

## 在 Oracle® Solaris 11.1 中管理 IP 服务质量

版权所有 © 1999, 2012, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

#### U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

# 目录

---

前言 .....	7
<b>1 IPQoS 介绍 (概述) .....</b>	<b>9</b>
IPQoS 基本知识 .....	9
何为区分服务? .....	9
IPQoS 功能 .....	10
何处获取有关服务质量的理论和实践的更多信息 .....	10
使用 IPQoS 提供服务质量 .....	12
实现服务级别协议 .....	12
保证单个组织的服务质量 .....	12
服务质量策略介绍 .....	12
使用 IPQoS 提高网络效率 .....	13
带宽如何影响网络通信 .....	13
使用服务类设置通信的优先级 .....	13
区分服务模型 .....	14
分类器 (ipgpc) 概述 .....	14
计量器 (tokenmt 和 tswtclmt) 概述 .....	15
标记器 (dscpmk 和 dlcosmk) 概述 .....	16
流记帐 (flowacct) 概述 .....	16
通信如何流过 IPQoS 模块 .....	17
启用了 IPQoS 的网络上的通信转发 .....	18
DS 代码点 .....	18
单跳行为 .....	18
<b>2 规划启用了 IPQoS 的网络 (任务) .....</b>	<b>23</b>
常规 IPQoS 配置规划 (任务列表) .....	23
规划 Diffserv 网络拓扑 .....	24

Diffserv 网络的硬件策略 .....	24
IPQoS 网络拓扑 .....	24
规划服务质量策略 .....	27
QoS 策略规划帮助 .....	27
QoS 策略规划（任务列表） .....	27
▼ 如何为 IPQoS 准备网络 .....	28
▼ 如何定义 QoS 策略类 .....	29
定义过滤器 .....	30
▼ 如何在 QoS 策略中定义过滤器 .....	31
▼ 如何规划流控制 .....	32
▼ 如何规划转发行为 .....	34
▼ 如何规划流记帐 .....	36
IPQoS 配置示例介绍 .....	37
IPQoS 拓扑 .....	37
<b>3 创建 IPQoS 配置文件（任务） .....</b>	<b>41</b>
在 IPQoS 配置文件中定义 QoS 策略（任务列表） .....	41
创建 QoS 策略所用的工具 .....	42
基本的 IPQoS 配置文件 .....	42
为 Web 服务器创建 IPQoS 配置文件 .....	43
▼ 如何创建 IPQoS 配置文件并定义通信类 .....	45
▼ 如何在 IPQoS 配置文件中定义过滤器 .....	47
▼ 如何在 IPQoS 配置文件中定义通信转发 .....	48
▼ 如何在 IPQoS 配置文件中为类启用记帐 .....	51
▼ 如何为尽力服务 Web 服务器创建 IPQoS 配置文件 .....	52
为应用服务器创建 IPQoS 配置文件 .....	54
▼ 如何为应用服务器配置 IPQoS 配置文件 .....	56
▼ 如何在 IPQoS 配置文件中为应用程序通信配置转发 .....	58
▼ 如何在 IPQoS 配置文件中配置流控制 .....	60
在路由器上提供区分服务 .....	63
▼ 如何在启用了 IPQoS 的网络中配置路由器 .....	63
<b>4 启动和维护 IPQoS（任务） .....</b>	<b>65</b>
管理 IPQoS（任务列表） .....	65
应用 IPQoS 配置 .....	66

---

▼ 如何将新配置应用于 IPQoS 内核模块 .....	66
▼ 如何确保每次重新引导系统之后都应用 IPQoS 配置 .....	67
启用 IPQoS 消息的 syslog 日志 .....	67
▼ 如何在引导过程中启用 IPQoS 消息的日志记录 .....	67
对 IPQoS 错误消息进行故障排除 .....	68
<b>5 使用流记帐和统计信息收集功能（任务） .....</b>	<b>73</b>
设置流记帐（任务列表） .....	73
记录有关通信流量的信息 .....	74
▼ 如何为流记帐数据创建文件 .....	74
收集统计信息 .....	76
<b>6 IPQoS 的详细介绍（参考信息） .....</b>	<b>77</b>
IPQoS 体系结构和 Diffserv 模型 .....	77
分类器模块 .....	78
计量器模块 .....	79
标记器模块 .....	82
flowacct 模块 .....	85
IPQoS 配置文件 .....	88
action 语句 .....	89
模块定义 .....	90
class 子句 .....	90
filter 子句 .....	91
params 子句 .....	91
ipqosconf 配置实用程序 .....	91
索引 .....	93



# 前言

---

欢迎阅读《在 Oracle Solaris 11.1 中管理 IP 服务质量》。本书是多卷集的一部分，该卷集包含 Oracle Solaris 系统管理信息中的重要部分。本书假定您已经安装 Oracle Solaris。您应该已经可以配置网络，或者已经可以配置网络上所需的任何网络软件。

---

注 - 此 Oracle Solaris 发行版支持使用 SPARC 和 x86 系列处理器体系结构的系统。支持的系统可以在 [Oracle Solaris OS: Hardware Compatibility Lists](#) (Oracle Solaris OS: 硬件兼容性列表) 中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

---

## 目标读者

本书适用于所有负责管理在网络中配置的、运行 Oracle Solaris 的系统的人员。要使用本书，您应当至少具备两年的 UNIX 系统管理经验。参加 UNIX 系统管理培训课程可能会对您有所帮助。

## 获取 Oracle 支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> (如果您听力受损)。

## 印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>

表 P-1 印刷约定 (续)

字体或符号	含义	示例
<b>AaBbCc123</b>	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name% su</code> <code>Password:</code>
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 <b>注意：</b> 有些强调的项目在联机时以粗体显示。
<b>新词术语强调</b>	新词或术语以及要强调的词	<b>高速缓存</b> 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

## 命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#
C shell	machine_name%
C shell 超级用户	machine_name#



# IPQoS 介绍 ( 概述 )

---

通过 IP 服务质量 (IP Quality of Service, IPQoS)，您可以控制、收集记帐统计信息并设置其优先级。还可以借助 IPQoS 为网络上的用户提供始终如一的服务水平。同时也可以管理通信来避免网络拥塞。

本章包含以下主题：

- 第 9 页中的“IPQoS 基本知识”
- 第 12 页中的“使用 IPQoS 提供服务质量”
- 第 13 页中的“使用 IPQoS 提高网络效率”
- 第 14 页中的“区分服务模型”
- 第 18 页中的“启用了 IPQoS 的网络上的通信转发”

---

注 - 未来发行版中可能会删除 IPQoS 功能。鼓励用户转而使用 `dladm`、`flowadm` 和相关命令，这些命令可支持类似的带宽资源控制功能。有关更多信息，请参见《[在 Oracle Solaris 11.1 中使用虚拟网络](#)》。

---

## IPQoS 基本知识

IPQoS 实现了区分服务 (Differentiated Service, Diffserv) 体系结构，此体系结构由 Internet 工程任务组 (Internet Engineering Task Force, IETF) 的区分服务工作组定义。在 Oracle Solaris OS 中，IPQoS 在 TCP/IP 协议栈的 IP 级别上实现。

### 何为区分服务？

通过启用 IPQoS，您可以为选定的客户和应用程序提供不同级别的网络服务。这些不同级别的服务统称为**区分服务**。您可以根据自己公司为客户提供的服务级别结构来向客户提供区分服务。还可以基于为网络上的应用程序或用户设置的优先级来提供区分服务。

提供服务质量涉及以下活动：

- 为不同的组（例如客户或企业中的部门）提供不同的服务级别
- 为提供给特定组或应用程序的网络服务设置优先级
- 发现和消除网络瓶颈区域以及其他形式的拥塞
- 监视网络性能并提供性能统计信息
- 控制进出网络资源的带宽

## IPQoS 功能

IPQoS 具有以下功能：

- 用于配置 QoS 策略的 `ipqosconf` 命令行工具
- 用于选择操作的分类器，这些操作基于用来配置组织 QoS 策略的过滤器
- 根据 Diffserv 模型度量网络通信的计量模块
- 服务区分功能，这种功能基于使用转发信息标记 IP 数据包头的的能力
- 收集通信流统计信息的流记帐模块
- 通过 UNIX® `kstat` 命令对通信类收集统计信息
- 支持 SPARC® 和 x86 体系结构
- 支持 IPv4 和 IPv6 寻址
- 与 IP 安全体系结构 (IPsec) 的互操作性
- 支持虚拟局域网 (virtual local area network, VLAN) 的 802.1D 用户优先级标记

## 何处获取有关服务质量的理论和实践的更多信息

您可以从印刷资料和联机资料中找到有关区分服务和服务质量的信息。

### 有关服务质量的书籍

有关服务质量的理论和实践的更多信息，请参阅以下书籍：

- 由 Ferguson, Paul 和 Geoff Huston 合著的《Quality of Service》。John Wiley & Sons, Inc. 出版，1998。
- 由 Kilkki, Kalevi 编著的《Differentiated Services for the Internet》。Macmillan Technical Publishing 出版，1999。

## 有关服务质量的请求注解文档 (Requests for Comments, RFC)

IPQoS 遵循以下 RFC 和 Internet 草案中所述的规范：

- [RFC 2474, Definition of the Differentiated Services Field \(DS Field\) in the IPv4 and IPv6 Headers](http://www.ietf.org/rfc/rfc2474.txt?number=2474) (RFC 2474, IPv4 和 IPv6 包头中区分服务字段 (DS 字段) 的定义) — 介绍支持区分服务的 IPv4 和 IPv6 包头的服务类型 (type of service, ToS) 字段或 DS 字段的增强功能。
- [RFC 2475, An Architecture for Differentiated Services](http://www.ietf.org/rfc/rfc2475.txt?number=2475) (RFC 2475, 区分服务的体系结构) — 提供 Diffserv 体系结构的组织和模块的详细说明。
- [RFC 2597, Assured Forwarding PHB Group](http://www.ietf.org/rfc/rfc2597.txt?number=2597) (RFC 2597, 保证转发 PHB 组) — 说明保证转发 (assured forwarding, AF) 单跳行为的工作原理。
- [RFC 2598, An Expedited Forwarding PHB](http://www.ietf.org/rfc/rfc2598.txt?number=2598) (RFC 2598, 加速转发 PHB) — 说明加速转发 (expedited forwarding, EF) 单跳行为的工作原理。
- Internet 草案 《An Informal Management Model for Diffserv Routers》 — 介绍在路由器上实现 Diffserv 体系结构的模型。

## 提供服务质量信息的 Web 站点

IETF 的区分服务工作组维护一个 Web 站点，该站点位于 <http://www.ietf.org/html.charters/diffserv-charter.html>，其中包含指向 Diffserv Internet 草案的链接。

路由器制造商（例如 Cisco Systems 和 Juniper Networks）的公司 Web 站点，这些站点介绍了如何在其产品中实现区分服务的信息。

## IPQoS 手册页

IPQoS 文档包含以下手册页：

- [ipqosconf\(1M\)](#) — 介绍用于设置 IPQoS 配置文件的命令
- [ipqos\(7ipp\)](#) — 介绍 Diffserv 体系结构模型的 IPQoS 实现
- [ipgpc\(7ipp\)](#) — 介绍 Diffserv 分类器的 IPQoS 实现
- [tokenmt\(7ipp\)](#) — 介绍 IPQoS tokenmt 计量器
- [tswtclmt\(7ipp\)](#) — 介绍 IPQoS tswtclmt 计量器
- [dscpmk\(7ipp\)](#) — 介绍 DSCP 标记器模块
- [dlcosmk\(7ipp\)](#) — 介绍 IPQoS 802.1D 用户优先级标记器模块
- [flowacct\(7ipp\)](#) — 介绍 IPQoS 流记帐模块
- [acctadm\(1M\)](#) — 介绍配置 Oracle Solaris 扩展记帐功能的命令。acctadm 命令包含 IPQoS 扩展。

# 使用 IPQoS 提供服务质量

IPQoS 功能使得 Internet 服务提供商 (Internet service provider, ISP) 和应用程序服务提供商 (application service provider, ASP) 能够为客户提供不同级别的网络服务。利用这些功能，公司和教育机构可以为内部组织或主要应用程序设置服务优先级。

## 实现服务级别协议

如果您的组织是一个 ISP 或 ASP，则可以基于贵公司提供给客户的服务级别协议 (service-level agreement, SLA) 来配置 IPQoS。在 SLA 中，服务提供商保证客户能够获得基于价格结构的特定网络服务级别。例如，高价的 SLA 可以确保客户每天 24 小时都能获得最高优先级的所有类型网络通信。相反，中等价位的 SLA 只能保证客户在上班时间针对电子邮件的收发获得较高的优先级。其他所有通信在一天 24 小时内应具有中等优先级。

## 保证单个组织的服务质量

如果贵组织是一个企业或机构，则还可以为您的网络提供服务质量功能。您可以保证特定组或特定应用程序的通信能够获得较高或较低程度的服务。

## 服务质量策略介绍

您可以通过定义服务质量 (*quality-of-service*, QoS) 策略来实现服务质量。QoS 策略定义各种网络属性，例如客户或应用程序的优先级以及处理不同通信类别的操作。可以在 IPQoS 配置文件中实现组织的 QoS 策略。此文件可以配置驻留在 Oracle Solaris 内核中的 IPQoS 模块。应用 IPQoS 策略的主机被视为启用了 IPQoS 的系统。

QoS 策略通常定义以下内容：

- 称为**服务类**的独立网络通信组。
- 用于控制各类网络通信的度量标准。这些度量标准管理称为**计量的通信度量过程**。
- IPQoS 系统和 Diffserv 路由器必须应用于包流的操作。此类操作称为**单跳行为** (per-hop behavior, PHB)。
- 组织需要针对服务类收集的任何统计信息。例如由客户或特定应用程序产生的通信。

当包传送到网络时，启用了 IPQoS 的系统将对包头进行评估。IPQoS 系统执行的操作由 QoS 策略确定。

设计 QoS 策略的任务在第 27 页中的“规划服务质量策略”中进行介绍。

## 使用 IPQoS 提高网络效率

IPQoS 包含的功能可帮助您在实现服务质量的同时提高网络性能。在扩展计算机网络的同时，需要加强管理因用户和功能更强大的处理器的数量增加而产生的网络通信。过度使用网络所产生的一些症状包括丢失数据和通信拥塞。这两种症状都将导致响应速度降低。

过去，系统管理员通过增加带宽来处理网络通信问题。通常，链路上通信级别变化很大。使用 IPQoS，您可以管理现有网络上的通信并帮助评估哪些位置需要进行扩展以及是否必须进行此扩展。

例如，对于一个企业或机构，必须保持高效的网络以避免出现通信瓶颈。还必须确保某个组或应用程序占用的带宽不能多于为它分配的带宽。对于 ISP 或 ASP，您必须管理网络性能以确保客户获得他们付款级别的网络服务。

## 带宽如何影响网络通信

您可以使用 IPQoS 控制网络带宽，即充分利用的网络链路或设备可传输的最大数据量。QoS 策略应设置带宽的使用优先级，以便为客户或用户提供服务质量。使用 IPQoS 计量模块，您可以度量并控制启用了 IPQoS 的主机上各种通信类之间的带宽分配。

在有效管理网络上的通信之前，您必须回答以下有关带宽使用的问题：

- 您的本地网络的通信问题是什么？
- 必须执行哪些操作才能优化可用带宽的使用？
- 哪些是站点关键的应用程序，必须为哪些应用程序提供最高优先级？
- 哪些应用程序对拥塞敏感？
- 哪些是不太关键的应用程序，可以为哪些应用程序提供较低的优先级？

## 使用服务类设置通信的优先级

要实现服务质量，需要分析网络通信以确定通信大致可分为哪几个分组。然后，将各个分组归入具有单独特征和单独优先级的服务类中。这些类构成了设置组织 QoS 策略所依据的基本类别。服务类表示想要控制的通信组。

例如，提供商可以按变动价格结构提供白金级、黄金级、白银级和青铜级服务。白金 SLA 可以保证以 ISP 为客户托管的 Web 站点为目标的传入通信获得最高优先级。因此，客户 Web 站点的传入通信可以是一个通信类。

