応とするかが決まっているものとします。

1. **QoS** ポリシーを構成するための表を作成します。  
推奨される表については、表 2-2 を参照してください。
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。  
次の質問は、ネットワークトラフィックを解析してクラス定義を行うためのガイドラインとなります。
  - 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。また、プレミアム Web サイトをサービスレベルの 1 つとして提供する SLA もあれば、ベストエフォートの個人用 Web サイトを割引き客に提供する SLA もあります。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 などがあります。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

もっとも高い優先順位の転送動作を必要とする重要なアプリケーションは、ルーターで特別な処理を受ける必要があります。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監視ユーティリティを使用して、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 ポリシーの設定に従って、問題のあるフローをキューに入れたりスケジューリング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トिंग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、アカウントिंग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、クラスを新たに定義して、監視中のトラフィックフローやセキュリティ上の目的でアクセスを制限しているトラフィックフローの統計情報を取得することもできます。

#### 4. QoS 構成表に定義したクラスを記入します。

#### 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。優先レベルは、構成上の目的で割り当てられているだけで、IPQoS が実際に使用するわけではありません。また、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。



クラスの優先順位付けの重要性については、次の項を参照してください。

6. クラスの定義が終わったら、次はクラスごとにフィルタを定義します (41 ページの「QoS ポリシーにフィルタを定義する方法」を参照)。

## クラスの優先順位付け

クラスを作成するうちに、どのクラスの優先順位が高いか、中程度か、ベストエフォートでよいかがすぐに明らかになってきます。クラスの優先順位付けは、PHB (ホップ単位動作) を発信トラフィックに割り当てるときに特に重要となります (46 ページの「転送動作を計画する方法」を参照)。

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時に帯域幅をめぐる競争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、他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。

## ▼ QoS ポリシーにフィルタを定義する方法

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、セレクタに定義されている基準を使ってトラフィックフローからパケットを取り出し、それらのパケットをクラスに対応付けます (フィルタの概要については、24 ページの「フィルタ」を参照)。

次の手順に進む前に、39 ページの「QoS ポリシーのクラスを定義する方法」を終わらせておく必要があります。

1. 39 ページの「QoS ポリシーのクラスを定義する方法」で作成した QoS 構成表のクラスごとにフィルタを最低 1 つ作成します。  
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。  
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。利用可能なセレクタについては、表 6-1 を参照してください。

注 – セレクタは正しい判断のもとで選択してください。また、クラスの packets 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。

表 2-4 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	使用するアドレス指定方式。V4 または V6 を使用する。デフォルトは V4
dsfield	DS フィールドの内容、つまり DS コードポイント。このセレクタは、特定の DSCP が付いている着信 packets 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。詳細については、41 ページの「クラスの優先順位付け」を参照
user	UNIX のユーザー ID またはユーザー名。上位アプリケーションの実行時に使用される
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。LOCAL_IN、LOCAL_OUT、FWD_IN、または FWD_OUT のいずれかの値

表 2-2 で紹介したテンプレートを使用して、定義したクラスのフィルタを記入します。

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

## 次に進む手順

作業	参照先
フロー制御方式を定義する	43 ページの「フロー制御を計画する方法」
フローがネットワークストリームに戻される ときの転送動作を定義する	46 ページの「転送動作を計画する方法」
特定の種類のトラフィックに対してフローア カウンティングを計画する	48 ページの「フローアカウンティングを計画 する方法」
QoS ポリシーにさらにクラスを追加する	39 ページの「QoS ポリシーのクラスを定義す る方法」
QoS ポリシーにさらにフィルタを追加する	41 ページの「QoS ポリシーにフィルタを定義 する方法」

### ▼ フロー制御を計画する方法

フロー制御では、まずクラスのトラフィックフローを測定し、次に定義された速度でパケットをネットワークに送出します。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。メーターの概要については、24 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。

次の手順の前に、41 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、フィルタとセレクタを定義してあるものとします。

1. ネットワークの最大帯域幅を調べます。
2. ネットワーク上でサポートされている **SLA** をすべて確認して、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。  
SLA はある一定レベルのサービスを顧客に保証しているため、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. 39 ページの「QoS ポリシーのクラスを定義する方法」で作成したクラスのリストを確認します。  
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確認します。  
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ものとします。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 – すべてのクラスを計測する必要があるとは限りません。クラスのリストを確認するときは、このガイドラインに留意してください。

---

4. クラスのどのフィルタがフロー制御を必要とするトラフィックを選択するかを確認することにより、計測するクラスのリストを作成し直します。  
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、ある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定義したとします。しかし、結果的にどちらかの方向のトラフィックしかフロー制御を必要としない場合があります。
5. フロー制御を行うクラスごとにメーターモジュールを選択します。  
作成した構成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構成表に追加します。  
tokenmt モジュールを使用する場合は、次の速度を bps 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 の速度の詳細については、101 ページの「tokenmt をツール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、102 ページの「tswtclmt メータリングモジュール」および twstclmt (7ipp) のマニュアルページに定義されています。
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、黄の3つです。定義した速度に当てはまるトラフィック適合結果を QoS 構成表に追加します。これらのメーターの結果については、100 ページの「メーターモジュール」で詳しく説明します。  
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。よく取られるアクションは、パケットヘッダーに PHB を付けることですが、常にそうするわけではありません。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 packets をドロップすることもできます。

次の表に、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。この IPQoS システムが配置されているネットワークには、合計 100 Mbps (100000000 bps) の帯域幅があります。QoS ポリシーは、電子メールクラスに対して、低い優先順位とベストエフォートの転送動作を割り当てています。

表 2-5 メーターが定義されている QoS ポリシーの例

クラス	優先順位	フィルタ	セレクタ	速度
email	8	mail_in	daddr 10.50.50.5 dport imap direction LOCAL_IN	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 =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
		mail_out	saddr 10.50.50.5 sport imap direction LOCAL_OUT	

## 次に進む手順

作業	参照先
フローがネットワークストリームに戻される ときの転送動作を定義する	46 ページの「転送動作を計画する方法」
特定の種類のトラフィックに対してフローア カウンティングを計画する	48 ページの「フローアカウンティングを計画 する方法」
QoS ポリシーにさらにクラスを追加する	39 ページの「QoS ポリシーのクラスを定義す る方法」
QoS ポリシーにさらにフィルタを追加する	41 ページの「QoS ポリシーにフィルタを定義 する方法」
別のフロー制御方式を定義する	43 ページの「フロー制御を計画する方法」

作業	参照先
IPQoS 構成ファイルを作成する	58 ページの「IPQoS 構成ファイルの開始方法 およびトラフィッククラスの定義方法」

## ▼ 転送動作を計画する方法

転送動作は、ネットワークに転送されようとしているトラフィックフローの優先順位とドロップ優先度を決めます。選択できる主な転送動作は2つあります。1つは他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。

- dscpmk。IP パケットの DS フィールドに DS コードポイントを付けるために使用する
- dlcosmk。データグラムの VLAN タグにサービスクラス (CoS) 値を付けるために使用する

---

注 - この節では、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。詳細については、106 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 DS コードポイントを割り当てる必要があります。dscpmk マーカーでは、パケットの DS フィールドに DS コードポイントを付けます。クラスの DS コードポイントは、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。DS コードポイントと転送の概要については、27 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。

次の手順の前に、QoS ポリシーのクラスとフィルタを定義してあるものとします。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 DS コードポイント 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 の詳細について

は、104 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。  
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 packets 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。次の表に、高いレベルのトラフィックを生成するよく使われるゲーム用アプリケーションのクラスを定義した、QoS ポリシー例の一部を示します。

クラス	優先順位	フィルタ	セクタ	速度	転送
games_app	9	games_in	sport 6080		
		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 DS コードポイントを割り当てます。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 packets をドロップするよう指示します。すべての AF コードポイントは、表 6-2 に示してあります。

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。

## 次に進む手順

作業	参照先
特定の種類のトラフィックに対してフローアカウンティングを計画する	48 ページの「フローアカウンティングを計画する方法」
QoS ポリシーにさらにクラスを追加する	39 ページの「QoS ポリシーのクラスを定義する方法」
QoS ポリシーにさらにフィルタを追加する	41 ページの「QoS ポリシーにフィルタを定義する方法」
フロー制御方式を定義する	43 ページの「フロー制御を計画する方法」
フローがネットワークストリームに戻される ときの追加の転送動作を定義する	46 ページの「転送動作を計画する方法」
IPQoS 構成ファイルを作成する	58 ページの「IPQoS 構成ファイルの開始方法 およびトラフィッククラスの定義方法」

### ▼ フローアカウンティングを計画する方法

IPQoS の flowacct モジュールを使用すると、課金やネットワーク管理の目的で、トラフィックフローを常時監視できます。次の手順に従って、QoS ポリシーにフローアカウンティングを含める必要があるかどうか判断してください。

1. 顧客に SLA を提示していますか  
提示している場合は、フローアカウンティングを使用する必要があります。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。
2. ネットワークの問題を防ぐために監視またはテストを必要とするアプリケーションはありますか  
ある場合は、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、監視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。
3. QoS 計画テンプレートで、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。



## 次に進む手順

作業	参照先
QoS ポリシーにさらにクラスを追加する	39 ページの「QoS ポリシーのクラスを定義する方法」
QoS ポリシーにさらにフィルタを追加する	41 ページの「QoS ポリシーにフィルタを定義する方法」
フロー制御方式を定義する	43 ページの「フロー制御を計画する方法」
フローがネットワークストリームに戻される ときの転送動作を定義する	46 ページの「転送動作を計画する方法」
特定の種類のトラフィックに対して追加のフ ローアカウンティングを計画する	48 ページの「フローアカウンティングを計画 する方法」
IPQoS 構成ファイルを作成する	58 ページの「IPQoS 構成ファイルの開始方法 およびトラフィッククラスの定義方法」

---

## IPQoS の構成例の紹介

このマニュアルの残りの章で説明する作業では、この節で紹介する IPQoS の構成例を使用します。この例は、BigISP と呼ばれる架空のサービスプロバイダのパブリックイントラネットに実装される差別化サービスのソリューションを示しています。BigISP は、専用回線で BigISP に接続する大企業と、モデムから電話回線で BigISP に接続する個人の、2 種類のユーザーにサービスを提供しています。