对于企业来说，您可以创建基于部门需求的服务类。或者，可以创建基于网络通信中特定应用程序的优势的类。

以下是某企业的一些通信类的示例：

- 常见应用程序，例如发送到特定服务器的电子邮件和传出 FTP，其中任何一项都可以组成一个类。由于员工会经常使用这些应用程序，因此，QoS 策略可以保证电子邮件和传出 FTP 具有少量的带宽和较低的优先级。
- 需要一天 24 小时运行的订单输入数据库。可以为数据库提供大量带宽和较高的优先级，具体视数据库应用程序对企业的重要性而定。
- 执行重要工作或敏感工作的部门，例如劳资部门。部门对企业的重要性将决定为其提供的优先级和带宽。
- 对公司外部 Web 站点的外来调用。您可以为该提供以低优先级运行的适量带宽。

## 区分服务模型

IPQoS 包含以下模块，这些模块属于 RFC 2475 中定义的**区分服务** (Differentiated Service, Diffserv) 体系结构的一部分：

- 分类器
- 计量器
- 标记器

IPQoS 在 Diffserv 模型中添加了以下增强功能：

- 流记帐模块
- 802.1D 数据报标记器

本节介绍由 IPQoS 使用的 Diffserv 模块。要设置 QoS 策略，您需要了解这些模块及其名称和用法。有关各个模块的详细信息，请参阅第 77 页中的“[IPQoS 体系结构和 Diffserv 模型](#)”。

## 分类器 (ipgpc) 概述

在 Diffserv 模型中，**分类器**从网络通信流中选择包。**通信流**由一组在以下 IP 数据包头字段中具有相同信息的包组成：

- 源地址
- 目标地址
- 源端口
- 目标端口
- 协议编号

在 IPQoS 中，这些字段称为 **5 元组**。

IPQoS 分类器模块名为 `ipgpc`。`ipgpc` 分类器将通信流归到各个类中，这些类基于在 IPQoS 配置文件中配置的特征划分。

有关 `ipgpc` 的详细信息，请参阅第 78 页中的“分类器模块”。

## IPQoS 类

类是一组具有类似特征的网络流。例如，ISP 可以将类定义为代表为客户提供的不同服务级别。ASP 可以将提供不同级别服务的 SLA 定义为各种应用程序。对于 ASP 的 QoS 策略，类可能包含送达特殊目标 IP 地址的传出 FTP 通信。还可以将公司外部 Web 站点的传出通信定义为一个类。

将通信分类是规划 QoS 策略的重要组成部分。使用 `ipqosconf` 实用程序创建类的过程实际上就是配置 `ipgpc` 分类器的过程。

有关如何定义类的信息，请参见第 29 页中的“如何定义 QoS 策略类”。

## IPQoS 过滤器

过滤器是包含称为**选定器**参数的规则集合。每个过滤器都必须指向一个类。IPQoS 根据每个过滤器的选定器来匹配包，从而确定此包是否属于该过滤器的类。您可以使用各种选定器过滤包，例如，IPQoS 5 元组和其他常用参数：

- 源地址和目标地址
- 源端口和目标端口
- 协议编号
- 用户 ID
- 项目 ID
- 区分服务代码点 (Differentiated Services Codepoint, DSCP)
- 接口索引

例如，简单过滤器可能包含值为 80 的目标端口。于是 `ipgpc` 分类器将选择所有送达目标端口 80 (HTTP) 的包并按照 QoS 策略中的指示处理这些包。

有关创建过滤器的信息，请参见第 31 页中的“如何在 QoS 策略中定义过滤器”。

## 计量器 ( `tokenmt` 和 `tswtclmt` ) 概述

在 Diffserv 模型中，**计量器**按类跟踪通信流的传输速率。计量器通过评估流的实际速率与配置的速率的符合程度来确定相应的结果。计量器会根据通信流的结果来选择后续操作。后续操作可能包括将包发送到其他操作或将包返回到网络而不进行进一步处理。

IPQoS 计量器确定网络流是否符合在 QoS 策略中为其所属的类定义的传输速率。IPQoS 包括两种计量模块：

- `tokenmt`—使用双令牌桶计量方案
- `tswtclmt`—使用时间滑动窗口计量方案

这两种计量模块可识别三种结果：红色、黄色和绿色。在参数 `red_action_name`、`yellow_action_name` 和 `green_action_name` 中定义针对每种结果所采取的操作。

此外，还可以将 `tokenmt` 配置为可识别颜色。可识别颜色的计量实例使用包大小、DSCP、通信率和配置的参数来确定结果。计量器使用 DSCP 将包的结果映射到绿色、黄色或红色。

有关为 IPQoS 计量器定义参数的信息，请参阅第 32 页中的“如何规划流控制”。

## 标记器（`dscpmk` 和 `dlcosmk`）概述

在 Diffserv 模块中，**标记器** 使用反映转发行为的值标记包。**标记**是将一个值放入包头以指示如何将此包转发到网络的过程。

IPQoS 包含两种标记器模块：

- `dscpmk`—使用称为**区分服务代码点**或 DSCP (Differentiated Services codepoint) 的数值来标记 IP 数据包头中的 DS 字段。可识别 Diffserv 的路由器便可以使用 DS 代码点对包应用相应的转发行为。
- `dlcosmk`—使用称为**用户优先级**的数值来标记以太网帧标题的虚拟局域网 (virtual local area network, VLAN) 标记。用户优先级指示**服务类** (class of service, CoS)，该类定义应用于数据报的相应转发行为。

按照 IETF 的设计，`dlcosmk` 是一个 IPQoS 添加操作，而不是 Diffserv 模型的一部分。

有关针对 QoS 策略实现标记器策略的信息，请参见第 34 页中的“如何规划转发行为”。

## 流记帐 (`flowacct`) 概述

IPQoS 将 `flowacct` 记帐模块添加到 Diffserv 模型中。您可以使用 `flowacct` 来收集有关通信流的统计信息，并根据客户的 SLA 为客户开帐单。流记帐对容量规划和系统监视也非常有用。

`flowacct` 模块可与 `acctadm` 命令一起使用来创建记帐日志文件。基本日志包括 IPQoS 5 元组和两个其他属性，如下所示：

- 源地址
- 源端口
- 目标地址
- 目标端口
- 协议编号
- 包编号
- 字节数



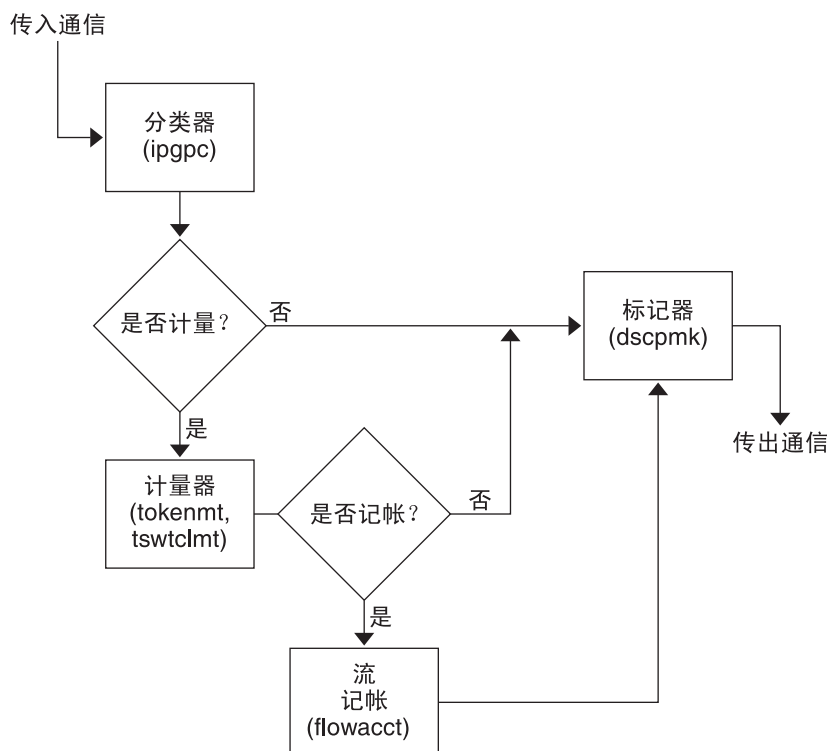
您还可以收集有关其他属性的统计信息，如第 74 页中的“记录有关通信流量的信息”，以及 flowacct(7ipp) 和 acctadm(1M) 手册页中所述。

有关规划流记帐策略的信息，请参见第 36 页中的“如何规划流记帐”。

## 通信如何流过 IPQoS 模块

下图显示了传入通信经过某些 IPQoS 模块时可能采用的路径。

图 1-1 通信流经过 Diffserv 模型的 IPQoS 实现



此图展示了启用 IPQoS 的计算机上常见的通信流顺序：

1. 分类器从包流中选择所有符合系统 QoS 策略中过滤条件的包。
2. 然后针对下一步要采取的操作评估选定的包。
3. 分类器将所有不需要流控制的通信发送到标记器。
4. 将流控制的通信发送到计量器。
5. 计量器强制执行已配置的速率。然后，计量器为流控制的包指定一个通信符合值。

6. 然后评估流控制的包以确定是否所有包都需要记帐。
7. 计量器将所有不需要流记帐的通信发送到标记器。
8. 流记帐模块收集有关已接收包的统计信息。然后，此模块将包传送到标记器。
9. 标记器为包头指定一个 DS 代码点。此 DSCP 指示可识别 Diffserv 的系统必须应用于包的单跳行为。

## 启用了 IPQoS 的网络上的通信转发

本节介绍在启用了 IPQoS 的网络上转发包所涉及的因素。启用了 IPQoS 的系统会处理网络流上将系统 IP 地址作为目标地址的所有包。之后，此 IPQoS 系统会对包应用其 QoS 策略以建立区分服务。

### DS 代码点

DS 代码点 (DS codepoint, DSCP) 在包头中定义任何可识别 Diffserv 的系统应针对已标记包采取的操作。Diffserv 体系结构为所要使用的启用了 IPQoS 的系统和 Diffserv 路由器定义了一组 DS 代码点。Diffserv 体系结构还定义了一组称为**转发行为**的操作，这些操作与 DSCP 相对应。启用了 IPQoS 的系统使用 DSCP 标记包头中 DS 字段的优先位。当路由器收到带有 DSCP 值的包时，便会应用与此 DSCP 关联的转发行为。然后，会将此包释放到网络。

---

注 - dLcosmk 标记器不使用 DSCP。dLcosmk 而是使用 CoS 值来标记以太网帧标题。如果计划在使用 VLAN 设备的网络上配置 IPQoS，请参阅第 82 页中的“标记器模块”。

---

### 单跳行为

在 Diffserv 术语中，指定给 DSCP 的转发行为称为**单跳行为** (per-hop behavior, PHB)。PHB 定义已标记的包获得的相对于可识别 Diffserv 的系统上其他通信的转发优先级。此优先级最终确定启用了 IPQoS 的系统或 Diffserv 路由器是转发还是丢弃已标记的包。对于转发的包，此包在到达目的地的途中遇到的每个 Diffserv 路由器都应用相同的 PHB。但也有例外的情况，那就是其他 Diffserv 系统更改了 DSCP。有关 PHB 的更多信息，请参阅第 82 页中的“使用 dscpmk 标记器转发包”。

PHB 的目标是为邻近网络上的通信类提供指定量的网络资源。您可以通过 QoS 策略达成此目标。定义 DSCP，以指示当通信流离开启用了 IPQoS 的系统时通信类的优先级。优先级的范围可以从高优先级/低丢弃率到低优先级/高丢弃率。

例如，您的 QoS 策略可以为一个通信类指定一个保证低丢弃 PHB 的 DSCP。然后，此通信类从所有可识别 Diffserv 的路由器中接收一个低丢弃优先级 PHB，这可保证此类包的带宽。您可以向 QoS 策略中添加可为其他通信类指定不同优先级的其他 DSCP。Diffserv 系统将根据包的 DSCP 指示的优先级，为优先级较低的包提供带宽。

IPQoS 支持两种类型的转发行为：加速转发和保证转发，这两种行为在 Diffserv 体系结构中定义。

## 加速转发

**加速转发** (expedited forwarding, EF) 单跳行为可确保所有带有与 EF 相关的 DSCP 的通信类都被赋予最高优先级。带有 EF DSCP 的通信无需排队。EF 具有低丢失、低延迟和低抖动等特征。建议的 EF DSCP 为 101110。标记为 101110 的包在通过可识别 Diffserv 的网络、路由器一直到目的地的途中享有低丢弃优先级。向享有高价 SLA 的客户或应用程序指定优先级时，使用 EF DSCP。

## 保证转发

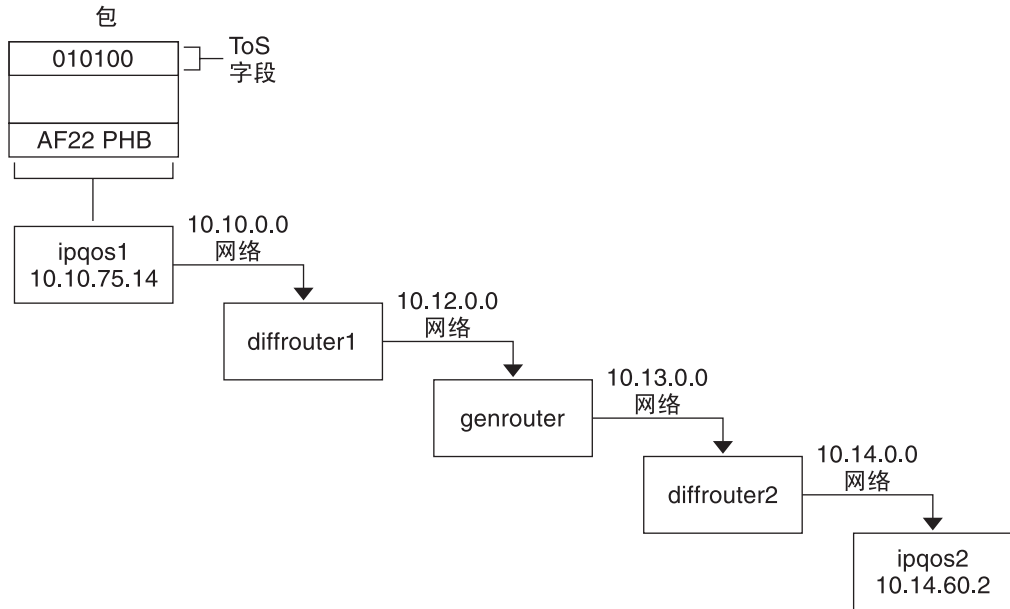
**保证转发** (assured forwarding, AF) 单跳行为提供了四种不同的转发类，您可以将这些类指定给包。每种转发类都提供三种丢弃优先级，如表 6-2 中所示。

各种 AF 代码点都提供为客户和应用程序指定不同服务级别的能力。对于 QoS 策略，可以在规划该策略时设置网络上通信和服务的优先级。然后向具有优先级的通信指定不同的 AF 级别。

## Diffserv 环境中的包转发

下图显示了某公司内联网的一部分，该环境局部启用了 Diffserv。在此方案中，网络 10.10.0.0 和 10.14.0.0 上的所有主机都启用了 IPQoS，并且这两个网络上的本地路由器均可识别 Diffserv。但是，没有针对 Diffserv 配置中间网络。

图 1-2 跨越可识别 Diffserv 的网络跃点的包转发



下面的步骤将跟踪此图中显示的包的流。这些步骤开始于源自主机 ipqos1 的包进程。然后，这些步骤连续通过若干跃点，一直到达主机 ipqos2。

1. ipqos1 上的用户通过运行 ftp 命令来访问相隔三个跃点的主机 ipqos2。
2. ipqos1 将其 QoS 策略应用于产生的包流。接着，ipqos1 成功地对 ftp 通信进行分类。

系统管理员已为源自本地网络 10.10.0.0 的所有传出 ftp 通信流量创建了一个类。为 ftp 类的通信指定了 AF22 单跳行为：二类、中丢弃优先级。为 ftp 类配置通信流速率 2Mb/sec。