### 例 — IPQoS トポロジ

次の図は、BigISP のパブリックイントラネットに使用されるネットワークトポロジを示しています。

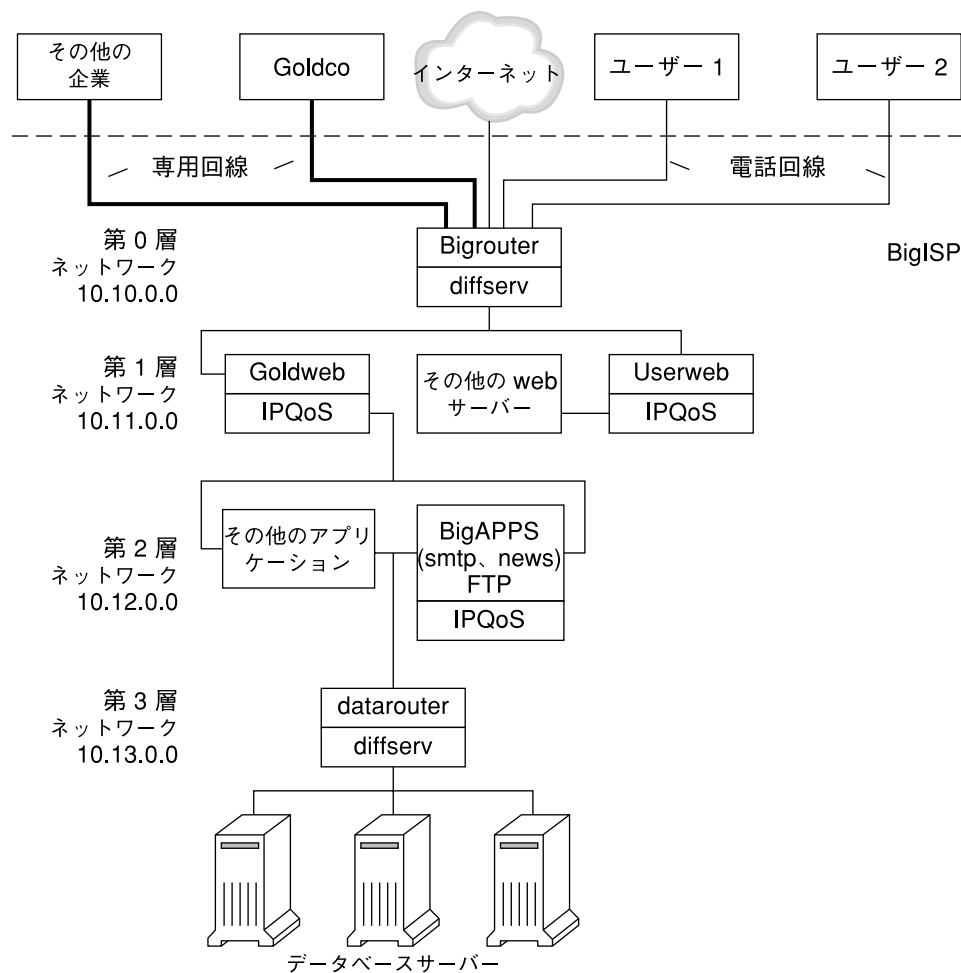


図 2-4 IPQoS のトポロジの例

前の図に示すように、BigISP のパブリックイントラネットには、4つの層が実装されています。

- 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 サイトのホストとして動作します。Goldweb と Userweb のどちらの

サーバーも 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

# IPQoS 構成ファイルの作成 (手順)

この章では、IPQoS 構成ファイルの作成方法および `ipqosconf` ユーティリティの使用方法を示します。

この章では、以下の内容について説明します。

- 53 ページの「IPQoS 構成ファイル内での QoS ポリシーの定義 (作業マップ)」
- 54 ページの「QoS ポリシー作成用のツール」
- 55 ページの「Web サーバー用 IPQoS 構成ファイルの作成」
- 70 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」
- 81 ページの「ルーター上での差別化サービスの提供」

この章の説明は、完全な QoS ポリシーを保持していること、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 ポリシーの計画方法については、36 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。

## IPQoS 構成ファイル内での QoS ポリシーの定義 (作業マップ)

次の表に、IPQoS 構成ファイルを作成するための一般的な作業を示します。

表 3-1 IPQoS 構成ファイルの作成 (作業マップ)

作業	説明	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	38 ページの「IPQoS 用のネットワークを準備する方法」

表 3-1 IPQoS 構成ファイルの作成 (作業マップ) (続き)

作業	説明	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明確なサービスクラスとして識別し、トラフィックの管理を必要とするフローを決定する	36 ページの「サービス品質ポリシーの計画」
3. IPQoS 構成ファイル作成を開始し、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	58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」
4. クラス用のフィルタを作成する	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	60 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. 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	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	67 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. メータリングモジュールを構成するパラメータを含むアクション文を追加する	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	77 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. マーカーを構成するパラメータを含むアクション文を追加する	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	62 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. 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アクション文を追加する	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	65 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. IPQoS 構成ファイルを使用する	作成した 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	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」
10. 転送動作をルーターファイル内で構成する	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	81 ページの「IPQoS 対応ネットワーク上でルーターを構成する方法」

## QoS ポリシー作成用のツール

ネットワーク用の QoS ポリシーは、IPQoS 構成ファイル内に格納します。テキストエディタでこの構成ファイルを作成し、このファイルを IPQoS 構成ユーティリティ `ipqosconf` の引数として指定します。`ipqosconf` に対し、構成ファイル内で定義されたポリシーの適用を指示すると、ポリシーがカーネル 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。`ipqosconf` の使用方法については、84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」を参照してください。

## IPQoS 構成ファイル

IPQoS 構成ファイルは、いくつかの「アクション文」から成るツリー構造をとりまします。これらのアクション文は、36 ページの「サービス品質ポリシーの計画」で定義した QoS ポリシーを実装します。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。

IPQoS 構成ファイルの完全な構文については、例 6-3 および `ipqosconf (1M)` のマニュアルページを参照してください。

## IPQoS トポロジ例の構成

この章では、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 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 つの構成ファイルを通して、最も一般的な IPQoS 構成を示します。これらをテンプレートとして使用して、独自の IPQoS 実装を作成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

この節では、まず、プレミアム Web サーバー用の構成ファイルの作成を通して、IPQoS 構成ファイルの作成方法を示します。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。どちらのサーバーも、図 2-4 に示したネットワーク例の一部です。

次の構成ファイルは、プレミアム SLA を購入した Goldco 社の Web サイトのホストとして機能する、Goldweb サーバーの IPQoS アクティビティを定義します。

例 3-1 プレミアム Web サーバー用 IPQoS 構成ファイルの例

```
fmt_version 1.0

action {
```

例 3-1 プレミアム Web サーバー用 IPQoS 構成ファイルの例 (続き)

```
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
    }
}
```



例 3-1 プレミアム Web サーバー用 IPQoS 構成ファイルの例 (続き)

```
    max_limit 2048
  }
}
```

次の構成ファイルは、低価格のベストエフォート SLA を利用する個人ユーザー向け Web サイトのホストとして機能する、Userweb サーバーの IPQoS アクティビティを定義します。この SLA 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 構成ファイルの開始方法およびトラフィッククラスの定義方法

最初の IPQoS 構成ファイルは、保守および使用しやすい任意のディレクトリに作成できます。この章では、IPQoS 構成ファイルの位置としてディレクトリ `/var/ipqos` を使用します。次の手順では、例 3-1 に示した IPQoS 構成ファイルの初期セグメントを構築します。

---

注 – IPQoS 構成ファイルを作成する際、各アクション文および句を必ず中括弧 ({} ) で囲んでください。括弧の使用例については、例 3-1 を参照してください。

---

1. プレミアム **Web** サーバーにログインし、新規 **IPQoS** 構成ファイルを拡張子 `.qos` を付けて作成します。

すべての IPQoS 構成ファイルで、最初の非コメント行にバージョン番号 `fmt_version 1.0` を記述する必要があります。

2. 冒頭のパラメータに続き、初期アクション文を記述して汎用の **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アクション文のツリーは、この初期アクションから始まります。たとえば、`/var/ipqos/Goldweb.qos` ファイルは、`ipgpc` クラシファイアを呼び出す初期アクション文で始まります。

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
```

エントリ	説明
<code>fmt_version 1.0</code>	IPQoS 構成ファイルを開始する
<code>action {</code> <code>module ipgpc</code>	構成ファイル内の最初のアクションとして <code>ipgpc</code> クラシファイアを構成する
<code>name ipgpc.classify</code>	クラシファイアのアクション文の名前を定義する。名前は常に <code>ipgpc.classify</code> でなければならない

アクション文の構文情報の詳細については、111 ページの「アクション文」および `ipqosconf (1M)` のマニュアルページを参照してください。

3. 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 アクション内でパラメータ `global_stats TRUE` を使用すると、このアクションに関する統計の取得が有効になります。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取得が可能になります。

統計の取得を有効にすると、パフォーマンスが影響を受けます。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、統計取得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、統計取得を無効にできます。

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` 句の構文およびその他の詳細については、114 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。

4. 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
}
```

上の文は、「クラス句」と呼ばれます。クラス句には次の内容が含まれます。

エントリ	説明
<code>name goldweb</code>	<code>goldweb</code> 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
<code>next_action markAF11</code>	<code>ipgpc</code> モジュールに対し、 <code>goldweb</code> クラスのケットをアクション文 <code>markAF11</code> に渡すよう指示する。アクション文 <code>markAF11</code> は、 <code>dscpmk</code> マーカーを呼び出す
<code>enable_stats FALSE</code>	<code>goldweb</code> クラスの統計取得を可能にする。ただし、 <code>enable_stats</code> の値が <code>FALSE</code> であるため、このクラスの統計取得は有効にはならない