3. ipqos-1 计量 ftp 流来确定此流是否超出确定的速率 2 Mbit/sec。
4. ipqos1 上的标记器使用 010100 DSCP（与 AF22 PHB 对应）标记传出 ftp 包中的 DS 字段。
5. 路由器 diffrouter1 接收 ftp 包。然后，diffrouter1 检查 DSCP。如果 diffrouter1 发生拥塞，将丢弃标记有 AF22 的包。
6. ftp 通信将根据在 diffrouter1 的文件中为 AF22 配置的单跳行为转发到协议中的下一个跃点。
7. ftp 通信遍历网络 10.12.0.0，一直到达不能识别 Diffserv 的 genrouter。这样，此通信便可获得“尽力服务”转发行为。
8. genrouter 将 ftp 通信传送到网络 10.13.0.0，其中通信由 diffrouter2 接收。

9. `diffrouter2` 可以识别 Diffserv。因此，此路由器根据路由器策略中为 AF22 包定义的 PHB 将 ftp 包转发到网络。
10. `ipqos2` 接收 ftp 通信。然后，`ipqos2` 提示 `ipqos1` 上的用户输入用户名和口令。



## 规划启用了 IPQoS 的网络（任务）

---

可以在任何运行 Oracle Solaris 的系统上配置 IPQoS。这样 IPQoS 系统便会与可识别 Diffserv 的路由器一起工作，在内联网上提供区分服务和通信流量管理。

本章包含在可识别 Diffserv 的网络上添加启用 IPQoS 的系统的规划任务。本章包含以下主题：

- 第 23 页中的“常规 IPQoS 配置规划（任务列表）”
- 第 24 页中的“规划 Diffserv 网络拓扑”
- 第 27 页中的“规划服务质量策略”
- 第 27 页中的“QoS 策略规划（任务列表）”
- 第 37 页中的“IPQoS 配置示例介绍”

---

注 – 未来发行版中可能会删除 IPQoS 功能。鼓励用户转而使用 `dladm`、`flowadm` 和相关命令，这些命令可支持类似的带宽资源控制功能。有关更多信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

---

### 常规 IPQoS 配置规划（任务列表）

在网络上实现区分服务（包括 IPQoS）需要进行大量规划。不仅必须考虑每个启用了 IPQoS 的系统的位置和功能，还要考虑每个系统与本地网络上的路由器的关系。下列任务映射列出了在网络中实施 IPQoS 的主要规划任务，以及指向完成这些任务的过程的链接。

任务	说明	参考
1. 规划与启用了 IPQoS 的系统结合使用的 Diffserv 网络拓扑。	了解各种 Diffserv 网络拓扑以确定用于您站点的最佳解决方案。	第 24 页中的“规划 Diffserv 网络拓扑”。

任务	说明	参考
2. 规划由 IPQoS 系统提供的不同服务类型。	将网络提供的服务类型组织成服务级别协议 (service-level agreements, SLA)。	第 27 页中的“规划服务质量策略”。
3. 为每个 IPQoS 系统规划 QoS 策略。	确定实现每种 SLA 所需的类、计量和记帐功能。	第 27 页中的“规划服务质量策略”。
4. 如果适用，为 Diffserv 路由器规划策略。	为与 IPQoS 系统一起使用的 Diffserv 路由器确定所有调度和排队策略。	有关排队和调度策略，请参阅路由器文档。

## 规划 Diffserv 网络拓扑

要为网络提供区分服务，至少需要一个启用了 IPQoS 的系统以及一个可识别 Diffserv 的路由器。您可以通过各种方法扩展这个基本方案，如本节中所述。

### Diffserv 网络的硬件策略

通常，客户针对服务器和服务器整合运行 IPQoS。与此相反，您也可以根据网络需求针对桌面系统运行 IPQoS。

以下列表介绍了可以配置 IPQoS 的系统：

- 提供各种服务的 Oracle Solaris 系统，例如 Web 服务器和数据库服务器
- 提供电子邮件、FTP 或其他常用网络应用程序的应用服务器
- Web 高速缓存服务器或代理服务器
- 由可识别 Diffserv 的负载均衡器管理的、启用了 IPQoS 的服务器场的网络
- 管理单个异构网络的通信的防火墙
- 属于虚拟局域网 (local area network, LAN) 一部分的 IPQoS 系统

您可以通过已经起作用的、可识别 Diffserv 的路由器将 IPQoS 系统引入网络拓扑中。如果您的路由器当前不提供 Diffserv，请考虑由 Cisco Systems、Juniper Networks 和其他路由器制造商提供的区分服务解决方案。如果本地路由器无法实现 Diffserv，则路由器会将标记的包传送到下一个跃点而不评估此标记。

### IPQoS 网络拓扑

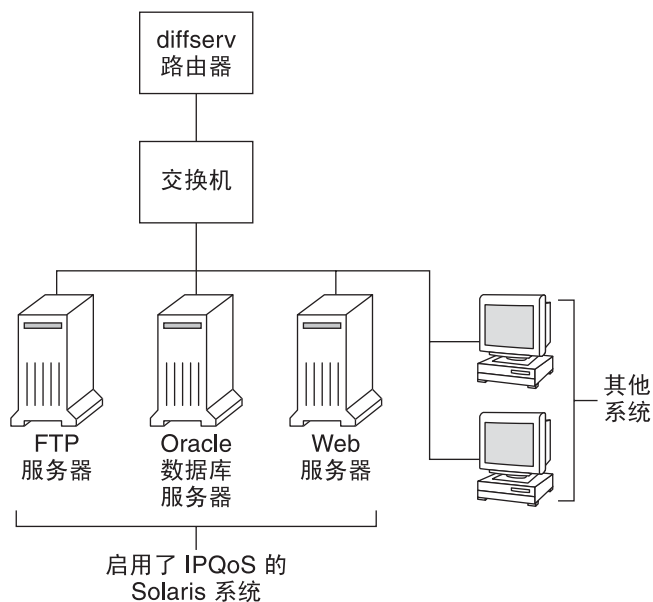
本节介绍可满足各种网络需求的 IPQoS 策略。

#### 单个主机上的 IPQoS

下图显示由启用了 IPQoS 的系统组成的单个网络。



图 2-1 网络段上的 IPQoS 系统



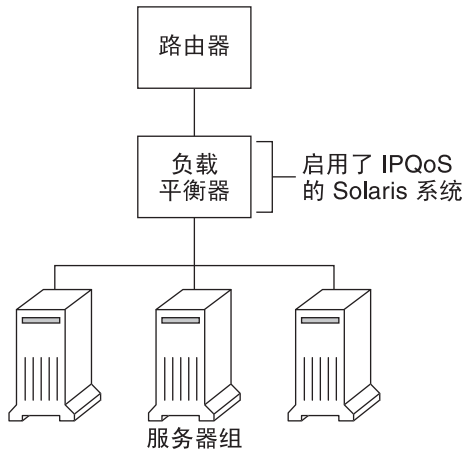
此网络仅为公司内联网的一段。通过在应用服务器和 Web 服务器上启用 IPQoS，您可以控制每个 IPQoS 系统释放传出通信的速率。如果使路由器可识别 Diffserv，则还可以进一步控制传入和传出的通信。

本指南中的示例使用“单个主机上的 IPQoS”方案。有关在整个指南中使用的拓扑示例，请参见图 2-4。

## 包含服务器场的网络上的 IPQoS

下图显示了包含数个异构服务器场的网络。

图 2-2 启用了 IPQoS 的服务器场的网络



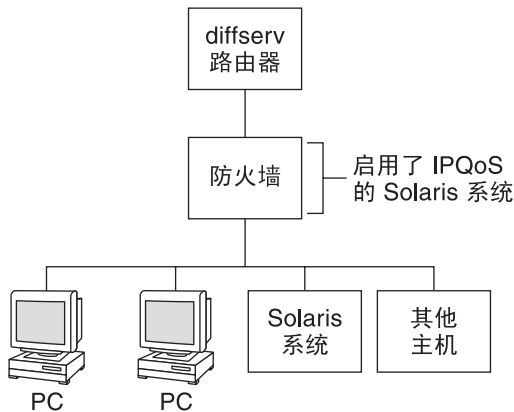
在此类拓扑中，路由器可识别 Diffserv，因此可以对传入和传出的通信进行排队和计速。负载均衡器也可识别 Diffserv，并且服务器场也启用了 IPQoS。负载均衡器可以使用选定器（例如用户 ID 和项目 ID）来提供路由器之外的其他过滤功能。这些选定器包括在应用程序数据中。

此方案提供了流控制和通信转发以管理本地网络上的拥塞。此方案还可防止服务器场的传出通信对内联网的其他部分造成过载。

## 防火墙上的 IPQoS

下图显示了公司网络的某一段，与其他段不同，该段通过防火墙保障安全。

图 2-3 由启用了 IPQoS 的防火墙保护的网路



在此方案中，通信流入可识别 Diffserv 的路由器，在此路由器中对包进行过滤和排队。然后，所有由路由器转发的传入通信通过启用 IPQoS 的防火墙。要使用 IPQoS，防火墙不能绕过 IP 转发栈。

防火墙的安全策略决定是允许传入通信进入内部网络还是远离内部网络。QoS 策略控制已经通过防火墙的传入通信的服务级别。根据 QoS 策略，还可以使用转发行为标记传出通信。

## 规划服务质量策略

规划服务质量 (quality-of-service, QoS) 策略时，您必须对网络所提供的服务进行检查、分类以及设置优先级。还必须评估可用的带宽，以确定在网络上释放每个通信类的速率。

### QoS 策略规划帮助

使用包括 IPQoS 配置文件所需信息的格式收集规划 QoS 策略的信息。例如，您可以使用以下模板来列出用于 IPQoS 配置文件的主要信息类别。

表 2-1 QoS 规划模板

类	优先级	过滤器	选定器	速率	是否转发？	是否记帐？
类 1	1	过滤器 1 过滤器 3	选定器 1 选定器 2	计量器速率，取决于计量器类型	标记器丢弃优先级	需要流记帐统计信息
类 1	1	过滤器 2	选定器 1 选定器 2	N/A	N/A	N/A
类 2	2	过滤器 1	选定器 1 选定器 2	计量器速率，取决于计量器类型	标记器丢弃优先级	需要流记帐统计信息
类 2	2	过滤器 2	选定器 1 选定器 2	N/A	N/A	N/A

您可以对每个主要类别进行划分，以进一步定义 QoS 策略。后续各节介绍了如何获取模板中显示的类别的信息。

### QoS 策略规划（任务列表）

本任务列表列出了规划 QoS 策略的主要任务以及执行各项任务的说明链接。

任务	说明	参考
1. 设计网络拓扑以支持 IPQoS。	标识网络上的主机和路由器以提供区分服务。	第 28 页中的“如何为 IPQoS 准备网络”
2. 定义网络上的服务必须归属的类。	检查站点所提供的服务和 SLA 的类型，并确定这些服务所归属的不同通信类。	第 29 页中的“如何定义 QoS 策略类”
3. 为类定义过滤器。	确定将特定类的通信从网络通信流中分离出来的最佳方法。	第 31 页中的“如何在 QoS 策略中定义过滤器”
4. 定义当包离开 IPQoS 系统时测量通信的流控制速率。	确定每个通信类的可接受流速率。	第 32 页中的“如何规划流控制”
5. 定义用于 QoS 策略的 DSCP 或用户优先级值。	规划一种方案，以确定当路由器或交换机处理流时，为通信流指定的转发行为。	第 34 页中的“如何规划转发行为”
6. 如果适用，为网络上的通信流设置统计信息监视规划。	评估通信类，以确定为进行记帐或统计而必须监视的通信流。	第 36 页中的“如何规划流记帐”

注 – 本节的其余部分介绍了如何规划启用了 IPQoS 的系统的 QoS 策略。要规划 Diffserv 路由器的 QoS 策略，请参阅路由器文档和路由器制造商的 Web 站点。

## ▼ 如何为 IPQoS 准备网络

以下过程列出了在创建 QoS 策略之前要执行的常规规划任务。

- 1 查看网络拓扑。然后，规划使用 IPQoS 系统和 Diffserv 路由器的策略。  
有关拓扑示例，请参见第 24 页中的“规划 Diffserv 网络拓扑”。
- 2 标识拓扑中要求 IPQoS 的主机或者可能会成为 IPQoS 服务优秀候选主机的主机。
- 3 确定可能会使用相同 QoS 策略的启用了 IPQoS 的系统。  
例如，如果您计划在网络中的所有主机上都启用 IPQoS，则标识出所有可能会使用相同 QoS 策略的主机。每个启用了 IPQoS 的系统都必须包含一个本地 QoS 策略，这在其 IPQoS 配置文件中实现。不过，您可以创建一个可供一系列系统使用的 IPQoS 配置文件。然后，可以将此配置文件复制到每个具有相同 QoS 策略要求的系统中。
- 4 查看并执行网络上的 Diffserv 路由器所要求的所有规划任务。  
有关详细信息，请参阅路由器文档和路由器制造商的 Web 站点。

## ▼ 如何定义 QoS 策略类

定义 QoS 策略的第一步是将通信流组织成多个类。您无需为 Diffserv 网络上的每种通信类型创建类。但是，根据您的网络拓扑，您可能必须为每个启用了 IPQoS 的系统创建不同的 QoS 策略。

---

注 - 有关类的概述，请参见第 15 页中的“IPQoS 类”。

---

下面的过程假设您已经确定网络上要启用 IPQoS 的系统，如第 28 页中的“如何为 IPQoS 准备网络”中所述。

### 1 创建用于管理 QoS 策略信息的 QoS 规划表。

有关建议，请参阅表 2-1。

### 2 针对网络上的每个 QoS 策略执行剩余步骤。

### 3 定义用于 QoS 策略的类。

以下问题是用于分析网络通信的可能类定义的指导。

#### ■ 贵公司是否为客户提供服务级别协议？

如果提供，请评估贵公司为客户提供的 SLA 的相对优先级别。可能会为被授予不同优先级别的客户提供相同的应用程序。

例如，贵公司可能会为每个客户提供 Web 站点宿主，这表示您需要为每个客户 Web 站点定义类。一种 SLA 可能提供一个优质 Web 站点作为一个服务级别，而另一种 SLA 可能为低端客户提供“尽力服务”的个人 Web 站点。此因素不仅指示了不同的 Web 站点类，而且指示了为 Web 站点类指定的单跳行为（单跳行为可能会彼此不同）。

#### ■ IPQoS 系统是否提供可能需要流控制的常用应用程序？

通过在产生过多通信流量的常用应用程序所在的服务器上启用 IPQoS，可以改进网络性能。常见示例包括电子邮件、网络新闻和 FTP。如果适用，请考虑为每种服务类型的传入和传出通信创建单独的类。例如，您可能为邮件服务器的 QoS 策略创建 mail-in 类和 mail-out 类。

#### ■ 您的网络是否运行需要最高优先级转发行为的特定应用程序？

所有需要最高优先级转发行为的关键应用程序都必须在路由器队列中拥有最高优先级。典型示例包括流视频和流音频。

为这些高优先级应用程序定义传入类和传出类。然后，将这些类添加到服务于这些应用程序的启用了 IPQoS 的系统以及 Diffserv 路由器的 QoS 策略中。

#### ■ 您的网络是否有过由于通信流占用大量带宽而必须受到控制的经历？

使用 `netstat`、`snoop` 和其他网络监视实用程序来搜索引起网络问题的通信类型。查看迄今已创建的类，然后为任何未定义的问题通信类别创建新的类。如果已经为某个问题通信类别定义了类，请定义计量器的速率以控制问题通信。

为网络上每个启用了 IPQoS 的系统的通信创建类。然后，每个 IPQoS 系统可以通过限制在网络上释放通信流的速率来处理问题通信。还要确保在 Diffserv 路由器上的 QoS 策略中定义这些问题类。这样，路由器可以按照其 QoS 策略中的配置对这些问题流进行排队和调度。

- 您是否需要获取有关特定通信类型的统计信息？

快速查看 SLA 可以了解需要记帐的客户通信类型。如果您的站点提供了 SLA，则您可能已经为需要记帐的通信创建了类。还可以定义类，以便收集正在监视的通信流的统计信息。还可以为出于安全原因而限制访问的通信创建类。

- 4 列出您在步骤 1 中创建的 QoS 规划表中已定义的类。

- 5 指定每个类的优先级。

例如，使优先级别 1 表示最高优先级类，然后为其余类指定按降序排列的优先级。所指定的优先级别仅用于进行组织。IPQoS 实际上并不使用您在 QoS 策略模板中设置的优先级别。此外，如果适用于 QoS 策略，您还可以为多个类指定相同的优先级。

- 6 完成定义类后，接下来为每个类定义过滤器，如第 31 页中的“如何在 QoS 策略中定义过滤器”中所述。

## 更多信息 设置类的优先级

创建类时，您会立即知道哪些类具有最高优先级、中等优先级以及尽力服务的优先级。当您按照第 34 页中的“如何规划转发行为”中所述为传出通信指定单跳行为时，优秀的设置类优先级方案变得尤为重要。

除了为类指定 PHB 之外，您还可以在过滤器中为类定义优先级选定器。优先级选定器仅可在启用了 IPQoS 的主机上使用。假定数个具有相同速率和 DSCP 的类有时会在离开 IPQoS 系统时争用带宽。每个类中的优先级选定器会对提供给其他同值类的服务级别进一步排序。

## 定义过滤器

您可以创建过滤器以将包流标识为特定类的成员。每个过滤器都包含选定器，这些选定器定义评估包流的条件。然后，启用了 IPQoS 的系统使用选定器中的条件从通信流中提取包。这样 IPQoS 系统便将包与类进行关联。有关过滤器的介绍，请参见第 15 页中的“IPQoS 过滤器”。

下表列出了最常用的选定器。前五个选定器表示 IPQoS 5 元组，IPQoS 系统使用这些选定器将包标识为流的成员。有关选定器的完整列表，请参见表 6-1。

表 2-2 常见的 IPQoS 选定器

名称	定义
saddr	源地址。
daddr	目标地址。
sport	源端口号。您既可以使用 <code>/etc/services</code> 中定义的已知端口号，也可以使用用户定义的端口号。
dport	目标端口号。
protocol	在 <code>/etc/protocols</code> 中，指定给通信流类型的 IP 协议号或协议名称。
ip_version	要使用的寻址样式。使用 IPv4 或 IPv6。IPv4 为缺省设置。
dsfield	DS 字段的内容，即 DSCP。使用此选定器来提取已使用特定 DSCP 标记的传入包。
priority	为类指定的优先级。有关更多信息，请参见第 29 页中的“如何定义 QoS 策略类”。
user	执行高级应用程序时使用的 UNIX 用户 ID 或用户名。
projid	执行高级应用程序时使用的项目 ID。
direction	通信流的方向。值为 LOCAL_IN、LOCAL_OUT、FWD_IN 或 FWD_OUT。

注 - 选择选定器时应谨慎。请根据需要仅使用相应数量的选定器来为类提取包。定义的选定器越多，对 IPQoS 性能的影响就越大。

## ▼ 如何在 QoS 策略中定义过滤器

**开始之前** 在执行下面的步骤之前，您应该已经完成第 29 页中的“如何定义 QoS 策略类”过程。

- 1 至少为您在第 29 页中的“如何定义 QoS 策略类”中创建的 QoS 规划表内的每个类创建一个过滤器。

如果适用，请考虑为每个类的传入和传出通信创建单独的过滤器。例如，向启用了 IPQoS 的 FTP 服务器的 QoS 策略中添加 `ftp-in` 过滤器和 `ftp-out` 过滤器。除了定义基本选定器之外，您还可以定义相应的 `direction` 选定器。

- 2 至少为类中的每个过滤器定义一个选定器。

使用在表 2-1 中介绍的 QoS 规划表为您定义的类填充过滤器。

### 示例 2-1 为 FTP 通信定义过滤器

下表为一示例，显示了如何为传出 FTP 通信定义过滤器。

类	优先级	过滤器	选定器
ftp-traffic	4	ftp-out	saddr 10.190.17.44 daddr 10.100.10.53 sport 21 direction LOCAL_OUT

- 另请参见
- 有关如何定义流控制方案的信息，请参阅第 32 页中的“如何规划流控制”。
  - 有关如何在流返回到网络流时为流定义转发行为的信息，请参阅第 34 页中的“如何规划转发行为”。
  - 有关如何规划特定通信类型的流记帐的信息，请参阅第 36 页中的“如何规划流记帐”。
  - 有关如何向 QoS 策略中添加更多类的信息，请参阅第 29 页中的“如何定义 QoS 策略类”。
  - 有关如何向 QoS 策略中添加更多过滤器的信息，请参阅第 31 页中的“如何在 QoS 策略中定义过滤器”。

## ▼ 如何规划流控制

流控制涉及测量类的通信流以及按照定义的速率在网络上释放包。规划流控制时，您将定义 IPQoS 计量模块使用的参数。计量器确定在网络上释放通信的速率。有关计量模块的介绍，请参见第 15 页中的“计量器 ( tokenmt 和 tswtclmt ) 概述”。

下面的过程假设您已经定义了过滤器和选定器，如第 31 页中的“如何在 QoS 策略中定义过滤器”中所述。

- 1 确定网络的最大带宽。
- 2 查看网络支持的所有 SLA。确定客户以及授予每个客户的服务类型。要授予某个服务级别，您可能需要计量由客户生成的特定通信类。
- 3 查看在第 29 页中的“如何定义 QoS 策略类”中创建的类列表。

确定除了那些与 SLA 关联的类之外，是否还有其他类需要计量。

假定 IPQoS 系统运行可生成高级别通信的应用程序。对应用程序的通信分类后，计量流以控制流的包返回到网络的速率。

---

注 – 并不需要计量所有的类。在查看类列表时应记住这一点。

---

- 4 确定每个类中由哪些过滤器选择需要流控制的通信。然后，完善需要计量的类列表。包含多个过滤器的类可能仅需要对一个过滤器进行计量。假定您为某个特定类的传入和传出通信定义过滤器。您可能会得出只有一个方向的通信需要流控制的结论。



## 5 为每个要进行流控制的类选择计量器模块。

将此模块名称添加到 QoS 规划表的计量器列中。

## 6 将每个待计量类的速率添加到组织表中。

如果您使用 `tokenmt` 模块，则需要定义以下速率（以位/秒为单位）。

- 承诺速率
- 峰值速率

如果这些速率足以计量特定类，则可以仅为 `tokenmt` 定义承诺速率和承诺突发速率。

如果需要，还可以定义以下速率：

- 承诺突发速率
- 峰值突发速率

有关 `tokenmt` 速率的完整定义，请参阅第 80 页中的“将 `tokenmt` 配置为双速率计量器”。您还可以在 `tokenmt (7ipp)` 手册页中获得更多详细信息。

如果您使用 `tswtclmt` 模块，则需要定义以下速率（以每秒位数为单位）。

- 承诺速率
- 峰值速率

您还可以定义时间窗口大小（以毫秒为单位）。这些速率在第 81 页中的“`tswtclmt` 计量模块”和 `twstclmt(7ipp)` 手册页中定义。

## 7 添加已计量通信的通信一致性结果。

两个计量模块的结果为绿色、红色和黄色。将适用于您所定义的速率的通信一致性结果添加到 QoS 组织表中。第 79 页中的“`计量器模块`”中全面介绍了计量器的结果。

您需要确定应该对符合或者不符合承诺速率的通信执行的操作。通常（但不总是），此操作是使用单跳行为标记包头。对绿色级别通信执行的可接受操作可能是在通信没有超过承诺速率时继续进行处理。另一个操作可能是在通信流量超过峰值速率时丢弃类的包。

### 示例 2-2 定义计量器

下表为一示例，显示了电子邮件通信类的计量器项。IPQoS 系统所在网络的总带宽为 100 兆位/秒或 10000000 位/秒。QoS 策略为电子邮件类指定低优先级。此类也接收尽力服务的转发行为。

类	优先级	过滤器	选定器	速率
email	8	mail_in	daddr10.50.50.5 dport imap direction LOCAL_IN	
email	8	mail_out	saddr10.50.50.5 sport imap direction LOCAL_OUT	计量器 = tokenmt 承诺速率 = 5000000 承诺突发速率 = 5000000 峰值速率 = 10000000 峰值突发速率 = 1000000 绿色优先级 = 继续进行处理 黄色优先级 = 标记黄色 PHB 红色优先级 = 丢弃

- 另请参见
- 有关如何在包返回到网络流时为流定义转发行为的信息，请参阅第 34 页中的“如何规划转发行为”。
  - 有关如何规划特定通信类型的流记帐的信息，请参阅第 36 页中的“如何规划流记帐”。
  - 有关如何向 QoS 策略中添加更多类的信息，请参阅第 29 页中的“如何定义 QoS 策略类”。
  - 有关如何向 QoS 策略中添加更多过滤器的信息，请参阅第 31 页中的“如何在 QoS 策略中定义过滤器”。
  - 有关如何定义其他流控制方案的信息，请参阅第 32 页中的“如何规划流控制”。
  - 有关如何创建 IPQoS 配置文件的信息，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

## ▼ 如何规划转发行为

转发行为确定要转发到网络的通信流的优先级和丢弃优先级。您可以选择两种主要转发行为：相对于其他通信类设置某个类的流优先级，或者完全丢弃流。

Diffserv 模型使用标记器为通信流指定所选的转发行为。IPQoS 提供以下标记器模块。

- dscpmk—使用 DSCP 标记 IP 包的 DS 字段
- dlcosmk—使用服务类 (class-of-service, CoS) 值标记数据报的 VLAN 标记

注 – 本节中的建议专指 IP 包。如果 IPQoS 系统包括 VLAN 设备，则可以使用 `dlcosmk` 标记器为数据报标记转发行为。有关更多信息，请参阅第 84 页中的“将 `dlcosmk` 标记器用于 VLAN 设备”。

要设置 IP 通信的优先级，您需要为每个包指定一个 DSCP。 `dscpmk` 标记器使用 DSCP 标记包的 DS 字段。您可以从与转发行为类型关联的一组已知代码点中为类选择 DSCP。这些已知代码点包括用于 EF PHB 的 46 (101110) 以及用于 AF PHB 的一系列代码点。有关 DSCP 和转发的概述信息，请参阅第 18 页中的“启用了 IPQoS 的网络上的通信转发”。

**开始之前** 下面的步骤假设您已经为 QoS 策略定义了类和过滤器。尽管您经常同时使用计量器和标记器来控制通信，但是也可以单独使用标记器来定义转发行为。

**1 查看迄今已创建的类以及已为每个类指定的优先级。**

并不需要标记所有的通信类。

**2 为优先级最高的类指定 EF 单跳行为。**

EF PHB 保证具有 EF DSCP 46 (101110) 的包在网络上的释放优先于具有任何 AF PHB 的包。请为最高优先级的通信使用 EF PHB。有关 EF 的更多信息，请参阅第 83 页中的“加速转发 (Expedited Forwarding, EF) PHB”。

**3 为要计量通信的类指定转发行为。**

**4 按照已经为类指定的优先级，为其余的类指定 DS 代码点。**

### 示例 2-3 游戏应用程序的 QoS 策略

通常，对通信进行计量的原因如下：

- SLA 保证在网络繁忙时为此类的包提供更多服务或更少服务。
- 较低优先级的类可能会对网络进行泛洪攻击。

您可以同时使用标记器和计量器为这些类提供区分服务和带宽管理。例如，下表显示了 QoS 策略的一部分。此策略为生成高级别通信的常用游戏应用程序定义类。

类	优先级	过滤器	选定器	速率	是否转发？
games_app	9	games_in	sport 6080	N/A	N/A

类	优先级	过滤器	选定器	速率	是否转发？
games_app	9	games_out	dport 6081	计量器 = tokenmt 承诺速率 = 5000000 承诺突发速率 = 5000000 峰值速率 = 10000000 峰值突发速率 = 15000000 绿色优先级 = 继续进行处理 黄色优先级 = 标记黄色 PHB 红色优先级 = 丢弃	绿色 = AF31 黄色 = AF42 红色 = 丢弃

转发行为将为符合承诺速率或低于峰值速率的 games\_app 通信指定低优先级的 DSCP。当 games\_app 通信流量超过峰值速率时，QoS 策略指示丢弃来自 games\_app 的包。表 6-2 中列出了所有 AF 代码点。

- 另请参见
- 有关如何规划特定通信类型的流记帐的信息，请参阅第 36 页中的“如何规划流记帐”。
  - 有关如何向 QoS 策略中添加更多类的信息，请参阅第 29 页中的“如何定义 QoS 策略类”。
  - 有关如何向 QoS 策略中添加更多过滤器的信息，请参阅第 31 页中的“如何在 QoS 策略中定义过滤器”。
  - 有关如何定义流控制方案的信息，请参阅第 32 页中的“如何规划流控制”。
  - 有关如何在包返回到网络流时为流定义其他转发行为的信息，请参阅第 34 页中的“如何规划转发行为”。
  - 有关如何创建 IPQoS 配置文件的信息，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

## ▼ 如何规划流记帐

您可以使用 IPQoS flowacct 模块跟踪通信流以实现记帐或网络管理。请使用以下过程来确定您的 QoS 策略是否应该包括流记帐。

### 1 贵公司是否为客户提供 SLA ？

如果提供，则应该使用流记帐。查看 SLA 来确定贵公司希望为客户进行记帐的网络通信类型。然后，查看您的 QoS 策略来确定由哪些类选择要记帐的通信。

## 2 是否存在可能需要进行监视或测试以免出现网络问题的应用程序？

如果存在，请考虑使用流记帐来查看这些应用程序的行为。查看您的 QoS 策略来确定已经为需要进行监视的通信指定的类。

## 3 在 QoS 规划表中，为每个需要流记帐的类在流记帐列中标记 Y。

- 另请参见
- 有关如何向 QoS 策略中添加更多类的信息，请参阅第 29 页中的“如何定义 QoS 策略类”。
  - 有关如何向 QoS 策略中添加更多过滤器的信息，请参阅第 31 页中的“如何在 QoS 策略中定义过滤器”。
  - 有关如何定义流控制方案的信息，请参阅第 32 页中的“如何规划流控制”。
  - 有关如何在包返回到网络流时为流定义转发行为的信息，请参阅第 34 页中的“如何规划转发行为”。
  - 有关如何规划特定通信类型的其他流记帐的信息，请参阅第 36 页中的“如何规划流记帐”。
  - 有关如何创建 IPQoS 配置文件的信息，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

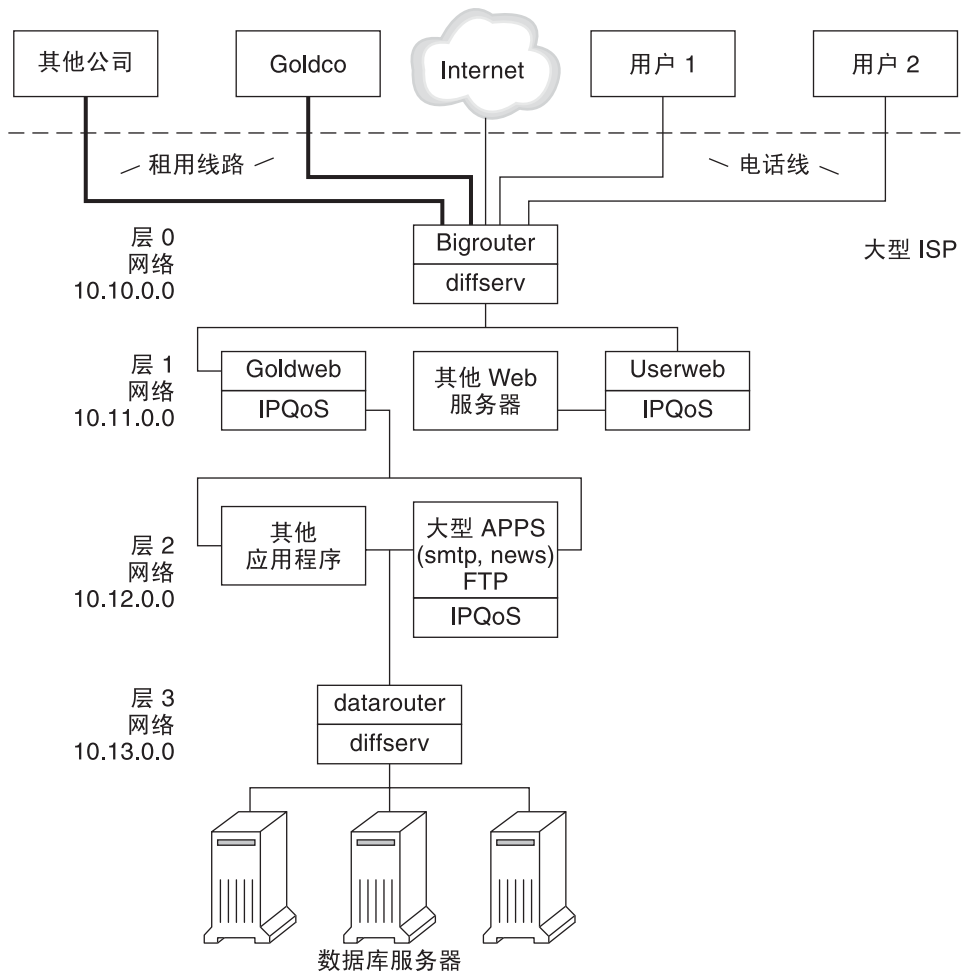
# IPQoS 配置示例介绍

本指南其余章节中的任务均使用本节中介绍的示例 IPQoS 配置。此示例显示了 BigISP 这一虚构服务提供商的公共内联网上的区分服务解决方案。BigISP 为通过租用线路访问 BigISP 的大型公司提供服务。通过调制解调器拨入的个人也可以购买 BigISP 提供的服务。