クラス句の構文の詳細については、113 ページの「クラス句」および `ipqosconf (1M)` のマニュアルページを参照してください。

5. 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
}
```

エントリ	説明
<b>name video</b>	video クラスを作成して、Goldweb 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
<b>next_action markEF</b>	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、アクション文 markEF に渡すよう指示する。アクション文 markEF は、dscpmk マーカーを呼び出す
<b>enable_stats FALSE</b>	video 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、このクラスの統計取得は有効にはならない

## 次に進む手順

作業	参照先
作成したばかりのクラス用のフィルタを定義する	60 ページの「IPQoS 構成ファイル内でフィルタを定義する方法」
構成ファイル用の別のクラス句を作成する	58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」

## ▼ IPQoS 構成ファイル内でフィルタを定義する方法

次の手順では、IPQoS 構成ファイル内でクラス用のフィルタを定義する方法を示します。次の手順の前に、構成ファイルを開始しており、クラスを定義してあるものとします。ここでは、58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」で紹介した `/var/ipqos/Goldweb.qos` ファイルの構築を続けます。

注 – IPQoS 構成ファイルを作成する際、各クラス句およびフィルタ句を必ず中括弧 (`{ }`) で囲んでください。括弧の使用例については、例 3-1 を参照してください。

### 1. IPQoS 構成ファイルを開き、最後に定義したクラスの末尾を探します。

たとえば、IPQoS 対応サーバー Goldweb 用の構成ファイル `/var/ipqos/Goldweb.qos` では、次のクラス句のあとから作業を始めます。

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

## 2. IPQoS システムの発信トラフィックを選択するフィルタ句を定義します。

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

エントリ	説明
<b>name webout</b>	フィルタに webout という名前を付ける
<b>sport 80</b>	ソースポート 80 のトラフィックを選択する。これは、既知の HTTP (Web) トラフィック用ポート
<b>direction LOCAL_OUT</b>	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
<b>class goldweb</b>	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 構成ファイル内のフィルタ句の構文および詳細情報については、113 ページの「フィルタ句」を参照してください。

## 3. 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フィルタ句を定義します。

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

エントリ	説明
<b>name videoout</b>	フィルタに videoout という名前を付ける
<b>sport videosrv</b>	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
<b>direction LOCAL_OUT</b>	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
<b>class video</b>	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

## 次に進む手順

作業	参照先
マーカーモジュールの転送動作を定義する	62 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」
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	77 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」
IPQoS 構成ファイルを有効にする	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」
追加のフィルタを定義する	60 ページの「IPQoS 構成ファイル内でフィルタを定義する方法」
アプリケーションからのトラフィックフロー用のクラスを作成する	72 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方法」

### ▼ IPQoS 構成ファイル内でトラフィック転送を定義する方法

次の手順では、クラスのホップ単位動作を IPQoS 構成ファイルに追加して、トラフィック転送を定義する方法を示します。次の手順の前に、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。ここでは、例 3-1 に示した /var/ipqos/Goldweb.qos ファイルの構築を続けます。

---

注 - 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。dlcosmk マーカーを使用した VLAN システムでのトラフィック転送については、106 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。

---

#### 1. IPQoS 構成ファイルを開き、最後に定義したフィルタの末尾を探します。

たとえば、IPQoS 対応サーバー Goldweb 用の構成ファイル /var/ipqos/Goldweb.qos では、次のフィルタ句のあとから作業を始めます。

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

このフィルタは、ipgpc クラシファイアのアクション文の最後に位置します。このため、フィルタを終了させる閉じ括弧のあとに、アクション文を終了させる閉じ括弧が必要です。

2. 次のアクション文を使用してマーカーを呼び出します。

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

エントリ	説明
<b>module dscpmk</b>	dscpmk マーカーモジュールを呼び出す
<b>name markAF11</b>	アクション文に markAF11 という名前を付ける

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。

3. 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 FALSE
    dscp_map{0-63:10}
    next_action continue
}
}
```

エントリ	説明
<b>global_stats FALSE</b>	マーカーアクション文 markAF11 の統計取得を可能にする。ただし、global_stats の値が FALSE であるため、統計取得は有効にはならない
<b>dscp_map{0-63:10}</b>	DS コードポイント 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
<b>next_action continue</b>	goldweb 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

DS コードポイント 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、DS コードポイント 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 で利用可能な DS コードポイントについては、表 6-2 を参照してください。

4. 別のマーカーアクション文を開始します。

```

action {
    module dscpmk
    name markEF
}

```

エントリ	説明
<code>module dscpmk</code>	dscpmk マーカーモジュールを呼び出す
<code>name markEF</code>	アクション文に markEF という名前を付ける

5. 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
    }
}

```

エントリ	説明
<code>global_stats TRUE</code>	video クラスの統計取得を有効にする。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
<code>dscp_map{0-63:46}</code>	DS コードポイント 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
<code>next_action acct</code>	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、アクション文 acct に渡すよう指示する。アクション文 acct は flowacct モジュールを呼び出す

DS コードポイント 46 は、dscpmk モジュールに対し、dscp マップ内のすべてのエントリを DS フィールド内で 10 進数値の 46 (バイナリ値 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 EF ホップ単位動作 (PHB) に従うことを示します。

注 - 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 DS コードポイントは、AF PHB をパケットに割り当てるときに使用します。

EF PHB は、DS コードポイント 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。完全優先転送 PHB の詳細については、104 ページの「完全優先転送 (Expedited Forwarding, EF) PHB」を参照してください。



- 作成したばかりの **DS** コードポイントを **diffserv** ルーターの適切なファイルに追加します。詳細については、**81** ページの「**IPQoS** 対応ネットワーク上でルーターを構成する方法」を参照してください。

## 次に進む手順

作業	参照先
トラフィックフローに関するフローアカウンティング統計の収集を開始する	65 ページの「 <b>IPQoS</b> 構成ファイル内でクラスのアカウントリングを有効にする方法」
マーカーモジュールの転送動作を定義する	62 ページの「 <b>IPQoS</b> 構成ファイル内でトラフィック転送を定義する方法」
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	77 ページの「 <b>IPQoS</b> 構成ファイル内でフロー制御を構成する方法」
<b>IPQoS</b> 構成ファイルを有効にする	84 ページの「新規構成の <b>IPQoS</b> カーネルモジュールへの適用方法」
追加のフィルタを定義する	60 ページの「 <b>IPQoS</b> 構成ファイル内でフィルタを定義する方法」
アプリケーションからのトラフィックフロー用のクラスを作成する	72 ページの「アプリケーションサーバー用 <b>IPQoS</b> 構成ファイルの作成方法」

### ▼ **IPQoS** 構成ファイル内でクラスのアカウントリングを有効にする方法

次の手順では、**IPQoS** 構成ファイル内でトラフィッククラスのアカウントリングを有効にする方法を示します。次の手順の前に、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。ここでは、例 3-1 に示した `/var/ipqos/Goldweb.qos` ファイルの構築を続けます。

次の手順では、58 ページの「**IPQoS** 構成ファイルの開始方法およびトラフィッククラスの定義方法」で示した `video` クラス用のフローアカウンティングを定義する方法を示します。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。

- IPQoS** 構成ファイルを開き、最後に定義したアクション文の末尾を探します。

たとえば、**IPQoS** 対応サーバー `Goldweb` 用の構成ファイル `/var/ipqos/Goldweb.qos` では、次のアクション文 `markEF` 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
        }
    }
}

```

2. フローカウンティングを呼び出すアクション文を開始します。

```

    action {
        module flowacct
        name acct
    }

```

エントリ	説明
<code>module flowacct</code>	flowacct フローカウンティングモジュールを呼び出す
<code>name acct</code>	アクション文に acct という名前を付ける

3. トラフィッククラスに関する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
    }
}

```

エントリ	説明
<code>global_stats TRUE</code>	video クラスの統計取得を有効にする。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
<code>timer 10000</code>	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
<code>timeout 10000</code>	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
<code>max_limit 2048</code>	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
<code>next_action continue</code>	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、指定されたタイムアウト値に達するまで、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。

## 次に進む手順

作業	参照先
ルーターのホップ単位動作を構成する	81 ページの「IPQoS 対応ネットワーク上でルーターを構成する方法」
IPQoS 構成ファイルを有効にする	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」
アプリケーションからのトラフィックフロー用のクラスを作成する	72 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方法」

### ▼ 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 構成ファイルとは若干異なります。次の手順を通して、異なるレベルの Web サービス用の構成ファイル間に見られる、類似点および相違点を知ることができます。次の手順では、例 3-2 に示した構成ファイルを使用します。

1. ベストエフォート **Web** サーバーにログインします。
2. 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
    }
}
```

`/var/ipqos/userweb.qos` ファイルは、`ipgpc` クラシファイアを呼び出す部分アクション文から始める必要があります。このアクション文には、統計取得を有効にする `params` 句も含めています。このアクション文については、58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」を参照してください。

3. 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
    }

```

エントリ	説明
<b>name userweb</b>	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
<b>next_action markAF12</b>	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、アクション文 markAF12 に渡すよう指示する。アクション文 markAF12 は、dscpmk マーカーを呼び出す
<b>enable_stats FALSE</b>	userweb 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、このクラスの統計取得は有効にはならない

クラス句の作業については、58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」を参照してください。

#### 4. userweb クラスのトラフィックフローを選択するフィルタ句を定義します。

```

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

```

エントリ	説明
<b>name webout</b>	フィルタに webout という名前を付ける
<b>sport 80</b>	ソースポート 80 のトラフィックを選択する。これは、既知の HTTP (Web) トラフィック用ポート
<b>direction LOCAL_OUT</b>	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
<b>class userweb</b>	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

フィルタ句の作業については、60 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。

#### 5. dscpmk マーカーを呼び出すアクション文を開始します。

```

action {
    module dscpmk
    name markAF12
}

```

エントリ	説明
<code>module dscpmk</code>	dscpmk マーカーモジュールを呼び出す
<code>name markAF12</code>	アクション文に markAF12 という名前を付ける

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、アクション文 markAF12 に送信します。

6. 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
}

```

エントリ	説明
<code>global_stats FALSE</code>	マーカーアクション文 markAF12 の統計取得を可能にする。ただし、 <code>global_stats</code> の値が FALSE であるため、統計取得は有効にはならない
<code>dscp_map{0-63:12}</code>	DS コードポイント 12 を、マーカーにより処理中の userweb クラスの packets ヘッダーに割り当てる
<code>next_action continue</code>	userweb クラスの packets 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 packets をネットワークストリームに戻してもよいことを示す

DS コードポイント 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 packets 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 DS コードポイント 12 を持つすべての packets 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。

IPQoS 構成ファイルの作成が完了したら、84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」の記述に従って構成を適用します。

## 次に進む手順

作業	参照先
アプリケーションからのトラフィックフロー用のクラスおよび他の構成を追加する	72 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方法」
ルーターのホップ単位動作を構成する	81 ページの「IPQoS 対応ネットワーク上でルーターを構成する方法」
IPQoS 構成ファイルを有効にする	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」

---

# アプリケーションサーバー用 IPQoS 構成ファイルの作成

この節では、組織内外の顧客に多数の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。次の手順では、図 2-4 に示した BigAPPS サーバーを例として使用します。

次の構成ファイルは、顧客の FTP、電子メール (SMTP)、およびネットワークニュース (NNTP) のホストとして機能する BigAPPS サーバーの、IPQoS アクティビティを定義します。

### 例 3-3 アプリケーション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 smtp
    enable_stats FALSE
    next_action markAF13
  }
  class {
    name news
    next_action markAF21
  }
  class {
    name ftp
    next_action meterftp
  }
}
```

例 3-3 アプリケーションサーバー用の構成例 (続き)

```
    }
    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 markAF31
        green_action_name markAF22
        global_stats TRUE
    }
}
action {
    module dscpmk
```

例 3-3 アプリケーションサーバー用の構成例 (続き)

```
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 構成ファイルの作成方法

1. IPQoS 対応アプリケーションサーバーにログインし、新規 IPQoS 構成ファイルを拡張子 .qos を付けて作成します。

たとえば、アプリケーションサーバー用に /var/ipqos/BigAPPS.qos ファイルを作成します。アクション文の最初に、ipgpc クラシファイアを呼び出す以下の記述を配置します。これらは必ず記述する必要があります。

```
fmt_version 1.0

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

冒頭のアクション文については、58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」を参照してください。

2. 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それぞれ選択するクラスを作成します。

冒頭のアクション文のあとに、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
    }

```

エントリ	説明
<b>name smtp</b>	smtp という名前のクラスを作成する。このクラスには、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
<b>enable_stats FALSE</b>	smtp 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、このクラスの統計取得は有効にはならない
<b>next_action markAF13</b>	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、アクション文 markAF13 に渡すよう指示する
<b>name news</b>	news という名前のクラスを作成する。このクラスには、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
<b>next_action markAF21</b>	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
<b>name ftp</b>	ftp という名前のクラスを作成する。このクラスには、FTP アプリケーションが扱う発信トラフィックが含まれる
<b>enable_stats TRUE</b>	ftp クラスの統計取得を有効にする
<b>next_action meterftp</b>	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、アクション文 meterftp に渡すよう指示する

クラス定義の詳細については、58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」を参照してください。

### 3. 定義済みのクラスのトラフィックを選択するフィルタ句を定義します。

```

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
}
}

```

エントリ	説明
<b>name smtpout</b>	フィルタに smtpout という名前を付ける
<b>sport smtp</b>	ソースポート 25 のトラフィックを選択する。これは、既知の sendmail (SMTP) アプリケーション用ポート
<b>class smtp</b>	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
<b>name newsout</b>	フィルタに newsout という名前を付ける
<b>sport nntp</b>	ソースポート名 nntp のトラフィックを選択する。これは、既知のネットワークニュースアプリケーション用ポート
<b>class news</b>	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
<b>name ftpout</b>	フィルタに ftpout という名前を付ける
<b>sport ftp</b>	ソースポート 21 の制御データを選択する。これは、既知の FTP トラフィック用ポート番号
<b>name ftpdata</b>	フィルタに ftpdata という名前を付ける
<b>sport ftp-data</b>	ソースポート 20 のトラフィックを選択する。これは、既知の FTP データトラフィック用ポート番号
<b>class ftp</b>	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

フィルタ定義の詳細については、60 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。

## 次に進む手順

作業	参照先
フィルタを定義する	60 ページの「IPQoS 構成ファイル内でフィルタを定義する方法」
アプリケーショントラフィックの転送動作を定義する	75 ページの「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」
メータリングモジュールを使用してフロー制御を設定する	77 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」
フローアカウンティングを構成する	65 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」

### ▼ 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を構成する方法を示します。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、ネットワーク上の他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。次の手順の前に、既存の IPQoS 構成ファイルに、マークを付け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します。ここでは、例 3-3 に示した `/var/ipqos/BigAPPS.qos` ファイルの構築を続けます。

1. アプリケーションサーバー用に作成した **IPQoS** 構成ファイルを開きます。

最後のフィルタ句の末尾を探します。`/var/ipqos/BigAPPS.qos` ファイルでは、最後のフィルタは次のとおりです。

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

2. 次の方法でマーカーを呼び出します。

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

エントリ	説明
<code>module dscpmk</code>	dscpmk マーカーモジュールを呼び出す
<code>name markAF13</code>	アクション文に markAF13 という名前を付ける

3. 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
    }
}

```

エントリ	説明
<code>global_stats FALSE</code>	マーカーアクション文 markAF13 の統計取得を可能にする。ただし、global_stats の値が FALSE であるため、統計取得は有効にはならない
<code>dscp_map{0-63:14}</code>	DS コードポイント 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
<code>next_action continue</code>	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

DS コードポイント 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。この値により、AF13 ホップ単位動作 (PHB) が設定され、smtp トラフィッククラスのパケットの DS フィールドに DS コードポイント 14 が指定されます。

AF13 は、DS コードポイント 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。利用可能な AF コードポイントについては、表 6-2 を参照してください。

4. マーカーアクション文を追加して、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
    }
}

```

次の表に、これまでの手順で説明の済んでいないパラメータだけを示します。

エン트리	説明
<code>name markAF21</code>	アクション文に markAF21 という名前を付ける
<code>dscp_map{0-63:18}</code>	DS コードポイント 18 を、マーカーにより処理中のトラフィッククラス nntp のパケットヘッダーに割り当てる

DS コードポイント 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。この値により、AF21 ホップ単位動作 (PHB) が設定され、news トラフィッククラスのパケットの DS フィールドに DS コードポイント 18 が指定されます。

AF21 は DS コードポイント 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。このため、ネットワークニュースの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、ルーターは Class 1 のトラフィッククラスの転送を優先します。

## 次に進む手順

作業	参照先
Web サーバーの構成情報を追加する	58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」
メータリングモジュールを使用してフロー制御を構成する	77 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」
フローアカウンティングを構成する	65 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」
ルーターの転送動作を構成する	81 ページの「IPQoS 対応ネットワーク上でルーターを構成する方法」
IPQoS 構成ファイルを有効にする	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」

## ▼ IPQoS 構成ファイル内でフロー制御を構成する方法

特定のトラフィックフローがネットワークに送出される速度を制御するには、メーターのパラメータを定義する必要があります。IPQoS 構成ファイル内で、2 つのメータリングモジュール tokenmt と tswtclmt とのどちらかを使用できます。

ここでは、例 3-3 に示したアプリケーションサーバー用 IPQoS 構成ファイルの構築を続けます。次の手順では、メーターを構成するだけでなく、メーターアクション文の内部で呼び出される 2 つのマーカーアクションも構成します。

1. アプリケーションサーバー用に作成した **IPQoS** 構成ファイルを開きます。

残りの手順の前に、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。/var/ipqos/BigAPPS.qos ファイルで、次のマーカーアクションのあとから作業を開始します。

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

2. ftp クラスのトラフィックをフロー制御するメーターアクション文を作成します。

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

エントリ	定義
<code>module tokenmt</code>	tokenmt メーターを呼び出す
<code>name meterftp</code>	アクション文に meterftp という名前を付ける

3. メーターの速度を設定するパラメータを追加します。

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

エントリ	説明
<code>committed_rate 50000000</code>	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
<code>committed_burst 50000000</code>	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 パラメータの詳細については、101 ページの「tokenmt をツールレートメーターとして構成する」を参照してください。

4. トラフィック適合度ごとの優先度を設定するパラメータを追加します。

```
red_action_name markAF31
green_action_name markAF22
global_stats TRUE
```

```

    }
}

```

エントリ	説明
<code>red_action_name markAF31</code>	ftp クラスのトラフィックフローが不適合になった (認定速度を超過した) 場合、パケットがマーカーアクション文 markAF31 に送られることを示す
<code>green_action_name markAF22</code>	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
<code>global_stats TRUE</code>	ftp クラスのメータリング統計取得を有効にする

トラフィック適合の詳細については、100 ページの「メーターモジュール」を参照してください。

5. 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アクション文を追加します。

```

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

```

エントリ	説明
<code>module dscpmk</code>	dscpmk マーカーモジュールを呼び出す
<code>name markAF31</code>	アクション文に markAF31 という名前を付ける
<code>global_stats TRUE</code>	ftp クラスの統計取得を有効にする
<code>dscp_map{0-63:26}</code>	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、DS コードポイント 26 を ftp クラスのパケットヘッダーに割り当てる
<code>next_action continue</code>	ftp 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

DS コードポイント 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。この値により、AF31

ホップ単位動作 (PHB) が設定され、ftp トラフィッククラスのパケットの DS フィールドに DS コードポイント 26 が指定されます。

AF31 は、DS コードポイント 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、ルーターは、Class 1 のトラフィッククラス、または Class 2 で低い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。利用可能な AF コードポイントについては、表 6-2 を参照してください。

6. 認定速度に適合する ftp クラスの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アクション文を追加します。

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

次の表に、これまでの手順で説明の済んでいないパラメータだけを示します。

エントリ	説明
<code>name markAF22</code>	marker アクションに markAF22 という名前を付ける
<code>dscp_map{0-63:20}</code>	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、DS コードポイント 20 をパケットヘッダーに割り当てる

DS コードポイント 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。この値により、AF22 ホップ単位動作 (PHB) が設定され、ftp トラフィッククラスのパケットの DS フィールドに DS コードポイント 20 が指定されます。

AF22 は、DS コードポイント 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。利用可能な AF コードポイントについては、表 6-2 を参照してください。

7. アプリケーションサーバー用に作成した DS コードポイントを、diffserv ルーターの適切なファイルに追加します。詳細については、81 ページの「IPQoS 対応ネットワーク上でルーターを構成する方法」を参照してください。



## 次に進む手順

作業	参照先
IPQoS 構成ファイルを有効にする	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」
Web サーバーの構成情報を追加する	58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」
フローカウンティングを構成する	65 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」
ルーターの転送動作を構成する	81 ページの「IPQoS 対応ネットワーク上でルーターを構成する方法」

## ルーター上での差別化サービスの提供

厳密な意味で差別化サービスを提供するには、32 ページの「diffserv ネットワークのハードウェア計画」の記述に従って、ネットワークトポロジに diffserv 対応の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、このマニュアルの扱う範囲ではありません。

この節では、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。次の手順の前に、この章のこれまでの作業を実行することにより、ネットワーク上で IPQoS システムを構成してあるものとします。

### ▼ IPQoS 対応ネットワーク上でルーターを構成する方法

次の手順では、図 2-4 に示したトポロジを例として使用します。

1. ネットワーク上のすべての IPQoS 対応システムの構成ファイルを確認します。
2. さまざまなポリシーで使用される各コードポイントを特定します。

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の表を作成します。作成した表から、同じコードポイントを使用した領域を知ることができます。同じコードポイントを使用したままでもかまいませんが、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する必要があります。

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。

表 3-2 ネットワーク例用に構成されたホップ単位動作

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

3. 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、**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** スケジューリング機構の設定に役立ちます。具体的な方法については、ルーター製造元のマニュアルおよび Web サイトを参照してください。

## 第 4 章

# IPQoS の起動と保守(手順)

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。

この章では、以下の内容について説明します。

- 83 ページの「IPQoS の管理 (作業マップ)」
- 84 ページの「IPQoS 構成の有効化」
- 85 ページの「IPQoS メッセージの syslog によるログ記録の有効化」
- 86 ページの「IPQoS エラーメッセージの使用」

## IPQoS の管理 (作業マップ)

次の表に、Solaris システムでの IPQoS の起動作業および保守作業を示します。作業を実行する前に、53 ページの「IPQoS 構成ファイル内での QoS ポリシーの定義 (作業マップ)」の記述に従って IPQoS 構成ファイルを完了しておく必要があります。

表 4-1 IPQoS の構成と保守 (作業マップ)