## IPQoS 拓扑

下图显示了用于 BigISP 公共内联网的网络拓扑。

图 2-4 IPQoS 拓扑示例



BigISP 在其公共内联网上实现以下四层：

- **0 层**—网络 10.10.0.0 包括名为 Bigrouter 的大型 Diffserv 路由器，它具有外部接口和内部接口。数家公司（包括一家名为 Goldco 的大型组织）已经租用了终止于 Bigrouter 的租用线路服务。0 层还处理通过电话线或 ISDN 呼叫的个体客户的业务。
- **1 层**—网络 10.11.0.0 提供 Web 服务。Goldweb 服务器承载 Goldco 从 BigISP 处购买的属于高级服务一部分的 Web 站点。服务器 Userweb 承载由个体客户购买的小型 Web 站点。Goldweb 和 Userweb 都启用了 IPQoS。
- **2 层**—网络 10.12.0.0 提供可供所有客户使用的应用程序。BigAPPS（应用服务器之一）启用了 IPQoS。BigAPPS 提供 SMTP、新闻和 FTP 服务。

- **3层**—网络 **10.13.0.0** 包含大型数据库服务器。对 3 层的访问由 Diffserv 路由器 **datarouter** 所控制。





## 创建 IPQoS 配置文件（任务）

---

本章介绍如何创建 IPQoS 配置文件。本章包含以下主题：

- 第 41 页中的“在 IPQoS 配置文件中定义 QoS 策略（任务列表）”
- 第 42 页中的“创建 QoS 策略所用的工具”
- 第 43 页中的“为 Web 服务器创建 IPQoS 配置文件”
- 第 54 页中的“为应用服务器创建 IPQoS 配置文件”
- 第 63 页中的“在路由器上提供区分服务”

本章假设您已经定义了完整的 QoS 策略，并可以使用此策略作为 IPQoS 配置文件的基础。有关 QoS 策略规划的说明，请参阅第 27 页中的“规划服务质量策略”。

---

注 – 未来发行版中可能会删除 IPQoS 功能。鼓励用户转而使用 `dladm`、`flowadm` 和相关命令，这些命令可支持类似的带宽资源控制功能。有关更多信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

---

## 在 IPQoS 配置文件中定义 QoS 策略（任务列表）

此任务列表列出了用于创建 IPQoS 配置文件的常规任务以及说明这些任务的执行步骤的章节链接。

任务	说明	参考
1. 规划启用了 IPQoS 的网络配置。	确定本地网络中应该启用 IPQoS 的系统。	第 28 页中的“如何为 IPQoS 准备网络”
2. 为网络中的 IPQoS 系统规划 QoS 策略。	将通信流标识为不同的服务类。然后，确定需要进行通信管理的流。	第 27 页中的“规划服务质量策略”
3. 创建 IPQoS 配置文件并定义其第一个操作。	创建 IPQoS 文件，调用 IP 分类器，并定义要处理的类。	第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”

任务	说明	参考
4. 为类创建过滤器。	添加用于控制选择哪些通信并将其归为一类的过滤器。	第 47 页中的“如何在 IPQoS 配置文件中定义过滤器”
5. 向 IPQoS 配置文件中添加更多的类和过滤器。	创建要由 IP 分类器处理的更多类和过滤器。	第 52 页中的“如何为尽力服务 Web 服务器创建 IPQoS 配置文件”
6. 添加具有可配置计量模块的参数的 action 语句。	如果 QoS 策略要求流控制，请为计量器指定流控制速率和一致性级别。	第 60 页中的“如何在 IPQoS 配置文件中配置流控制”
7. 添加具有可配置标记器的参数的 action 语句。	如果 QoS 策略要求区分转发行为，请定义如何转发通信类。	第 48 页中的“如何在 IPQoS 配置文件中定义通信转发”
8. 添加具有可配置流记帐模块的参数的 action 语句。	如果 QoS 策略要求收集有关通信流的统计信息，请定义如何收集记帐统计信息。	第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”
9. 应用 IPQoS 配置文件。	将指定的 IPQoS 配置文件内容添加到相应的内核模块中。	第 66 页中的“如何将新配置应用于 IPQoS 内核模块”
10. 在路由器文件中配置转发行为。	如果网络中的任何 IPQoS 配置文件定义了转发行为，请将最终的 DSCP 添加到路由器上的相应调度文件中。	第 63 页中的“如何在启用了 IPQoS 的网络中配置路由器”

## 创建 QoS 策略所用的工具

用于网络的 QoS 策略位于 IPQoS 配置文件中。您可以使用文本编辑器创建此配置文件。然后，将此文件作为参数提供给 IPQoS 配置实用程序 `ipqosconf`。指示 `ipqosconf` 应用配置文件中定义的策略时，会将此策略写入内核 IPQoS 系统。有关 `ipqosconf` 命令的详细信息，请参阅 `ipqosconf(1M)` 手册页。有关如何使用 `ipqosconf` 的说明，请参阅第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。

## 基本的 IPQoS 配置文件

IPQoS 配置文件由 action 语句树组成，这些语句可实现在第 27 页中的“规划服务质量策略”中定义的 QoS 策略。IPQoS 配置文件将配置 IPQoS 模块。每个操作语句都包含一组类、过滤器或参数，这些内容要由在操作语句中调用的模块进行处理。

有关 IPQoS 配置文件的完整语法，请参阅示例 6-3 和 `ipqosconf(1M)` 手册页。

## 配置 IPQoS 拓扑示例

本章中的任务说明如何为三个启用了 IPQoS 的系统创建 IPQoS 配置文件。这些系统是图 2-4 中介绍的 BigISP 公司的网络拓扑的一部分。

- Goldweb—托管已经购买了高级 SLA 的客户的 Web 站点的 Web 服务器
- Userweb—托管已经购买了“尽力服务”SLA 的家庭用户的个人 Web 站点并且功能较弱的 Web 服务器
- BigAPPS—为黄金级客户和“尽力服务”客户提供邮件、网络新闻和 FTP 服务的应用服务器

这三个配置文件说明了最常见的 IPQoS 配置。您可以将下一节中的样例文件用作自己的 IPQoS 实现的模板。

## 为 Web 服务器创建 IPQoS 配置文件

本节通过说明如何为高级 Web 服务器创建配置来介绍 IPQoS 配置文件。同时，本节还说明如何在其他配置文件中为托管个人 Web 站点的服务器配置完全不同的服务级别。这两台服务器是图 2-4 中所示网络示例的一部分。

以下配置文件定义了 Goldweb 服务器的 IPQoS 活动。此服务器可托管已经购买了高级 SLA 的公司 Goldco 的 Web 站点。

示例 3-1 高级 Web 服务器的 IPQoS 配置文件样例

```

fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name goldweb
        next_action markAF11
        enable_stats FALSE
    }
    class {
        name video
        next_action markEF
        enable_stats FALSE
    }
}
filter {
    name webout
    sport 80
    direction LOCAL_OUT
    class goldweb
}
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
}

```

示例 3-1 高级 Web 服务器的 IPQoS 配置文件样例 (续)

```
action {
  module dscpmk
  name markAF11
  params {
    global_stats FALSE
    dscp_map{0-63:10}
    next_action continue
  }
}
action {
  module dscpmk
  name markEF
  params {
    global_stats TRUE
    dscp_map{0-63:46}
    next_action acct
  }
}
action {
  module flowacct
  name acct
  params {
    enable_stats TRUE
    timer 10000
    timeout 10000
    max_limit 2048
  }
}
}
```

以下配置文件定义了 Userweb 上的 IPQoS 活动。此服务器托管具有低价位 SLA 或**尽力服务** SLA 的个人 Web 站点。此服务级别保证 IPQoS 系统在处理来自更昂贵 SLA 的客户的通信之后，为“尽力服务”客户提供最佳的服务。

示例 3-2 尽力服务 Web 服务器的配置样例

```
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
  }
}
```

示例 3-2 尽力服务 Web 服务器的配置样例 (续)

```

}
action {
    module dscpmk
    name markAF12
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}

```

## ▼ 如何创建 IPQoS 配置文件并定义通信类

您可以在任何最易于维护的目录中创建第一个 IPQoS 配置文件。本章中的任务使用 `/var/ipqos` 目录作为 IPQoS 配置文件的存储位置。下面的过程将构建示例 3-1 中介绍的 IPQoS 配置文件的初始段。

---

注 - 创建 IPQoS 配置文件时，请务必慎用花括号 ({} ) 将每条 `action` 语句和子句括起来。有关花括号的用法示例，请参见示例 3-1。

---

- 1 登录到高级 Web 服务器，并创建扩展名为 `.qos` 的新 IPQoS 配置文件。  
每个 IPQoS 配置文件都必须以版本号 `fmt_version 1.0` 开头，作为其第一个未注释的行。
- 2 在第一个参数之后添加初始 `action` 语句，此语句将配置通用 IP 分类器 `ipgpc`。  
此初始操作将开始生成组成 IPQoS 配置文件的 `action` 语句树。例如，`/var/ipqos/Goldweb.qos` 文件以初始 `action` 语句开头来调用 `ipgpc` 分类器。

```
fmt_version 1.0
```

```
action {
    module ipgpc
    name ipgpc.classify

```

`fmt_version 1.0`      开始启用 IPQoS 配置文件。

`action {`              开始操作语句。

`module ipgpc`          将 `ipgpc` 分类器配置为配置文件中的第一个操作。

`name ipgpc.classify`    定义分类器 `action` 语句的名称，此名称必须始终为 `ipgpc.classify`。

有关 `action` 语句的详细语法信息，请参阅第 89 页中的“`action` 语句”和 `ipqosconf(1M)` 手册页。

### 3 添加带有统计参数 `global_stats` 的 `params` 子句。

```
params {
    global_stats TRUE
}
```

在 `ipgpc.classify` 语句中使用参数 `global_stats TRUE` 可为此操作启用统计信息收集。如果在类子句定义中指定了 `enable_stats TRUE`，`global_stats TRUE` 还允许按类启用统计信息收集。

打开统计功能会影响性能。您可能需要收集有关新 IPQoS 配置文件的统计信息，以检验 IPQoS 是否正常运行。随后，可以通过将 `global_stats` 参数更改为 `FALSE` 来关闭统计信息收集。

但是，全局统计功能是一种在 `params` 子句中定义参数类型的参数类型。有关 `params` 子句的语法信息和其他详细信息，请参阅第 91 页中的“[params 子句](#)”和 `ipqosconf(1M)` 手册页。

### 4 定义类以标识要送达高级服务器的通信。

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

此语句称为类子句。class 子句具有以下内容。

<code>name goldweb</code>	创建类 <code>goldweb</code> 以标识要送达 Goldweb 服务器的通信。
<code>next_action markAF11</code>	指示 <code>ipgpc</code> 模块将 <code>goldweb</code> 类的包传递到 <code>markAF11</code> 操作语句。 <code>markAF11</code> 操作语句将调用 <code>dscpmk</code> 标记器。
<code>enable_stats FALSE</code>	用于提取 <code>goldweb</code> 类的统计信息。但是，由于 <code>enable_stats</code> 的值为 <code>FALSE</code> ，因此不会为此类启用统计功能。

有关 class 子句语法的详细信息，请参见第 90 页中的“[class 子句](#)”和 `ipqosconf(1M)` 手册页。

### 5 定义类以标识必须具有最高优先级转发的应用程序。

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

<code>name video</code>	创建类 <code>video</code> 以标识从 Goldweb 服务器传出的流视频通信。
<code>next_action markEF</code>	指示 <code>ipgpc</code> 模块在 <code>ipgpc</code> 完成处理之后，将 <code>video</code> 类的包传递到 <code>markEF</code> 语句。 <code>markEF</code> 语句将调用 <code>dscpmk</code> 标记器。
<code>enable_stats FALSE</code>	用于针对 <code>video</code> 类启用统计信息收集。但是，由于 <code>enable_stats</code> 的值为 <code>FALSE</code> ，因此不会为此类启用统计信息收集。

- 另请参见
- 要为您刚创建的类定义过滤器，请参阅第 47 页中的“如何在 IPQoS 配置文件中定义过滤器”。
  - 要为配置文件创建另一个类子句，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

## ▼ 如何在 IPQoS 配置文件中定义过滤器

以下过程说明如何在 IPQoS 配置文件中为类定义过滤器。

**开始之前** 此过程假设您已经开始创建文件并已定义类。这些步骤将继续构建在中第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”创建的 `/var/ipqos/Goldweb.qos` 文件。

---

注 - 当您创建 IPQoS 配置文件时，必须注意使用花括号 ({} ) 括起每条 `class` 子句和每条 `filter` 子句。有关花括号的用法示例，请参见示例 3-1。

---

### 1 打开 IPQoS 配置文件，并定位到已定义的最后一个人的类的结尾。

例如，在启用了 IPQoS 的服务器 Goldweb 上，应从 `/var/ipqos/Goldweb.qos` 中的以下 `class` 子句之后开始：

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

### 2 定义 `filter` 子句以选择 IPQoS 系统的传出通信。

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

`name webout` 为过滤器提供名称 `webout`。

`sport 80` 选择源端口 80 传出的通信，此端口是众所周知的用于 HTTP (Web) 通信的端口。

`direction LOCAL_OUT` 进一步选择从本地系统传出的通信。

`class goldweb` 标识过滤器所属的类，在此实例中为类 `goldweb`。

有关 IPQoS 配置文件中 `filter` 子句的语法信息和其他详细信息，请参阅第 91 页中的“`filter` 子句”。

### 3 定义 filter 子句以选择 IPQoS 系统上的流视频通信。

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

name videoout	为过滤器提供名称 videoout。
sport videosrv	选择源端口 videosrv 传出的通信，此端口是先前针对此系统上的流视频应用程序定义的端口。
direction LOCAL_OUT	进一步选择从本地系统传出的通信。
class video	标识过滤器所属的类，在此实例中为类 video。

- 另请参见
- 要定义标记器模块的转发行为，请参阅第 48 页中的“如何在 IPQoS 配置文件中定义通信转发”。
  - 要定义计量模块的流控制参数，请参阅第 60 页中的“如何在 IPQoS 配置文件中配置流控制”。
  - 要激活 IPQoS 配置文件，请参阅第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。
  - 要定义其他过滤器，请参阅第 47 页中的“如何在 IPQoS 配置文件中定义过滤器”。
  - 要从应用程序中为通信流创建类，请参阅第 56 页中的“如何为应用服务器配置 IPQoS 配置文件”。

## ▼ 如何在 IPQoS 配置文件中定义通信转发

以下过程说明如何通过将类的单跳行为添加到 IPQoS 配置文件来定义通信转发。

- 开始之前** 此过程假设您具有已定义类和过滤器的 IPQoS 配置文件。这些步骤将继续构建示例 3-1 中的 /var/ipqos/Goldweb.qos 文件。

---

注 - 此过程显示如何使用 dscpmk 标记器模块来配置通信转发。有关在 VLAN 系统上使用 dlcosmk 标记器转发通信的信息，请参阅第 84 页中的“将 dlcosmk 标记器用于 VLAN 设备”。