作業	説明	参照先
1. システムで IPQoS を構成する	ipqosconf ユーティリティを使用して、システムの IPQoS 構成ファイルを有効化する	84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」
2. Solaris 起動スクリプトを使用して、各システムの起動後にデバッグ済みの IPQoS 構成ファイルを適用する	システムをリポートするたびに、IPQoS 構成ファイルが確実に適用されるようにする	85 ページの「リポート後にも IPQoS 構成を適用する方法」

表 4-1 IPQoS の構成と保守 (作業マップ) (続き)

作業	説明	参照先
3. syslog を使用した IPQoS のログ記録を有効にする	エントリを追加して、syslog による IPQoS メッセージのログ記録を有効にする	86 ページの「ブート時に IPQoS メッセージを記録する方法」
4. IPQoS に発生した問題を修正する	エラーメッセージを利用して IPQoS の問題を解決する	表 4-2 に示すエラーメッセージを参照

## IPQoS 構成の有効化

IPQoS 構成の有効化およびそのほかの操作には、`ipqosconf` コマンドを使用します。

### ▼ 新規構成の IPQoS カーネルモジュールへの適用方法

IPQoS 構成ファイルを読み取り、UNIX カーネル内で IPQoS モジュールを構成するには、`ipqosconf` ツールを使用します。次の手順では、55 ページの「Web サーバー用 IPQoS 構成ファイルの作成」で作成したファイル `/var/ipqos/Goldweb.qos` を例として使用します。詳細については、`ipqosconf(1M)` のマニュアルページを参照してください。

1. IPQoS 対応のシステムでスーパーユーザーになります。
2. 新規構成を適用します。

```
# /usr/sbin/ipqosconf -a /var/ipqos/Goldweb.qos
```

`ipqosconf` により、指定された IPQoS 構成ファイル内の情報が Solaris カーネル内の IPQoS モジュールに書き込まれます。上の例では、`/var/ipqos/Goldweb.qos` の内容が現行の Solaris カーネルに適用されます。

---

注 `-a` オプションを指定して IPQoS 構成ファイルを適用すると、ファイル内のアクションが現行のセッションの間だけ有効になります。

---

3. 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。こうして、構成が期待どおりに機能するかどうかを調べます。

## 次に進む手順

作業	参照先
IPQoS モジュールの動作状況に関する統計を表示する	95 ページの「統計情報の収集」
ipqosconf メッセージをログに記録する	85 ページの「IPQoS メッセージの syslog によるログ記録の有効化」
現行の IPQoS 構成をリポート後も適用する	85 ページの「リポート後も IPQoS 構成を適用する方法」

### ▼ リポート後にも IPQoS 構成を適用する方法

リポート後にも IPQoS 構成を持続させるには、明示的に指定する必要があります。そのように指定しないと、システムのリポート後に現行の構成が適用されません。システムで IPQoS が適正に動作するときは、次の操作を実行してリポート後にも構成が持続するようにします。

1. IPQoS 対応のシステムにスーパーユーザーとしてログインします。
2. カーネルモジュール内に **IPQoS** 構成が存在することを確認します。

```
# ipqosconf -l
```

構成がすでに存在する場合は、ipqosconf によって画面に表示されます。出力が表示されない場合は、84 ページの「新規構成の IPQoS カーネルモジュールへの適用方法」の説明に従って構成を適用します。

3. IPQoS システムをリポートするたびに既存の **IPQoS** 構成が適用されるようにします。

```
# /usr/sbin/ipqosconf -c
```

-c オプションを指定すると、現行の IPQoS 構成が、ブート時の構成ファイル /etc/inet/ipqosinit.conf に書き込まれます。

---

## IPQoS メッセージの syslog によるログ記録の有効化

IPQoS ブート時のメッセージを記録するには、次に示す手順に従って /etc/syslog.conf ファイルを変更する必要があります。

## ▼ ブート時に IPQoS メッセージを記録する方法

1. IPQoS 対応のマシンでスーパーユーザーになります。
2. `/etc/syslog.conf` ファイルを開きます。
3. ファイルの最後に、次のエントリを追加します。  
列の区切りは、空白ではなくタブを使用してください。

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

このエントリを指定すると、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。

4. システムをリブートして設定を適用します。

### 例 — `/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、次のような IPQoS ログメッセージが出力されます。

```
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.
```

また、IPQoS システムの `/var/adm/messages` ファイル内に、次のような IPQoS エラーメッセージが表示される場合もあります。

```
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.
```

上記のエラーメッセージについては、表 4-2 を参照してください。

---

## IPQoS エラーメッセージの使用

次の表では、IPQoS が生成するエラーメッセージおよびその解決策を示します。

表 4-2 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 <i>parameter-name</i> 's action <i>action-name</i>	<i>parameter-name</i> に指定したアクション名が IPQoS 構成ファイル内に存在しない	アクションを作成するか、あるいは存在する別のアクションをパラメータ内で参照させる
action <i>action-name</i> involved in cycle	IPQoS 構成ファイル内の <i>action-name</i> はアクション循環の一部である。これは IPQoS では許可されない	アクション循環を調べ、IPQoS 構成ファイルからどちらかの循環参照を削除する
Action <i>action-name</i> isn't referenced by any other actions	ipgpc アクション以外で、他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 構成内にある。これは IPQoS では許可されない	参照されていないアクションを削除するか、あるい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る
Missing/Invalid config file <i>fmt_version</i>	構成ファイルのフォーマットがファイルの最初のエンタリに指定されていない。これは IPQoS では必須	58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」の説明に従ってフォーマットのバージョンを追加する
Unsupported config file format version	IPQoS がサポートしないフォーマットのバージョンが、構成ファイル内で指定されている	フォーマットのバージョンを、Solaris 9 9/02 バージョンの IPQoS 実行に必要な <i>fmt_version</i> 1.0 に変更する
No ipgpc action defined.	構成ファイル内で、ipgpc クラスファイアのアクションが定義されていない。これは IPQoS では必須	58 ページの「IPQoS 構成ファイルの開始方法およびトラフィッククラスの定義方法」の説明に従って ipgpc のアクションを定義する
Can't commit a null configuration	ipgosconf -c を実行して空の構成をコミットしようとした。IPQoS は空の構成を許可しない	構成ファイルを確実に適用してから構成をコミットする
Invalid CIDR mask on line <i>line_number</i>	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する
Address masks aren't allowed for host names line <i>line_number</i>	構成ファイル内で、ホストの CIDR マスク値を定義した。これは IPQoS では許可されない	マスクを削除するか、あるいはホスト名を IP アドレスに変更する
Invalid module name line <i>line_number</i>	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た	モジュール名のスペルに入力ミスがないか確認する。IPQoS モジュールについては、表 6-5 を参照

表 4-2 IPQoS のエラーメッセージ (続き)

エラーメッセージ	説明	解決方法
ipgpc action has incorrect name line <i>line_number</i>	構成ファイル内で ipgpc アクションに付けた名前が、必須の ipgpc.classify ではない	アクション名を ipgpc.classify に変更する
Second parameter clause not supported line <i>line_number</i>	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た。これは IPQoS では許可されない	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する
Duplicate named action	構成ファイル内で、2 つのアクションに同じ名前を付けた	どちらかのアクションの名前を変更するか、あるいは削除する
Duplicate named filter/class in action <i>action_name</i>	1 つのアクション内の 2 つのフィルタまたは 2 つのクラスに同じ名前を付けた。これは IPQoS 構成ファイルでは許可されない	どちらかのフィルタまたはクラスの名前を変更するか、あるいは削除する
Undefined class in filter <i>filter_name</i> in action <i>action_name</i>	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する	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する
Undefined action in class <i>class_name</i> action <i>action_name</i>	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する	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する
Invalid parameters for action <i>action_name</i>	構成ファイル内のパラメータのどれかが無効である	名前付きのアクションで呼び出されるモジュールについては、97 ページの「IPQoS アーキテクチャと diffserv モデル」のモジュールに関する説明を参照する。あるいは、ipqosconf (1M) のマニュアルページを参照する
Mandatory parameter missing for action <i>action_name</i>	アクションに必要なパラメータが構成ファイル内に定義されていない	名前付きのアクションで呼び出されるモジュールについては、97 ページの「IPQoS アーキテクチャと diffserv モデル」のモジュールに関する説明を参照する。あるいは、ipqosconf (1M) のマニュアルページを参照する
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た。最大数は 10007	構成ファイルを確認して、不要なクラスを削除する。あるいは、/etc/system ファイルにエントリ ipgpc_max_classes <i>class_number</i> を追加して、クラスの最大数を引き上げる
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た。最大数は 10007	構成ファイルを確認して、不要なフィルタを削除する。あるいは、/etc/system ファイルにエントリ ipgpc_max_filters <i>filter_number</i> を追加して、フィルタの最大数を引き上げる



表 4-2 IPQoS のエラーメッセージ (続き)

エラーメッセージ	説明	解決方法
Invalid/missing parameters for filter <i>filter_name</i> in action <i>ipgpc</i> .	構成ファイル内で、フィルタ <i>filter_name</i> 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る	<code>ipqosconf (1M)</code> のマニュアルページで、有効なパラメータのリストを参照する
Name not allowed to start with '!', line <i>line_number</i>	アクション、フィルタ、またはクラスの名前が感嘆符 (!) で始まっている。これは IPQoS ファイルでは許可されない	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する
Name exceeds the maximum name length line <i>line_number</i>	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 文字を超えている	アクション、クラス、またはフィルタの名前を短くする
Array declaration line <i>line_number</i> is invalid	構成ファイル内で、 <i>line_number</i> 行のパラメータの配列宣言が無効である	アクション文で呼び出される配列宣言の正しい構文については、97 ページの「IPQoS アーキテクチャと <code>diffserv</code> モデル」を参照する。あるいは、 <code>ipqosconf (1M)</code> のマニュアルページを参照する
Quoted string exceeds line, <i>line_number</i>	文字列の最後の閉じ引用符が同一行に存在しない。これは構成ファイルでは必須	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る
Invalid value, line <i>line_number</i>	構成ファイルの <i>line_number</i> 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る。	アクション文で呼び出されるモジュールに指定可能な値については、97 ページの「IPQoS アーキテクチャと <code>diffserv</code> モデル」のモジュールに関する説明を参照する。あるいは、 <code>ipqosconf (1M)</code> のマニュアルページを参照する
Unrecognized value, line <i>line_number</i>	構成ファイルの <i>line_number</i> 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る	パラメータの列挙値が適正であるかどうかを確認する。認識されない行番号のアクション文で呼び出されるモジュールについては、97 ページの「IPQoS アーキテクチャと <code>diffserv</code> モデル」を参照する。あるいは、 <code>ipqosconf (1M)</code> のマニュアルページを参照する
Malformed value list line <i>line_number</i>	構成ファイルの <i>line_number</i> で指定された列挙が、仕様構文に適合しない	アクション文で呼び出されるモジュールの正しい構文については、97 ページの「IPQoS アーキテクチャと <code>diffserv</code> モデル」のモジュールに関する説明を参照する。あるいは、 <code>ipqosconf (1M)</code> のマニュアルページを参照する
Duplicate parameter line <i>line_number</i>	重複したパラメータが <i>line_number</i> に指定されている。これは構成ファイルでは許可されない	重複したパラメータのどちらかを削除する

表 4-2 IPQoS のエラーメッセージ (続き)

エラーメッセージ	説明	解決方法
Invalid action name line <i>line_number</i>	構成ファイルの <i>line_number</i> のアクションに、定義済みの名前 continue または drop を含む名前を付けた	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する
Failed to resolve src/dst host name for filter at line <i>line_number</i> , 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 が解釈処理できない。このため、このフィルタは無視される	フィルタが重要な場合、あとで構成の適用を試みる
Incompatible address version line <i>line_number</i>	<i>line_number</i> 上の IP バージョンのアドレスが、構成ファイル内ですでに指定済みの IP アドレスのバージョンまたは ip_version パラメータと互換性がない	競合する 2 つのエントリを変更して、互換性を持たせる
Action at line <i>line_number</i> 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た。これは許可されない	現行の構成をフラッシュしてから、新しい構成を適用する

## 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 (手順)

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。

この章では、以下の内容について説明します。

- 91 ページの「フローアカウンティングの設定 (作業マップ)」
- 92 ページの「フローに関する情報の記録」
- 95 ページの「統計情報の収集」

## フローアカウンティングの設定 (作業マップ)

次の表に、flowacct モジュールを使ってトラフィックフローに関する情報を取得するための一般的な作業を示します。

表 5-1 フローアカウンティングの設定 (作業マップ)

作業	説明	参照先
1. トラフィックフローのアカウンティング情報を格納するためのファイルを作成する	acctadm コマンドを使用して、flowacct による処理結果を格納するファイルを作成する	92 ページの「フローアカウンティングデータ用のファイルの作成方法」
2. flowacct のパラメータを IPQoS 構成ファイルに定義する	timer、timeout、および max_limit の各パラメータの値を定義する	65 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」

表 5-1 フローアカウンティングの設定 (作業マップ) (続き)

作業	説明	参照先
3. ファイルの内容を表示する	アカウンティングレコードをファイルから読み取るためのユーティリティの作成に使用できるプログラム例を表示する	94 ページの「フローアカウンティングファイルを表示する方法」

## フローに関する情報の記録

IPQoS の flowacct モジュールを使用すると、発信元アドレス、着信先アドレス、フローに含まれるパケット量などの、トラフィックフローに関する情報を収集できます。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 の属性については、表 6-4 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ために、非常に役立ちます。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。この節では、flowacct を 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作業について説明します。

この章以外の場所からも次の情報が入手できます。

- flowacct のアクション文を IPQoS 構成ファイルに作成する手順については、77 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
- flowacct の機能については、98 ページの「クラシファイアモジュール」を参照
- 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

### ▼ フローアカウンティングデータ用のファイルの作成方法

flowacct アクションを IPQoS 構成ファイルに追加する前に、flowacct モジュールから得られるフローレコードを格納するためのファイルを作成する必要があります。このためには、acctadm コマンドを使用します。acctadm では、基本属性または拡張属性のどちらもファイルに記録できます。flowacct のすべての属性については、表 6-4 を参照してください。acctadm の詳細については、acctadm (1M) のマニュアルページを参照してください。

1. スーパーユーザーとして **IPQoS** 対応システムにログインします。
2. 基本フローアカウンティングファイルを作成します。  
次の例では、例 3-1 で構成したプレミアム Web サーバー用に、基本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
```

文	定義
acctadm -e	-e オプションを使って acctadm を呼び出す。-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 によって次の出力が生成されます。

```
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、Solaris 9 のリソースマネージャ機能で使用されます。次の表では、IPQoS に固有のエントリについて説明します。

エントリ	説明
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

エントリ	説明
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	ファイルに記録されない flowacct の属性を示す

4. (オプション) 次の方法で、拡張属性をアカウントリ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トリ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トリングが無効になります。
```

## 次に進む手順

作業	参照先
flowacct のパラメータを IPQoS 構成ファイルに定義する	65 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」
acctadm を使って作成したファイルのデータを表示する	94 ページの「フローアカウンティングファイルを表示する方法」

## ▼ フローアカウンティングファイルを表示する方法

acctadm で作成したフローアカウンティングファイルの内容を表示するには、スクリプトを作成する必要があります。作成するスクリプトのベースとして、資源管理作業用およびアカウンティング処理用のデモスクリプトを使用できます。次の手順では、デモスクリプトに関する情報の入手方法を示します。

次の手順の前に、92 ページの「フローアカウンティングデータ用のファイルの作成方法」の説明に従って、フローレコードを格納するファイルを作成している必要があります。また、トラフィッククラスが flowacct によって追跡されるように、flowacct のアクションとパラメータを IPQoS 構成ファイルに追加している必要もあります。

次の手順では、プロセスや作業を表示するために acctadm ファイルの出力を行う、libexacct プログラムインタフェースと exdump ユーティリティを紹介します。技術的な情報については、libexacct (3LIB) のマニュアルページを参照してください。

1. **IPQoS** システムでスーパーユーザーになり、ディレクトリ /usr/demo/libexacct にアクセスします。

このディレクトリには、Makefile と exdump.c スクリプトが入っています。

2. README の説明に従って、exdump の構築手順を実行します。
3. README の説明に従って、フローカウンティングファイルのデータを表示します。

---

## 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。次の構文を使用します。

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

表 6-5 に示すように、有効な IPQoS モジュール名を任意に指定できます。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、次の形の kstat 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 — IPQoS の kstat 統計情報

ここでは、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
```

エントリ	説明
class: flacct	トラフィックフローが属するクラスの名前 (このインスタンスでは flacct) を示す

エン트리	説明
bytes_in_tbl	フローテーブルの総バイト数。すなわち、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合計バイト数は 84。フローテーブルにフローがまったく存在しない場合、bytes_in_tbl の値は 0 になる
crttime	この kstat が最後に作成された時間
epackets	処理中にエラーが発生したパケットの数 (このインスタンスでは 0)
flows_in_tbl	フローテーブルに格納されているフローレコードの数 (このインスタンスでは 1)。フローテーブルにレコードがまったく存在しない場合、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)。この値は、フローテーブルに現在格納されているバイト数と、タイムアウトしてすでにフローテーブルに存在しないバイト数とを含む
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 は、フローテーブルに現在格納されているパケットと、タイムアウトしてすでにフローテーブルに存在しないパケットとを含む
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

---

# IPQoS の詳細 (リファレンス)

---

この章は、IPQoS の詳細を説明するリファレンスです。

この章では、以下の内容について説明します。

- 97 ページの「IPQoS アーキテクチャと diffserv モデル」
- 110 ページの「IPQoS 構成ファイル」
- 114 ページの「ipqosconf 構成ユーティリティ」

IPQoS の概要については、第 1 章を参照してください。計画情報については、第 2 章を参照してください。IPQoS の構成手順については、第 3 章を参照してください。

---

## IPQoS アーキテクチャと diffserv モデル

この節では、IPQoS アーキテクチャとこのアーキテクチャが RFC 2475 (*An Architecture for Differentiated Services*) 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 にはフローカウンティングモジュールおよび VLAN デバイスで使用する `dlcosmk` マーカーも含まれます。

## 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。ipgpc の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、「セレクトラ」と呼ばれます。ipgpc クラシファイアは、これらのセレクトラを、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 (DS) コードポイントの存在を確認します  
DS コードポイントすなわち 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 packets 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については、23 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 構成ファイル内でのクラシファイアの呼び出しに関しては、110 ページの「IPQoS 構成ファイル」を参照してください。

## セレクトラ

ipgpc は、IPQoS 構成ファイルのフィルタ句で使用可能なさまざまなセレクトラをサポートします。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トラ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 で利用可能なセレクトラを示します。

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	DS コードポイント。デフォルトはゼロ (0)	DS コードポイント。パケットに適用される転送動作を定義する
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
if_groupname	インタフェースグループ名	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グループ
user	選択する UNIX userID の番号またはユーザー名。パケットに userID 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 UserID
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 packets 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	次の引数のいずれかを指定できる。	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

表 6-1 IPQoS クラシファイアで利用可能なフィルタセクタ (続き)

セクタ	引数	選択される情報
	0	LOCAL_IN と LOCAL_OUT、または FORWARD_IN と FORWARD_OUT を表すワイルドカード
precedence	優先度の値。もっとも高い優先度は 0	優先度は、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マ (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、設定されたパラメータにパケットが適合するかどうかを調べます。メーターモジュールは、パケットサイズ、設定されたパラメータ、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設定できます。

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。

- 認定速度 - 特定クラスの packets に容認可能な転送速度を bps で定義する
- 最大速度 - 特定クラスの packets 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。

- 緑 - パケットの生成するフローは認定速度内である
- 黄 - 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
- 赤 - 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。認定速度および最大速度については、次に説明します。

## tokenmt メータリングモジュール

`tokenmt` モジュールは、「トークンパケット」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 つのトークンパケットを管理します。これらのトークンパケットは、トラフィックフローが設定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は、IPQoS がトークンメーターパラダイムを実装する方法を説明しています。トークンパケットに関する一般的な情報は、Kalevi Kilkki 著『*Differentiated Services for the Internet*』および多数の Web サイトで入手できます。

tokenmt の構成パラメータを次に示します。

- `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することもできます。2つの結果を識別するシングルレート tokenmt の例については、例 3-3 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バーストサイズを超過します。`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 アクション用の `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、すべて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」を有効にするパラメータを追加する必要があります。次に、カラーアウェアの tokenmt を構成するアクション文の例を示します。

#### 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、ツール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 DS コードポイントも使用してパケットを評価します。

color\_map パラメータには、パケットヘッダー内の DSCP がマッピングされた配列が含まれます。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持っていますが、必要に応じ color\_map パラメータを使用してカラーマップを変更できます。

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 markAF22 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「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。速度概算アルゴリズムは、RFC 2859 (A Time Sliding Window Three Colour Marker) に基づいています。

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` – タイムウィンドウをミリ秒で定義する。このタイム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。`tswtclmt` に似た速度シェーパの一般的な情報については、RFC 2963 (*A Rate Adaptive Shaper for Differentiated Services*) 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。ここでは、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` を使用する必要があります。