---

### 1 打开 IPQoS 配置文件，并定位到已定义的最后一个过滤器的结尾。

例如，在启用了 IPQoS 的服务器 Goldweb 上，应从 /var/ipqos/Goldweb.qos 中的以下 filter 子句之后开始：

```
filter {
    name videoout
    sport videosrv
```



```

        direction LOCAL_OUT
        class video
    }
}

```

请注意，此 `filter` 子句位于 `ipgpc` 分类器 `action` 语句的结尾。因此，需要使用闭花括号来终止过滤器，并使用第二个闭花括号来终止 `action` 语句。

## 2 使用以下 `action` 语句调用标记器。

```

action {
    module dscpmk
    name markAF11
}

```

`module dscpmk` 调用标记器模块 `dscpmk`。

`name markAF11` 为 `action` 语句提供名称 `markAF11`。

先前定义的类 `goldweb` 包括 `next_action markAF11` 语句。分类器结束处理之后，此语句将通信流发送到 `markAF11` 操作语句。

## 3 定义标记器处理通信流的操作。

```

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

```

`global_stats FALSE` 对 `markAF11` 标记器 `action` 语句启用统计信息收集。但是，由于 `enable_stats` 的值为 `FALSE`，因此不会收集统计信息。

`dscp_map{0-63:10}` 将 DSCP 10 指定给通信类 `goldweb` 的包头，标记器当前正在处理此通信类。

`next_action continue` 指示不需要对通信类 `goldweb` 的包进行进一步处理，并指示可以将这些包返回到网络流中。

DSCP 为 10 指示标记器将 `dscp` 映射中的所有项都设置为十进制值 10（二进制值 001010）。此代码点指示 `goldweb` 通信类的包遵守 AF11 单跳行为。AF11 保证 DSCP 为 10 的所有包都拥有低丢弃率、高优先级的服务。因此，对于 `Goldweb` 上高级客户的传出通信，将提供可用于保证转发 (Assured Forwarding, AF) PHB 的最高优先级。有关说明 AF 可能的 DSCP 的表，请参阅表 6-2。

## 4 开始另一条标记器 `action` 语句。

```

action {
    module dscpmk
    name markEF
}

```

`module dscpmk` 调用标记器模块 `dscpmk`。

`name markEF` 为 `action` 语句提供名称 `markEF`。

5 为标记器定义操作以处理通信流。

```

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

```

`global_stats TRUE` 用于针对类 `video` 启用统计信息收集，此类将选择流视频包。

`dscp_map{0-63:46}` 将 DSCP 46 指定给通信类 `video` 的包头，标记器当前正在处理此通信类。

`next_action acct` 指示 `dscpmk` 模块在 `dscpmk` 完成处理之后，将类 `video` 的包传递到 `acct action` 语句。`acct action` 语句将调用 `flowacct` 模块。

DS 字段中的 DSCP 为 46 指示 `dscpmk` 模块将 `dscp` 映射中的所有项都设置为十进制值 46（二进制值 101110）。此代码点指示 `video` 通信类的包受加速转发 (Expedited Forwarding, EF) 单跳行为的限制。

---

注 - 建议用于 EF 的代码点为 46（二进制值 101110）。其他 DSCP 将为包指定 AF PHB。

---

EF PHB 保证 IPQoS 系统和可识别区分服务的系统为 DSCP 为 46 的包提供最高优先级。流应用程序需要最高优先级的服务，这是在 QoS 策略中为流应用程序指定 EF PHB 的基本原因。有关加速转发 PHB 的更多详细信息，请参阅第 83 页中的“加速转发 (Expedited Forwarding, EF) PHB”。

6 将刚创建的 DSCP 添加到 Diffserv 服务路由器上的相应文件中。

有关更多信息，请参阅第 63 页中的“如何在启用了 IPQoS 的网络中配置路由器”。

另请参见

- 要开始收集通信流上的流记帐统计信息，请参阅第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”。
- 要定义标记器模块的转发行为，请参阅第 48 页中的“如何在 IPQoS 配置文件中定义通信转发”。
- 要定义计量模块的流控制参数，请参阅第 60 页中的“如何在 IPQoS 配置文件中配置流控制”。
- 要激活 IPQoS 配置文件，请参阅第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。
- 要定义其他过滤器，请参阅第 47 页中的“如何在 IPQoS 配置文件中定义过滤器”。
- 要从应用程序中为通信流创建类，请参阅第 56 页中的“如何为应用服务器配置 IPQoS 配置文件”。

## ▼ 如何在 IPQoS 配置文件中为类启用记帐

以下过程说明如何在 IPQoS 配置文件中对通信类启用记帐。此过程说明了如何为 video 类定义流记帐，有关该类的信息在第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”中进行了介绍。此类将选择流视频通信，此通信必须作为高级客户的 SLA 的一部分进行记帐。

**开始之前** 此过程假设您具有已定义类、过滤器、计量操作（如果适用）以及标记操作（如果适用）的 IPQoS 配置文件。这些步骤将继续构建示例 3-1 中的 /var/ipqos/Goldweb.qos 文件。

- 1 打开 IPQoS 配置文件，并定位到已定义的最后一条 action 语句的结尾。

例如，在启用了 IPQoS 的服务器 Goldweb 上，应从 /var/ipqos/Goldweb.qos 中的以下 markEF action 语句之后开始。

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

- 2 开始编写一个调用流记帐的 action 语句。

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

module flowacct      调用流记帐模块 flowacct。

name acct              为 action 语句提供名称 acct

- 3 定义 params 子句以控制通信类记帐。

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

global\_stats TRUE      为类 video 启用统计信息收集，此类将选择流视频包。

timer 10000            指定扫描流表以查找超时流的时间间隔（以毫秒为单位）。在此参数中，此时间间隔为 10000 毫秒。

<code>timeout 10000</code>	指定最小时间间隔超时值。如果在超时时间间隔内未发现流的包，则表示流“超时”。在此参数中，包将在 10000 毫秒之后超时。
<code>max_limit 2048</code>	在流表中针对该操作实例设置活动流记录的最大数目。
<code>next_action continue</code>	指示不需要对通信类 video 的包进行进一步处理，并指示可以将这些包返回到网络流中。

flowacct 模块将收集有关特定类的包的统计信息，直到达到指定的 timeout 值为止。

- 另请参见
- 要在路由器上配置单跳行为，请参阅第 63 页中的“如何在启用了 IPQoS 的网络中配置路由器”。
  - 要激活 IPQoS 配置文件，请参阅第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。
  - 要从应用程序中为通信流创建类，请参阅第 56 页中的“如何为应用服务器配置 IPQoS 配置文件”。

## ▼ 如何为尽力服务 Web 服务器创建 IPQoS 配置文件

尽力服务 Web 服务器的 IPQoS 配置文件与高级 Web 服务器的 IPQoS 配置文件稍有不同。以下过程以示例 3-2 中的配置文件为例。

- 1 登录到尽力服务 Web 服务器。
- 2 创建扩展名为 .qos 的新 IPQoS 配置文件。

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

/var/ipqos/userweb.qos 文件必须以调用 ipgpc 分类器的局部 action 语句开始。此外，action 语句还包含 params 子句，用于启用统计信息收集。有关 action 语句的说明，请参见第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

- 3 定义用于标识要送达尽力服务 Web 服务器的通信的类。

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

name userweb                      创建名为 userweb 的类以转发来自用户的 Web 通信。

`next_action markAF1` 指示 `ipgpc` 模块在 `ipgpc` 完成处理之后，将类 `userweb` 的包传递到 `markAF12 action` 语句。`markAF12 action` 语句将调用 `dscpmk` 标记器。

`enable_stats FALSE` 用于针对 `userweb` 类启用统计信息收集。但是，由于 `enable_stats` 的值为 `FALSE`，因此不会为此类收集统计信息。

有关此 `class` 子句任务的解释，请参见第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

#### 4 定义 `filter` 子句以选择 `userweb` 类的通信流。

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

`name webout` 为过滤器提供名称 `webout`。

`sport 80` 选择源端口 80 传出的通信，此端口是众所周知的用于 HTTP (Web) 通信的端口。

`direction LOCAL_OUT` 进一步选择从本地系统传出的通信。

`class userweb` 标识过滤器所属的类，在此实例中为类 `userweb`。

有关此 `filter` 子句任务的说明，请参见第 47 页中的“如何在 IPQoS 配置文件中定义过滤器”。

#### 5 以调用 `dscpmk` 标记器来开始 `action` 语句。

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

`module dscpmk` 调用标记器模块 `dscpmk`。

`name markAF12` 为 `action` 语句提供名称 `markAF12`。

先前定义的类 `userweb` 包括 `next_action markAF12` 语句。分类器结束处理之后，此语句将通信流发送到 `markAF12 action` 语句。

#### 6 定义标记器用于处理通信流的参数。

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

<code>global_stats FALSE</code>	对 <code>markAF12</code> 标记器 <code>action</code> 语句启用统计信息收集。但是，由于 <code>enable_stats</code> 的值为 <code>FALSE</code> ，因此不会收集统计信息。
<code>dscp_map{0-63:12}</code>	将 DSCP 12 指定给通信类 <code>userweb</code> 的包头，标记器当前正在处理此通信类。
<code>next_action continue</code>	指示不需要对通信类 <code>userweb</code> 的包进行进一步处理，并指示可以将这些包返回到网络流中。

DSCP 12 指示标记器将 `dscp` 映射中的所有项都设置为十进制值 12（二进制值 001100）。此代码点指示 `userweb` 通信类的包遵守 AF12 单跳行为。AF12 保证 DS 字段中的 DSCP 为 12 的所有包都拥有中丢弃率、高优先级的服务。

## 7 完成 IPQoS 配置文件后，应用配置。

- 另请参见
- 要从应用程序中为通信流添加类和其他配置，请参阅第 56 页中的“[如何为应用服务器配置 IPQoS 配置文件](#)”。
  - 要在路由器上配置单跳行为，请参阅第 63 页中的“[如何在启用了 IPQoS 的网络中配置路由器](#)”。
  - 要激活 IPQoS 配置文件，请参阅第 66 页中的“[如何将新配置应用于 IPQoS 内核模块](#)”。

# 为应用服务器创建 IPQoS 配置文件

本节介绍如何为可向客户提供主要应用程序的应用服务器创建配置文件。以下过程以图 2-4 中的 BigAPPS 服务器为例。

以下配置文件定义了 BigAPPS 服务器的 IPQoS 活动。此服务器为客户托管 FTP、电子邮件 (SMTP) 以及网络新闻 (NNTP)。

示例 3-3 应用服务器的 IPQoS 配置文件样例

```
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
  }
}
```

示例 3-3 应用服务器的 IPQoS 配置文件样例 (续)

```

    }
    class {
        name ftp
        next_action meterftp
    }
    filter {
        name smtpout
        sport smtp
        class smtp
    }
    filter {
        name newsout
        sport nntp
        class news
    }
    filter {
        name ftpout
        sport ftp
        class ftp
    }
    filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
action {
    module dscpmk
    name markAF13
    params {
        global_stats FALSE
        dscp_map{0-63:14}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
action {
    module tokenmt
    name meterftp
    params {
        committed_rate 50000000
        committed_burst 50000000
        red_action_name AF31
        green_action_name markAF22
        global_stats TRUE
    }
}
}

```

示例 3-3 应用服务器的 IPQoS 配置文件样例 (续)

```

action {
  module dscpmk
  name markAF31
  params {
    global_stats TRUE
    dscp_map{0-63:26}
    next_action continue
  }
}
action {
  module dscpmk
  name markAF22
  params {
    global_stats TRUE
    dscp_map{0-63:20}
    next_action continue
  }
}
}

```

## ▼ 如何为应用服务器配置 IPQoS 配置文件

- 1 登录到启用了 IPQoS 的应用服务器，并创建扩展名为 `.qos` 的新 IPQoS 配置文件。

例如，可以为应用服务器创建 `/var/ipqos/BigAPPS.qos` 文件。请以下列必需的短语开头以开始调用 `ipgpc` 分类器的 `action` 语句：

```
fmt_version 1.0
```

```

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

```

有关开始 `action` 语句的说明，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

- 2 创建类以选择 BigAPPS 服务器上三个应用程序产生的通信。

在开始 `action` 语句之后添加类定义。

```

class {
  name smtp
  enable_stats FALSE
  next_action markAF13
}
class {
  name news
  next_action markAF21
}
class {

```



```

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

```

name smtp	创建名为 <code>smtp</code> 的类，此类包括要由 SMTP 应用程序处理的电子邮件通信流。
enable_stats FALSE	用于针对 <code>smtp</code> 类启用统计信息收集。但是，由于 <code>enable_stats</code> 的值为 <code>FALSE</code> ，因此不会为此类收集统计信息。
next_action markAF13	指示 <code>ipgpc</code> 模块在 <code>ipgpc</code> 完成处理之后，将 <code>smtp</code> 类的包传递到 <code>markAF13</code> action 语句。
name news	创建名为 <code>news</code> 类，此类包括要由 NNTP 应用程序处理的网络新闻通信流。
next_action markAF21	指示 <code>ipgpc</code> 模块在 <code>ipgpc</code> 完成处理之后，将 <code>news</code> 类的包传递到 <code>markAF21</code> 操作语句。
name ftp	创建名为 <code>ftp</code> 的类，此类包括要由 FTP 应用程序处理的传出通信。
enable_stats TRUE	用于针对 <code>ftp</code> 类启用统计信息收集。
next_action meterftp	指示 <code>ipgpc</code> 模块在 <code>ipgpc</code> 完成处理之后，将 <code>ftp</code> 类的包传递到 <code>meterftp</code> action 语句。

有关定义类的更多信息，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。

### 3 定义 filter 子句以选择属于在步骤 2 中定义的类的通信。

```

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
}
}

```

<code>name smtpout</code>	为过滤器提供名称 <code>smtpout</code> 。
<code>sport smtp</code>	选择源端口 25 传出的通信，此端口是众所周知的用于 <code>sendmail</code> (SMTP) 应用程序的端口。
<code>class smtp</code>	标识过滤器所属的类，在此实例中为类 <code>smtp</code> 。
<code>name newsout</code>	为过滤器提供名称 <code>newsout</code> 。
<code>sport nntp</code>	选择名称为 <code>nntp</code> 的源端口传出的通信，此名称是众所周知的用于网络新闻 (NNTP) 应用程序的端口名称。
<code>class news</code>	标识过滤器所属的类，在此实例中为类 <code>news</code> 。
<code>name ftpout</code>	为过滤器提供名称 <code>ftpout</code> 。
<code>sport ftp</code>	选择源端口 21 传出的控制数据，此端口号是众所周知的用于 FTP 通信的端口号。
<code>name ftpdata</code>	为过滤器提供名称 <code>ftpdata</code> 。
<code>sport ftp-data</code>	选择源端口 20 传出的通信，此端口号是众所周知的用于 FTP 数据通信的端口号。
<code>class ftp</code>	标识 <code>ftpout</code> 和 <code>ftpdata</code> 过滤器所属的类，在此实例中为 <code>ftp</code> 。

- 另请参见
- 要定义过滤器，请参阅第 47 页中的“如何在 IPQoS 配置文件中定义过滤器”。
  - 要定义应用程序通信的转发行为，请参阅第 58 页中的“如何在 IPQoS 配置文件中为应用程序通信配置转发”。
  - 要通过使用计量模块配置流控制，请参阅第 60 页中的“如何在 IPQoS 配置文件中配置流控制”。
  - 要配置流记帐，请参阅第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”。

## ▼ 如何在 IPQoS 配置文件中为应用程序通信配置转发

以下过程说明如何为应用程序通信配置转发。在此过程中，您将优先级可能低于网络中其他通信的应用程序通信类定义单跳行为。这些步骤将继续构建示例 3-3 中的 `/var/ipqos/BigAPPS.qos` 文件。

**开始之前** 此过程假设您具有已为要标记的应用程序定义类和过滤器的 IPQoS 配置文件。

- 1 打开为应用服务器创建的 IPQoS 配置文件，并定位到最后一条 `filter` 子句的结尾。在 `/var/ipqos/BigAPPS.qos` 文件中，最后一个过滤器为：