`dscpmk` の技術情報については、`dscpmk` (7ipp) のマニュアルページを参照してください。`dlcosmk` の技術情報については、`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、トラフィックに転送動作のマークを付けます。転送動作は、フローが IPQoS システムを離れたあとに実行されます。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。PHB の詳細については、28 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既知の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 を割り当てるときには、IPQoS システムはパケットヘッダーの DS フィールドに、差別化サービス (DS) コードポイントすなわち DSCP と呼ばれる値を付けます。`diffserv` アーキテクチャは、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 DS コードポイントを使用します。DS コードポイントの概要については、27 ページの「DS コードポイント (DSCP)」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 DS コードポイントを読み取り、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じます。

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。ルーターは、DS コードポイントを使用して、並行するトラフィックフロー間の優先順位付けおよびスケジューリングを行い、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、ネットワークに送られる時に、可能なかぎり最良の扱いを受けることを保証します。EF 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。EF の技術情報については、RFC 2598 (*An Expedited Forwarding PHB*) を参照してください。

### 相対的優先転送 (*Assured Forwarding, AF*) PHB

相対的優先転送 (AF) では、4 つのクラスの転送動作をマーカーに指示できます。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。



たとえば、QoS ポリシーにより 2 つのトラフィッククラスに対してそれぞれ AF31 と AF13 の DSCP を割り当てるとします。AF31 (011010) の付いたパケットは、IPQoS システムからの送信時に、AF13 (001110) の付いたパケットよりも低い転送優先順位が与えられます。

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 DS コードポイント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 DS コードポイント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 DS コードポイントを、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 コードポイント表に関する技術情報については、RFC 2597 を参照してください。ルーター製造元の Cisco Systems 社および Juniper Networks 社は、それぞれ自社の Web サイトで AF PHB の設定に関する詳細な情報を提供しています。この情報を使用して、IPQoS システムおよびルーター用の AF PHB を定義できます。また、ルーター製造元のマニュアルには、自社製品での DS コードポイントの設定方法が記載されています。

## DS コードポイントをマーカーに提供する

DS コードポイントの長さは 6 ビットです。DS フィールドの長さは 1 バイトです。IPQoS 構成ファイルで DS コードポイントを定義すると、マーカーはパケットヘッダーの最初の 6 ビットに DS コードポイントをマークします。残りの 2 ビットは、使用されません。

DS コードポイントを定義するには、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_codepoint}
```

dscp\_map パラメータは、「DS コードポイント (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 に示したコードポイントを、10 進数値に変換する必要があります。

## VLAN デバイスでの dlcsmk マーカーの使用

dlcsmk 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、dlcsmk を使用できます。

dlcsmk 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、サービスクラス (CoS) マークを実装します。CoS マークは、商用スイッチで一般的に使われています。

次の表に示すサービスクラスマークを定義することにより、dlcsmk マーカーアクション内で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	バックグラウンド
2	スペア
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

dlcsmk の詳細は、dlcsmk (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

ここでは、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。

次に示す machine1 用 IPQoS 構成ファイルは、machine2 へのスイッチを介してトラフィックをマークするための簡単なソリューションを表します。

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 {
```

## 例 6-2 VLAN デバイスを持つシステムの IPQoS 構成ファイル (続き)

```
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。マーカーアクション mark4 は、dlcosmk に対し、myclass クラスのデータグラムに CoS 4 の VLAN マークを追加するよう指示します。ユーザー優先順位位置 4 は、2 つのマシン間のスイッチが、machine1 からの myclass トラフィックフローに対して制御された負荷転送を与えることを示します。

## 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ます。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によって得られたデータは、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 モデル内での flowacct の位置については、図 1-1 を参照してください。flowacct の詳細な技術情報については、flowacct (7ipp) のマニュアルページを参照してください。

フローカウンティングを有効にするには、`flowacct` に加えて、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設定に関する全般的な手順については、表 5-1 を参照してください。

## flowacct パラメータ

`flowacct` 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。フローアカウントテーブルは、表示できません。

フローレコードを測定してテーブルへ書き込むには、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 - `timer` と `timeout` には異なる値を指定できます。

---

- `max_limit` - 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

IPQoS 構成ファイル内での `flowacct` パラメータの使用例については、77 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。

## フローレコードテーブル

`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、次のパラメータによって特定されます。これらを、`flowacct` の 8 タプルと呼びます。

- 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

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` 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間隔は 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ト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、次の属性が含まれます。

表 6-4 flowacct レコードの属性

属性名	属性の内容	タイプ
<code>src -addr-address_type</code>	オリジネータの発信元アドレス。 <code>address_type</code> 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	基本 (Basic)
<code>dest_ addr_address_type</code>	パケットの着信先アドレス。 <code>address_type</code> 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	基本 (Basic)
<code>src- port</code>	フローの起点となる発信元ポート	基本 (Basic)
<code>dest-port</code>	フローの宛先となる着信先ポート番号	基本 (Basic)
<code>protocol</code>	フローのプロトコル番号	基本 (Basic)
<code>total-packets</code>	フロー内のパケット数	基本 (Basic)
<code>total-bytes</code>	フロー内のバイト数	基本 (Basic)
<code>action_name</code>	このフローを記録した flowacct アクションの名前	基本 (Basic)
<code>creation_time</code>	flowacct がそのフローのパケットを最初に認識した時間	拡張 (Extended) のみ
<code>last_seen</code>	そのフローのパケットを最後に認識した時間	拡張 (Extended) のみ
<code>diffserv-field</code>	フローの発信パケットヘッダー内の DS コードポイント	拡張 (Extended) のみ
<code>user</code>	アプリケーションから取得される UNIX UserID またはユーザー名	拡張 (Extended) のみ
<code>projid</code>	アプリケーションから取得されるプロジェクト ID	拡張 (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、拡張アカウント機能と連動して動作します。acctadm の技術情報については、acctadm(1M) のマニュアルページを参照してください。

flowacct は、フローを監視し、テーブルにフローレコードを記録します。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、アカウント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 type -f filename flow
```

acctadm -e	acctadm を -e オプションを指定して呼び出す。-e は、直後にタイプを指定することを示す
type	収集するタイプを指定する。file-type は、basic または extended に置き換える必要がある。各ファイルタイプの属性については、表 6-4 を参照
-f file_name	フローレコードを格納するファイル file_name を作成する
flow	acctadm を IPQoS 上で実行することを示す

## IPQoS 構成ファイル

この節では、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、別の名前で作成ファイルを作成するのが最善の方法です。IPQoS 構成の適用作業およびデバッグ作業については、第 4 章を参照してください。

次の例に、IPQoS 構成ファイルの構文を示します。この例では、次の表記上の規則に従います。

- computer-style type - 構成ファイル各部を説明する構文情報。このテキストは、入力しない
- **bold type** - IPQoS 構成ファイルに入力する必要のあるリテラルテキスト。たとえば、IPQoS 構成ファイルは、常に **fmt\_version** で始める必要がある

- *italics type* - 構成に関する記述的信息に置き換える可変テキスト。たとえば、*action\_name* または *module\_name* は、常に構成に関する情報で置き換える必要がある

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
```

以下では、IPQoS 構成ファイルの各主要部分について説明します。

## アクション文

action 文を使用して、97 ページの「IPQoS アーキテクチャと diffserv モデル」に示されたさまざまな 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、次のアクションを追加して、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
}
```

クラシファイアアクション文の次に、params 句または クラス句を記述します。

ほかのすべてのアクション文では、次の構文を使用します。

```
action {
  name action_name
  module module_name
  params_clause | ""
  cf_clauses
}
```

文	定義
<code>name <i>action_name</i></code>	アクションに名前を付ける
<code>module <i>module_name</i></code>	呼び出す IPQoS モジュールを特定する。モジュールは、表 6-5 に示したモジュールのどれかでなければならない
<code>params_clause</code>	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
<code>cf_clauses</code>	クラス句またはフィルタ句のゼロ以上のセット

## モジュール定義

モジュール定義は、アクション文のパラメータを処理するモジュールを示します。IPQoS 構成ファイルには、次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 DS コードポイント作成に使用するマーカー



表 6-5 IPQoS モジュール (続き)

モジュール名	定義
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

## クラス句

トラフィックのクラスごとに「クラス句」を定義します。

IPQoS 構成内の残りのクラスを定義するには、次の構文を使用します。

```
class {
    name class_name
    next_action next_action_name
}
```