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

```
    }
}
```

## 2 按以下方式调用标记器：

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

module dscpmk 调用标记器模块 dscpmk。

name markAF13 为 action 语句提供名称 markAF13。

## 3 定义要在电子邮件通信流上标记的单跳行为。

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

global\_stats FALSE 对 markAF13 标记器 action 语句启用统计信息收集。但是，由于 enable\_stats 的值为 FALSE，因此不会收集统计信息。

dscp\_map{0-63:14} 将 DSCP 14 指定给通信类 smtp 的包头，标记器当前正在处理此通信类。

next\_action continue 指示不需要对通信类 smtp 的包进行进一步处理。这样，可以将这些包返回到网络流中。

DSCP 为 14 指示标记器将 dscp 映射中的所有项都设置为十进制值 14（二进制值 001110）。DSCP 为 14 将设置 AF13 单跳行为。标记器在 DS 字段中使用 DSCP 14 来标记属于 smtp 通信类的包。

AF13 针对 DSCP 为 14 的所有包指定高丢弃率的优先级。但是，由于 AF13 还确保优先级为类 1，因此路由器仍保证其队列中的传出电子邮件通信具有高优先级。有关可能的 AF 代码点表的信息，请参阅表 6-2。

## 4 添加标记器 action 语句以便为网络新闻通信定义单跳行为：

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

name markAF21 为 action 语句提供名称 markAF21。

dscp\_map{0-63:18} 将 DSCP 18 指定给通信类 nntp 的包头，标记器当前正在处理此通信类。

DSCP 为 18 指示标记器将 `dscp` 映射中的所有项都设置为十进制值 18（二进制值 010010）。DSCP 为 18 将设置 AF21 单跳行为。标记器在 DS 字段中使用 DSCP 18 来标记属于 news 通信类的包。

AF21 保证 DSCP 为 18 的所有包都拥有低丢弃率的优先级，但优先级仅为类 2。因此，丢弃网络新闻通信的可能性很低。

- 另请参见
- 要添加 Web 服务器的配置信息，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。
  - 要通过使用计量模块配置流控制，请参阅第 60 页中的“如何在 IPQoS 配置文件中配置流控制”。
  - 要配置流记帐，请参阅第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”。
  - 要在路由器上配置转发行为，请参阅第 63 页中的“如何在启用了 IPQoS 的网络中配置路由器”。
  - 要激活 IPQoS 配置文件，请参阅第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。

## ▼ 如何在 IPQoS 配置文件中配置流控制

要控制将特定通信流释放到网络的速率，必须为计量器定义参数。您可以在 IPQoS 配置文件中使以下两个计量器模块之一：`tokenmt` 或 `tswtclmt`。

以下过程将继续构建示例 3-3 中应用服务器的 IPQoS 配置文件。在此过程中，您不仅要配置计量器，还要配置在计量器 `action` 语句中调用的两种标记器操作。

**开始之前** 以下步骤假设已为要进行流控制的应用程序定义了类和过滤器。

### 1 打开为应用服务器创建的 IPQoS 配置文件。

在 `/var/ipqos/BigAPPS.qos` 文件中，可以从以下标记器操作之后开始：

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

### 2 创建计量器 `action` 语句以便对 ftp 类的通信进行流控制。

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

`module tokenmt` 调用 `tokenmt` 计量器。

name meterftp 为 action 语句提供名称 meterftp。

### 3 添加参数以配置计量器的速率。

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

committed\_rate 50000000 指定 ftp 类的通信的传输速率为 50,000,000 bps。

committed\_burst 50000000 指定 ftp 类的通信的突发大小为 50,000,000 位。

有关 tokenmt 参数的解释，请参阅第 80 页中的“将 tokenmt 配置为双速率计量器”。

### 4 添加参数以配置通信一致性优先级：

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

red\_action\_name markAF31 指示当 ftp 类的通信流超过承诺速率时，将包发送到 markAF31 标记器 action 语句。

green\_action\_name markAF22 指示当类 ftp 的通信流符合承诺速率时，将包发送到 markAF22 操作语句。

global\_stats TRUE 用于针对 ftp 类启用计量统计信息。

有关通信一致性的更多信息，请参见第 79 页中的“计量器模块”。

### 5 添加标记器 action 语句以便为属于 ftp 类的非一致通信流指定单跳行为。

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

module dscpmk 调用标记器模块 dscpmk。

name markAF31 为 action 语句提供名称 markAF31。

global\_stats TRUE 用于针对 ftp 类启用统计信息。

dscp\_map{0-63:26} 当 ftp 类的通信超过承诺速率时，将 DSCP 26 指定给该类的包头。

next\_action continue 指示不需要对通信类 ftp 的包进行进一步处理。这样，可以将这些包返回到网络流中。

DSCP 26 指示标记器将 dscp 映射中的所有项都设置为十进制值 26（二进制值 011010）。DSCP 26 设置 AF31 单跳行为。标记器使用 DS 字段中的 DSCP 26 来标记属于 ftp 通信类的包。

AF31 保证 DSCP 为 26 的所有包都拥有低丢弃率的优先级，但优先级仅为类 3。因此，丢弃非一致的 FTP 通信的可能性很低。有关可能的 AF 代码点表的信息，请参阅表 6-2。

## 6 添加标记器 action 语句以便为符合承诺速率的 ftp 通信流指定单跳行为。

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

name markAF22 为 marker 操作提供名称 markAF22。

dscp\_map{0-63:20} 当 ftp 通信符合配置的速率时，将 DSCP 20 指定给该类的包头。

DSCP 20 指示标记器将 dscp 映射中的所有项都设置为十进制值 20（二进制值 010100）。DSCP 20 设置 AF22 单跳行为。标记器使用 DS 字段中的 DSCP 20 来标记属于 ftp 通信类的包。

AF22 保证 DSCP 为 20 的所有包都拥有中丢弃率的优先级，优先级为类 2。因此，在由 IPQoS 系统同时释放的流中，一致性 FTP 通信具有中丢弃率的优先级。但是，路由器会为具有中丢弃率的类 1 优先级标记或更高优先级的通信类提供更高的转发优先级。有关可能的 AF 代码点表的信息，请参阅表 6-2。

## 7 将为应用服务器创建的 DSCP 添加到 Diffserv 路由器上的相应文件中。

- 另请参见
- 要激活 IPQoS 配置文件，请参阅第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。
  - 要添加 Web 服务器的配置信息，请参阅第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”。
  - 要配置流记帐，请参阅第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”。
  - 要在路由器上配置转发行为，请参阅第 63 页中的“如何在启用了 IPQoS 的网络中配置路由器”。

# 在路由器上提供区分服务

要提供真正的区分服务，必须在网络拓扑中添加可识别 Diffserv 的路由器，如第 24 页中的“Diffserv 网络的硬件策略”中所述。在路由器上配置 Diffserv 以及更新此路由器的文件的实际步骤不在本指南叙述的范围之内。

本节介绍在网络和 Diffserv 路由器上的各种启用了 IPQoS 的系统间协调转发信息的常规步骤。

## ▼ 如何在启用了 IPQoS 的网络中配置路由器

以下过程以图 2-4 中的拓扑为例。

**开始之前** 以下过程假设您已经通过执行本章中的上述任务在网络中配置了 IPQoS 系统。

- 1 查看网络中所有启用了 IPQoS 的系统的配置文件。
- 2 标识在 QoS 各种策略中使用的每个代码点。

列出代码点以及这些代码点所应用到的系统和类。下表列出了可能已经使用相同代码点的区域。这种做法是可以接受的。但是，应该在 IPQoS 配置文件中提供其他条件（例如 precedence 选定器），以便确定具有相同标记的类的优先级。

例如，对于本章过程中使用的网络样例，可能会构造以下代码点表。

系统	类	PHB	DS 代码点
Goldweb	video	EF	46 (101110)
Goldweb	goldweb	AF11	10 (001010)
Userweb	webout	AF12	12 (001100)
BigAPPS	smtp	AF13	14 (001110)
BigAPPS	news	AF18	18 (010010)
BigAPPS	ftp 一致性通信	AF22	20 (010100)
BigAPPS	ftp 非一致的通信	AF31	26 (011010)

- 3 将网络 IPQoS 配置文件中的代码点添加到 Diffserv 路由器上的相应文件中。

提供的代码点应有助于配置路由器的 Diffserv 调度机制。请参阅路由器制造商提供的文档和 Web 站点以获得相关的说明。





## 启动和维护 IPQoS ( 任务 )

---

本章介绍激活 IPQoS 配置文件和记录 IPQoS 相关事件的任务。本章包含以下主题：

- 第 65 页中的“管理 IPQoS ( 任务列表 )”
- 第 66 页中的“应用 IPQoS 配置”
- 第 67 页中的“启用 IPQoS 消息的 syslog 日志”
- 第 68 页中的“对 IPQoS 错误消息进行故障排除”

---

注 - 未来发行版中可能会删除 IPQoS 功能。鼓励用户转而使用 `dladm`、`flowadm` 和相关命令，这些命令可支持类似的带宽资源控制功能。有关更多信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

---

### 管理 IPQoS ( 任务列表 )

本节列出用于在 Oracle Solaris 系统上启动和维护 IPQoS 的任务组。执行这些任务之前，您必须具有已完成的 IPQoS 配置文件，如第 41 页中的“在 IPQoS 配置文件中定义 QoS 策略 ( 任务列表 )”中所述。

下表列出并描述了这些任务，并且包含指向详细说明如何完成这些任务的章节的链接。

任务	说明	参考
1. 在系统上配置 IPQoS。	在系统上使用 <code>ipqosconf</code> 命令来激活 IPQoS 配置文件。	第 66 页中的“如何将新配置应用于 IPQoS 内核模块”
2. 每次引导系统之后，使 Oracle Solaris 启动脚本应用调试过的 IPQoS 配置文件。	确保每次重新引导系统时都应用 IPQoS 配置。	第 67 页中的“如何确保每次重新引导系统之后都应用 IPQoS 配置”。
3. 启用 IPQoS 的 syslog 日志记录。	添加一个项以启用 IPQoS 消息的 syslog 日志记录。	第 67 页中的“如何在引导过程中启用 IPQoS 消息的日志记录”。

任务	说明	参考
4. 解决发生的任何 IPQoS 问题。	使用错误消息对 IPQoS 的问题进行故障排除。	请参阅表 4-1 中的错误消息。

## 应用 IPQoS 配置

您可以使用 `ipqosconf` 命令激活 IPQoS 的配置并对其执行其他操作。

### ▼ 如何将新配置应用于 IPQoS 内核模块

您可以使用 `ipqosconf` 命令读取 IPQoS 配置文件和配置 UNIX 内核中的 IPQoS 模块。以下过程以第 43 页中的“为 Web 服务器创建 IPQoS 配置文件”中创建的文件 `/var/ipqos/Goldweb.qos` 为例。有关详细信息，请参阅 `ipqosconf(1M)` 手册页。

#### 1 成为管理员。

有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

#### 2 应用新配置。

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

`ipqosconf` 将指定的 IPQoS 配置文件中的信息写入 Oracle Solaris 内核中的 IPQoS 模块。在此示例中，会将 `/var/ipqos/Goldweb.qos` 中的内容应用于当前的 Oracle Solaris 内核。

---

注 – 使用 `-a` 选项应用 IPQoS 配置文件时，文件中的操作只对当前会话有效。

---

#### 3 测试和调试新的 IPQoS 配置。

使用 UNIX 实用程序跟踪 IPQoS 行为并收集有关 IPQoS 实现的统计信息。此信息有助于您确定配置是否按预期执行。

- 另请参见
- 要查看有关 IPQoS 模块工作方式的统计信息，请参阅第 76 页中的“收集统计信息”。
  - 要记录 `ipqosconf` 消息，请参阅第 67 页中的“启用 IPQoS 消息的 `syslog` 日志”。
  - 要确保每次引导系统之后都应用当前的 IPQoS 配置，请参阅第 67 页中的“如何确保每次重新引导系统之后都应用 IPQoS 配置”。

## ▼ 如何确保每次重新引导系统之后都应用 IPQoS 配置

您应该明确地使 IPQoS 配置在每次重新引导系统之后都能应用。否则，当前配置将只在下次重新引导系统之前有效。当 IPQoS 在系统上正常运行时，请执行以下操作以使此配置在每次重新引导系统之后都能应用。

### 1 成为管理员。

有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

### 2 测试内核模块中是否存在某个 IPQoS 配置。

```
# ipqosconf -l
```

如果配置已经存在，ipqosconf 便会在屏幕上显示此配置信息。如果没有显示输出信息，请应用配置，如第 66 页中的“如何将新配置应用于 IPQoS 内核模块”中所述。

### 3 确保每次重新引导 IPQoS 系统时都应用现有的 IPQoS 配置。

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

使用 -c 选项，使当前的 IPQoS 配置在引导时配置文件 /etc/inet/ipqosinit.conf 中表示。

## 启用 IPQoS 消息的 syslog 日志

要记录 IPQoS 引导时消息，您需要按照以下过程所示修改 /etc/syslog.conf 文件。

## ▼ 如何在引导过程中启用 IPQoS 消息的日志记录

### 1 成为管理员。

有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

### 2 打开 /etc/syslog.conf 文件。

### 3 添加以下文本，将其作为文件中的最后一项。

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

在列间使用 Tab 键而不是空格键。

此项将所有由 IPQoS 生成的引导时消息都记录到 /var/adm/messages 文件中。

### 4 重新引导系统以应用这些消息。

## 示例 4-1 来自 /var/adm/messages 的 IPQoS 输出

在系统重新引导之后查看 /var/adm/messages 时，输出信息可能包含与以下内容相似的 IPQoS 日志消息。

```
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.
```

您还可能在 IPQoS 系统的 /var/adm/messages 文件中看到与以下内容相似的 IPQoS 错误消息。

```
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.
```

有关这些错误消息的说明，请参见表 4-1。

## 对 IPQoS 错误消息进行故障排除

本节用一张表说明 IPQoS 生成的错误消息以及可能的解决方法。

表 4-1 IPQoS 错误消息

错误消息	说明	解决方案
Undefined action in parameter <i>parameter-name's</i> action <i>action-name</i>	在 IPQoS 配置文件中，不存在 <i>parameter-name</i> 中指定的操作名称。	创建操作。或者参阅参数中其他现有的操作。
action <i>action-name</i> involved in cycle	在 IPQoS 配置文件中， <i>action-name</i> 是操作循环的一部分，IPQoS 不允许出现此情况。	确定操作循环，然后从 IPQoS 配置文件中删除其中的一个循环引用。
Action <i>action-name</i> isn't referenced by any other actions	在 IPQoS 配置中定义的其他操作未引用非 ipgpc 操作的定义，IPQoS 不允许出现此情况。	删除未引用的操作。或者，使另一个操作引用当前未引用的操作。
Missing/Invalid config file <i>fmt_version</i>	未将配置文件的格式指定为文件中的第一项，IPQoS 要求执行此操作。	添加格式版本，如第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”中所述。
Unsupported config file format <i>version</i>	IPQoS 不支持在配置文件中指定的格式版本。	将格式版本改为 <i>fmt_version 1.0</i> ，从 Solaris 9 9/02 发行版的 IPQoS 起，需要该版本。
No ipgpc action defined.	未在配置文件中为 ipgpc 分类器定义操作，IPQoS 要求执行此操作。	为 ipgpc 定义一个操作，如第 45 页中的“如何创建 IPQoS 配置文件并定义通信类”中所述。

表 4-1 IPQoS 错误消息 (续)