特定のクラスに関する統計情報取得を有効にするには、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。

詳細については、111 ページの「アクション文」を参照してください。

クラスに関する統計を有効にしたいときは、`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。

## フィルタ句

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 クラシファイアで使用可能なセレクタについては、表 6-1 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には、「フィルタ句」を使用します。フィルタ句の構文を次に示します。

```
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アクション文に関する 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、アクション文の統計を有効にする必要があります。

---

## ipqosconf 構成ユーティリティ

IPQoS 構成ファイルを読んだり、UNIX カーネル内の IPQoS モジュールを構成したりするには、`ipqosconf` ユーティリティを使用します。`ipqosconf` では、次のアクションを実行できます。

- 構成ファイルを IPQoS カーネルモジュールに適用する (`ipqosconf -a filename`)
- カーネル内に現在常駐している IPQoS 構成ファイルを表示する (`ipqosconf -l`)
- マシンをリブートするたびに、現行の IPQoS 構成を読み取り、適用するようにする (`ipqosconf -c`)
- 現行の IPQoS カーネルモジュールをフラッシュする (`ipqosconf -f`)

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

## 用語集

---

<b>diffserv</b> モデル	IP ネットワークで差別化サービスを実装するための IETF (Internet Engineering Task Force) のアーキテクチャ標準。主なモジュールとして、クラシファイア、メーター、マーカー、スケジューラ、およびドロップがある。IPQoS では、クラシファイア、メーター、およびマーカーの各モジュールを実装する。diffserv モデルについては、RFC 2475 ( <i>An Architecture for Differentiated Services</i> ) に解説されている。
<b>DS</b> コードポイント ( <b>DSCP</b> )	IP ヘッダーの DS フィールドに含まれていて、パケットの転送方法を指示する 6 ビットの値。
<b>IPQoS</b>	Solaris 9/9/02 のソフトウェア機能。diffserv 標準に加えて、フローカウンティングおよび仮想 LAN 用の 802.1D マーキングも実装する。IPQoS を使用すると、IPQoS 構成ファイル内に定義したとおりに、さまざまなレベルのネットワークサービスを顧客やアプリケーションに提供できる。
仮想 LAN ( <b>VLAN</b> ) デバイス	IP プロトコルスタックの Ethernet (データリンク) レベルでトラフィック転送を行うネットワークインタフェース。
クラス	似たような特性を共有するネットワークフローのグループ。クラスは、IPQoS 構成ファイル内に定義する。
結果 ( <b>outcome</b> )	トラフィックの計測結果に基づいて実行されるアクション。IPQoS メーターには、赤、黄、および緑の 3 種類の結果 ( <b>outcome</b> ) があり、IPQoS 構成ファイル内に定義される。
セレクタ	ネットワークストリームからトラフィックを選択するために、特定クラスの packets に適用される条件を具体的に定義する要素。セレクタは、IPQoS 構成ファイル内のフィルタ句に定義する。
フィルタ	クラスの特性を IPQoS 構成ファイル内に定義するための規則セット。IPQoS システムでは、IPQoS 構成ファイル内に定義されたフィルタに適合するトラフィックフローを選択して処理する。

フローカウンティング	トラフィックフローに関する情報を蓄積して記録するプロセス。フローカウンティングを確立するには、flowacct モジュールのパラメータを IPQoS 構成ファイル内に定義する。
ホップ単位動作 ( <b>Per-Hop Behavior</b> 、PHB)	トラフィッククラスに割り当てられる優先順位。PHB 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を示す。
マーカー	<p>1. diffserv アーキテクチャおよび IPQoS のモジュールの 1 つ。パケットの転送方法を指示する値を IP パケットの DS フィールドに付ける。IPQoS 実装では、このマーカーモジュールは dscpmk。</p> <p>2. IPQoS 実装のモジュールの 1 つ。ユーザー優先順位の値を Ethernet データグラムの仮想 LAN タグに付ける。ユーザー優先順位の値は、VLAN デバイスを備えたネットワーク上でデータグラムが転送される方法を示す。このモジュールは dlcosmk と呼ばれる。</p>
メーター	特定クラスのトラフィックフローの速度を測定する diffserv アーキテクチャのモジュール。IPQoS 実装には、tokenmt および tswtclmt という 2 つのメーターがある。
ユーザー優先順位	サービスクラスのマークを実装する 3 ビットの値。VLAN デバイスのネットワーク上で Ethernet データグラムが転送される方法を定義する。

# 索引

---

## A

acctadm コマンド, フローアカウンティングでの使用, 25, 92, 94, 110

## C

CoS (サービスクラス) マーク, 25

## D

diffserv 対応ルーター  
DS コードポイントの評価, 105  
計画, 38  
構成, 81

diffserv モデル  
IPQoS での実装, 22, 24, 25, 26  
クラシファイアモジュール, 23  
フローの例, 26  
マーカーモジュール, 25  
メーターモジュール, 24

dlcosmk マーカー, 25  
VLAN タグ, 106  
データグラム転送の計画, 46  
ユーザー優先順位値の表, 106

dscpmk マーカー, 25  
パケット転送での PHB, 103  
パケット転送の計画, 46  
呼び出し, マーカーアクションでの, 63, 69, 75, 79

DS コードポイント (DSCP), 25, 27  
AF コードポイント, 29, 104

DS コードポイント (DSCP) (続き)

dscp\_map パラメータ, 105  
EF コードポイント, 28, 104  
PHB および DSCP, 28  
カラーウェア構成での, 102  
計画, QoS ポリシーでの, 46  
構成, diffserv ルーター上での, 81, 104  
定義, IPQoS 構成ファイルでの, 63

## F

flowacct モジュール, 25, 107  
acctadm コマンド, フローアカウンティング  
ファイルを作成する, 110  
flowacct のアクション文, 66  
パラメータ, 108  
フローレコード, 92  
フローレコードテーブル, 108  
フローレコードの属性, 109

## I

ipgpc クラシファイア  
クラシファイアモジュールを参照

IPQoS, 17  
diffserv モデルの実装, 22  
ipgosconf ユーティリティ, 84  
IPQoS ネットワーク上のルーター, 81  
QoS ポリシーの計画, 36  
VLAN デバイスのサポート, 106  
エラーメッセージ, 87

## IPQoS (続き)

- 関連する RFC, 18
- 機能, 18
- 構成計画, 31
- 構成ファイル, 55
- 構成ファイルの構文, 110
- サポートするネットワークトポロジ, 32, 33, 34, 35
- 統計情報の生成, 95
- トラフィック管理機能, 21, 22
- ネットワーク例, 49, 55
- マニュアルページ, 19
- メッセージのログ記録, 85

## ipqosconf, 54

- 現行の構成の表示, 85
- 構成の適用, 84, 85
- コマンドオプション, 114

## K

kstat コマンド, IPQoS での使用, 95

## P

### params 句

- flowacct アクションでの使用, 66
- グローバル統計の定義, 58, 114
- 構文, 114
- マーカーアクションでの使用, 63
- メータリングアクションでの使用, 78

## Q

### QoS ポリシー, 20

- IPQoS 構成ファイルでの実装, 53
- クラスの作成, 39
- 計画の作業マップ, 37
- トラフィック転送の計画, 46
- フィルタの作成, 41
- フローアカウンティングの計画, 48
- フロー制御の計画, 43
- ポリシー構成テンプレート, 36

## R

RFC (Requests for Comments), IPQoS の, 18

## S

syslog.conf ファイルのログ記録, IPQoS の, 85

## T

### tokenmt メーター, 24

- カラーアウェアとして構成, 25, 101
- シングルレートメーターとして構成, 101
- 速度の計測, 100
- 速度パラメータ, 101
- ツーレートメーターとして構成, 101

### tswtc1mt メーター, 24, 102

- 速度の計測, 102

## W

### Web サーバー

- IPQoS 用の構成, 55, 57, 58, 59, 60, 62, 64, 65, 67, 68, 70

## あ

### アクション文, 111

アプリケーションサーバー, IPQoS 用の構成, 70

## え

エラーメッセージ, IPQoS の, 87, 90

## か

仮想 LAN (VLAN) デバイス, IPQoS ネットワーク上の, 106

カラーアウェアネス, 25, 101

完全優先転送 (EF), 28, 104

定義, IPQoS 構成ファイルでの, 64

## く

- クラシファイアモジュール, 23
  - アクション文, 58
  - クラシファイアの機能, 98
- クラス, 23
  - クラス句の構文, 113
  - 計画, QoS ポリシーでの, 39
  - セレクタの表, 98
  - 定義, IPQoS 構成ファイルでの, 58, 67, 72
  - クラス句, IPQoS 構成ファイルの, 59, 113

## こ

- 構成ファイル, IPQoS の, 55
  - IPQoS モジュールの表, 112
  - アクション文の構文, 112
  - クラス句, 59
  - 構文, 111
  - 初期アクション文, 58, 111
  - フィルタ句, 61
  - マーカーアクション文, 63
- 構成ファイルの例, IPQoS の
  - VLAN デバイス構成, 106
  - アプリケーションサーバー, 70
  - カラーアウェアのセグメント, 102
  - プレミアム Web サーバー, 55
  - ベストエフォート Web サーバー, 57
- 構成例, IPQoS の, 49

## さ

- サービスクラス
  - クラスを参照
- サービス品質 (QoS)
  - QoS ポリシー, 20
  - 作業, 17
- サービスレベル契約 (SLA), 20
  - 顧客への課金, フローアカウンティングに基づく, 92
  - サービスクラス, 23
  - さまざまなサービスクラスの提供, 22
  - 実装, QoS ポリシーでの, 39
- 作業マップ, IPQoS の
  - IPQoS の構成と保守, 83
  - QoS ポリシーの計画, 37
  - 構成計画, 31

- 作業マップ, IPQoS の (続き)
  - 構成ファイルの作成, 53
  - フローアカウンティングの設定, 91
- 差別化サービス, 17
  - 差別化サービスモデル, 22
  - さまざまなサービスクラスの提供, 22
  - ネットワークポロジ, 32

## せ

- セレクタ, 24
  - IPQoS 5 タプル, 23
  - 計画, QoS ポリシーでの, 41
  - セレクタの表, 98

## そ

- 相対的優先転送 (AF), 29, 104
  - AF コードポイントの表, 104
  - マーカーアクションでの使用, 63

## た

- 帯域幅の調整, 21
  - 計画, QoS ポリシーでの, 40

## と

- 統計情報, IPQoS の
  - クラスの統計取得の有効化, 113
  - グローバル統計取得の有効化, 59, 113
  - 生成, kstat コマンドによる, 95
- トラフィック管理
  - 帯域幅の調整, 21
  - トラフィック転送, 27, 28, 29
  - トラフィックフローの優先順位付け, 22
  - ネットワークポロジの計画, 33
  - フロー制御, 24
- トラフィック適合
  - 結果 (outcome), 24, 100
  - 結果の計画, QoS ポリシーでの, 44
  - 結果の定義, 78
  - 速度の計画, QoS ポリシーでの, 44
  - 速度の定義, 78

## トラフィック適合 (続き)

速度パラメータ, 100, 101

## 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, 29

IP パケットの転送, DSCP を使用した, 27

アプリケーショントラフィックの転送, 75

計画, QoS ポリシーでの, 40, 46

実装, IPQoS 構成ファイルでの, 62

データグラムの転送, 106

パケット転送での PHB の影響, 103

## ね

ネットワークトポロジ, IPQoS の, 32

IPQoS 対応サーバーファームを備えた

LAN, 33

IPQoS 対応のファイアウォールを備えた

LAN, 35

IPQoS 対応ホストを備えた LAN, 33

構成例, 49

トポロジの準備, 38

ネットワーク例, IPQoS の, 55

## は

ハードウェア, IPQoS 対応ネットワークの, 32

## ふ

フィルタ, 24

計画, QoS ポリシーでの, 41

作成, IPQoS 構成ファイルでの, 60, 68, 73

セレクタの表, 98

フィルタ句の構文, 113

フィルタ句, IPQoS 構成ファイルの, 61, 113

フローアカウンティング, 92, 107

計画, QoS ポリシーでの, 48

実装, IPQoS 構成ファイルでの, 65

フローアカウンティングファイルの作成, 92

フローレコードテーブル, 108

## フロー制御

計画, QoS ポリシーでの, 43

定義, IPQoS 構成ファイルでの, 77

メータリングモジュールによる, 24

## ほ

ホップ単位動作 (PHB), 28

AF 転送, 29

EF 転送, 28

使用, dscpmk マーカーでの, 103

定義, IPQoS 構成ファイルでの, 62, 80

## ま

マーカーモジュール, 25

dicosmk マーカーも参照

dscpmk マーカーも参照

DS コードポイントの指定, 105

IP 転送の計画, QoS ポリシーでの, 46

PHB, IP パケット転送での, 28

VLAN デバイスのサポート, 106

マニュアルページ, IPQoS の, 19

## め

メータリングモジュール, 24

tokenmt メーターも参照

tswtclmt メーターも参照

計画, QoS ポリシーでの, 43

メータリングの結果, 24, 100

呼び出し, IPQoS 構成ファイルでの, 78

## ゆ

ユーザー優先順位の値, 25

## る

ルーター, IPQoS ネットワーク上の

diffserv 対応ルーターを参照