错误消息	说明	解决方案
Can't commit a null configuration	运行 <code>ipqosconf -c</code> 以提交配置时, 配置为空, IPQoS 不允许出现此情况。	确保在尝试提交配置之前应用配置文件。有关说明, 请参见第 66 页中的“如何将新配置应用于 IPQoS 内核模块”。
Invalid CIDR mask on line <i>line-number</i>	在配置文件中, 将超出 IP 地址有效范围的 CIDR 掩码用作 IP 地址的一部分。	更改掩码值, 使其位于 1-32 范围内 (对于 IPv4) 或者 1-128 范围内 (对于 IPv6)。
Address masks aren't allowed for host names line <i>line-number</i>	在配置文件中, 为主机名定义了 CIDR 掩码, IPQoS 不允许出现此情况。	删除掩码或将主机名更改为 IP 地址。
Invalid module name line <i>line-number</i>	在配置文件中, 操作语句中指定的模块名称无效。	检查模块名称的拼写。有关 IPQoS 模块列表, 请参阅表 6-5。
ipgpc action has incorrect name line <i>line-number</i>	在配置文件中指定给 <code>ipgpc</code> 操作的名称不是要求的 <code>ipgpc.classify</code> 。	将操作重命名为 <code>ipgpc.classify</code> 。
Second parameter clause not supported line <i>line-number</i>	在配置文件中, 为单个操作指定了两条参数子句, IPQoS 不允许出现此情况。	将此操作的所有参数合并到单个参数子句中。
Duplicate named action	在配置文件中, 为两个操作指定了相同的名称。	重命名或删除其中一个操作。
Duplicate named filter/class in action <i>action-name</i>	为同一操作中的两个过滤器或类指定了相同的名称, IPQoS 配置文件中不允许出现此情况。	重命名或删除其中一个过滤器或类。
Undefined class in filter <i>filter-name</i> in action <i>action-name</i>	在配置文件中, 过滤器引用操作中未定义的类。	创建类, 或者将该过滤器引用更改为已经存在的类。
Undefined action in class <i>class-name</i> action <i>action-name</i>	类引用配置文件中未定义的操作。	创建操作, 或者将此引用更改为已经存在的操作。
Invalid parameters for action <i>action-name</i>	在配置文件中, 其中有一个参数无效。	有关由指定操作调用的模块的信息, 请参阅第 77 页中的“IPQoS 体系结构和 Diffserv 模型”中的模块项。或者, 请参阅 <code>ipqosconf(1M)</code> 手册页。
Mandatory parameter missing for action <i>action-name</i>	在配置文件中, 没有为操作定义必要的参数。	有关由指定操作调用的模块的信息, 请参阅第 77 页中的“IPQoS 体系结构和 Diffserv 模型”中的模块项。或者, 请参阅 <code>ipqosconf(1M)</code> 手册页。
Max number of classes reached in ipgpc	指定的类的数目超出 IPQoS 配置文件的 <code>ipgpc</code> 操作中允许的类数目。最大数目为 10007。	查看配置文件, 删除不需要的类。或者, 可通过将 <code>ipgpc_max_classesclass-number</code> 条目添加到 <code>/etc/system</code> 文件来增加类的最大数目。

表 4-1 IPQoS 错误消息 (续)

错误消息	说明	解决方案
Max number of filters reached in action ipgpc	指定的过滤器的数目超出 IPQoS 配置文件的 ipgpc 操作中允许的过滤器数目。最大数目为 10007。	查看配置文件，删除不需要的过滤器。或者，可通过将 ipgpc_max_filters <code>filter-number</code> 条目添加到 <code>/etc/system</code> 文件来增加过滤器的最大数目。
Invalid/missing parameters for filter <code>filter-name</code> in action ipgpc	在配置文件中，过滤器 <code>filter-name</code> 具有无效或缺失的参数。	有关有效参数的列表，请参阅 ipqosconf(1M) 手册页。
Name not allowed to start with '!', line <code>line-number</code>	操作、过滤器或类名称的开头有叹号 (!)，IPQoS 文件中不允许出现此情况。	删除叹号，或者重命名操作、类或过滤器。
Name exceeds the maximum name length line <code>line-number</code>	在配置文件中为操作、类或过滤器定义的名称的长度超过了最大长度 23 个字符。	为操作、类或过滤器指定较短的名称。
Array declaration line <code>line-number</code> is invalid	在配置文件中，行 <code>line-number</code> 上的参数的数组声明无效。	有关数组无效的 action 语句所调用的数组声明的正确语法，请参阅第 77 页中的“IPQoS 体系结构和 Diffserv 模型”。或者，请参阅 ipqosconf(1M) 手册页。
Quoted string exceeds line, <code>line-number</code>	字符串的终止引号没有与其位于同一行上，而在配置文件中要求位于同一行。	确保在配置文件中，引用的字符串在同一行开始和结束。
Invalid value, line <code>line-number</code>	对于参数而言，不支持在配置文件的 <code>line-number</code> 上指定的值。	有关 action 语句所调用的模块的可接受值，请参阅第 77 页中的“IPQoS 体系结构和 Diffserv 模型”中的模块说明。或者，请参阅 ipqosconf(1M) 手册页。
Unrecognized value, line <code>line-number</code>	对于参数而言，配置文件的 <code>line-number</code> 上的值不是受支持的枚举值。	检查枚举值对于参数是否正确。有关行号无法识别的 action 语句所调用的模块的说明，请参阅第 77 页中的“IPQoS 体系结构和 Diffserv 模型”。或者，请参阅 ipqosconf(1M) 手册页。
Malformed value list line <code>line-number</code>	在配置文件的 <code>line-number</code> 上指定的枚举不符合规范语法。	有关值列表格式错误的 action 语句所调用的模块的正确语法，请参阅第 77 页中的“IPQoS 体系结构和 Diffserv 模型”中的模块说明。或者，请参阅 ipqosconf(1M) 手册页。
Duplicate parameter line <code>line-number</code>	在 <code>line-number</code> 上指定了重复的参数，配置文件中不允许出现此情况。	删除其中一个重复的参数。
Invalid action name line <code>line-number</code>	为配置文件的 <code>line-number</code> 上的操作指定的名称使用了预定义名称 "continue" 或 "drop"。	重命名操作以使用此操作不使用预先定义的名称。
Failed to resolve src/dst host name for filter at line <code>line-number</code> , ignoring filter	ipqosconf 不能解析在配置文件中为给定过滤器定义的源地址或目标地址。因此，忽略了过滤器。	如果过滤器很重要，则稍后尝试应用此配置。

表 4-1 IPQoS 错误消息 (续)

错误消息	说明	解决方案
Incompatible address version line <i>line-number</i>	<i>line-number</i> 处地址的 IP 版本与先前指定的 IP 地址或 <code>ip_version</code> 参数不兼容。	更改两个冲突的项使其兼容。
Action at line <i>line-number</i> has the same name as currently installed action, but is for a different module	尝试更改系统 IPQoS 配置中已存在的操作的模块，不允许执行此操作。	在应用新配置之前刷新当前配置。





## 使用流记帐和统计信息收集功能（任务）

---

本章介绍如何获取有关由 IPQoS 系统处理的流量的记帐和统计信息。本章包含以下主题：

- 第 73 页中的“设置流记帐（任务列表）”
- 第 74 页中的“记录有关通信流量的信息”
- 第 76 页中的“收集统计信息”

---

注 – 未来发行版中可能会删除 IPQoS 功能。鼓励用户转而使用 `dladm`、`flowadm` 和相关命令，这些命令可支持类似的带宽资源控制功能。有关更多信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

---

### 设置流记帐（任务列表）

以下任务列表列出了使用 `flowacct` 模块获取有关通信流的信息的一般任务。此列表还链接到执行这些任务的过程。

任务	说明	参考
1. 创建包含通信流的记帐信息的文件。	使用 <code>acctadm</code> 命令创建一个文件，用来包含 <code>flowacct</code> 处理所得的结果。	第 74 页中的“如何为流记帐数据创建文件”
2. 在 IPQoS 配置文件中定义 <code>flowacct</code> 参数。	为 <code>timer</code> 、 <code>timeout</code> 和 <code>max_limit</code> 参数定义值。	第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”

## 记录有关通信流量的信息

您可以使用 IPQoS `flowacct` 模块来收集有关通信流的信息。例如，您可以收集源地址和目标地址、流中包的数量及类似数据。积累和记录有关流的信息的过程称为**流记帐**。

对属于特定类的通信进行流记帐的结果会记录在一个**流记录表**中。每个流记录都包括一系列属性。这些属性包含有关一个时间间隔内特定类的通信流量的数据。有关 `flowacct` 属性的列表，请参阅表 6-4。

流记帐对于按照客户机服务级别协议 (service-level agreement, SLA) 中的定义为这些客户机记帐特别有用。您还可以使用流记帐来获取关键应用程序的流统计信息。本节包含使用 `flowacct` 和 Oracle Solaris 扩展记帐功能来获取有关通信流数据的任务。

以下信息将在除本章以外的其他章节中介绍：

- 有关在 IPQoS 配置文件中创建 `flowacct` 的操作语句的说明，请参阅第 60 页中的“如何在 IPQoS 配置文件中配置流控制”。
- 要了解 `flowacct` 的工作原理，请参阅第 78 页中的“分类器模块”。
- 有关技术信息，请参阅 `flowacct(7ipp)` 手册页。

### ▼ 如何为流记帐数据创建文件

在向 IPQoS 配置文件添加 `flowacct` 操作之前，必须通过 `flowacct` 模块为流记录创建文件。为此，可以使用 `acctadm` 命令。`acctadm` 可以将基本属性或扩展属性记录到该文件中。表 6-4 中列出了所有的 `flowacct` 属性。有关 `acctadm` 的详细信息，请参阅 `acctadm(1M)` 手册页。

#### 1 成为管理员。

有关更多信息，请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

#### 2 创建基本流记帐文件。

以下示例说明如何为示例 3-1 中配置的高级 Web 服务器创建基本流记帐文件。

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

<code>acctadm -e</code>	调用带有 <code>-e</code> 选项的 <code>acctadm</code> 。 <code>-e</code> 选项会启用后跟的参数。
<code>basic</code>	声明仅在文件中记录 <code>flowacct</code> 的八个基本属性的数据。
<code>/var/ipqos/goldweb/account.info</code>	指定用于包含 <code>flowacct</code> 所获流记录的文件的限定路径名。
<code>flow</code>	指示 <code>acctadm</code> 启用流记帐。

### 3 通过键入不带参数的 `acctadm`，查看有关 IPQoS 系统上的流记帐的信息。

`acctadm` 会生成以下输出：

```
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
```

所有项（最后四项除外）都可用于 Oracle Solaris 资源管理器功能。下表介绍了特定于 IPQoS 的项。

项	说明
Flow accounting: active	指示流记帐已打开。
Flow accounting file: /var/ipqos/goldweb/account.info	指定当前流记帐文件的名称。
Tracked flow resources: basic	指示仅跟踪基本流属性。
Untracked flow resources: dsfield,ctime,lseen,projid,uid	列出文件中未被跟踪的 <code>flowacct</code> 属性。

### 4 （可选）将扩展属性添加到记帐文件中。

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

### 5 （可选）返回，仅记录记帐文件中的基本属性。

```
# acctadm -d extended -e basic -f /var/ipqos/goldweb/account.info
-d 选项可禁用扩展记帐。
```

### 6 查看流记帐文件的内容。

有关查看流记帐文件内容的说明位于《Oracle Solaris 11.1 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的“libexacct 的 Perl 接口”。

- 另请参见
- 有关扩展记帐功能的详细信息，请参阅《Oracle Solaris 11.1 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的第 4 章“扩展记帐（概述）”。
  - 要在 IPQoS 配置文件中定义 `flowacct` 参数，请参阅第 51 页中的“如何在 IPQoS 配置文件中为类启用记帐”。
  - 要输出使用 `acctadm` 创建的文件中的数据，请参阅《Oracle Solaris 11.1 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的“libexacct 的 Perl 接口”。

## 收集统计信息

可以使用 `kstat` 命令生成 IPQoS 模块的统计信息。使用以下语法：

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

可以指定任何有效的 IPQoS 模块名称，如表 6-5 中所示。例如，要查看由 `dscpmk` 标记器生成的统计信息，请使用以下形式的 `kstat`：

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

有关详细技术信息，请参阅 `kstat(1M)` 手册页。

### 示例 5-1 IPQoS 的 `kstat` 统计信息

以下是运行 `kstat` 以获取有关 `flowacct` 模块的统计信息的可能结果示例。

```
# kstat -m flowacct
module: flowacct                instance: 3
name: Flowacct statistics        class:  flacct
      bytes_in_tbl               84
      crtime                     345728.504106363
      epackets                   0
      flows_in_tbl               1
      nbytes                     84
      npackets                   1
      snaptime                   345774.031843301
      usedmem                    256
```

`class: flacct` 指定通信流量所属的类的名称，在本示例中为 `flacct`。

`bytes_in_tbl` 流表中的总字节数。总字节数是指当前驻留在流表中的所有流记录的字节总和。此流表的总字节数为 84。如果表中没有流，则 `bytes_in_tbl` 的值为 0。

`crtime` 上次创建此 `kstat` 输出的时间。

`epackets` 在处理过程中导致错误的包的数量，在本示例中为 0。

`flows_in_tbl` 流表中的流记录的数量，在本示例中为 1。该表中没有记录时，`flows_in_tbl` 的值为 0。

`nbytes` 此 `flowacct` 操作实例发现的总字节数，在本示例中为 84。该值包括流表中当前的字节，还包括已超时而不再在流表中的字节。

`npackets` 此 `flowacct` 操作实例发现的总包数，在本示例中为 1。`npackets` 包括当前位于流表中的包。`npackets` 还包括已超时（不再位于流表中）的包。

`usedmem` 由此 `flowacct` 实例维护的流表所使用的内存（以字节为单位）。在本示例中，`usedmem` 值为 256。当流表没有任何流记录时，`usedmem` 的值为 0。

## IPQoS 的详细介绍（参考信息）

---

本章包含可提供有关以下 IPQoS 主题的详细信息的参考资料：

- 第 77 页中的“IPQoS 体系结构和 Diffserv 模型”
- 第 88 页中的“IPQoS 配置文件”
- 第 91 页中的“ipqosconf 配置实用程序”

有关概述信息，请参阅第 1 章，IPQoS 介绍（概述）。有关规划信息，请参阅第 2 章，规划启用了 IPQoS 的网络（任务）。有关 IPQoS 的配置过程，请参阅第 3 章，创建 IPQoS 配置文件（任务）。

---

注 - 未来发行版中可能会删除 IPQoS 功能。鼓励用户转而使用 `dladm`、`flowadm` 和相关命令，这些命令可支持类似的带宽资源控制功能。有关更多信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

---

## IPQoS 体系结构和 Diffserv 模型

本节介绍 IPQoS 体系结构以及 IPQoS 如何实现区分服务 (differentiated service, Diffserv) 模型，在 RFC 2475, *An Architecture for Differentiated Services* (<http://www.ietf.org/rfc/rfc2475.txt?number=2475>) (RFC 2475, 区分服务的体系结构) 中定义了此模型。IPQoS 中包括以下 Diffserv 模型元素：

- 分类器
- 计量器
- 标记器

此外，IPQoS 还包括流记帐模块和用于虚拟局域网 (virtual local area network, VLAN) 设备的 `dlcosmk` 标记器。

## 分类器模块

在 Diffserv 模型中，**分类器**负责将选定的通信流组织到应用不同服务级别的组中。RFC 2475 中定义的分类器最初是为边界路由器设计的。相反，IPQoS 分类器 `ipgpc` 是为了处理本地网络内部的主机上的通信流而设计的。因此，同时包含 IPQoS 系统和 Diffserv 路由器的网络可以提供更程度的区分服务。有关 `ipgpc` 的技术说明，请参阅 `ipgpc(7ipp)` 手册页。

`ipgpc` 分类器具有以下功能：

1. 选择满足条件的通信流，该条件在启用了 IPQoS 的系统中的 IPQoS 配置文件中指定。  
QoS 策略可以定义各种必须出现在包头中的条件。这些条件称为**选定器**。`ipgpc` 分类器将这些选定器与 IPQoS 系统接收的包头进行比较，然后，`ipgpc` 选择所有匹配的包。
2. 将包流**分类**（即具有相同特征的网络通信流量），如 IPQoS 配置文件中所定义。
3. 检查包的区分服务 (differentiated service, DS) 字段中的值，确定是否存在区分服务代码点 (differentiated service codepoint, DSCP)。  
DSCP 的存在与否指示发送者是否使用转发行为标记了传入通信。
4. 确定 IPQoS 配置文件中针对特定类的包指定的进一步操作
5. 将包传送到 IPQoS 配置文件指定的下一个 IPQoS 模块中，或者将包返回到网络流中

有关分类器的概述，请参阅第 14 页中的“**分类器 (ipgpc) 概述**”。有关在 IPQoS 配置文件中调用分类器的信息，请参阅第 88 页中的“**IPQoS 配置文件**”。

### IPQoS 选定器

`ipgpc` 分类器支持多种可在 IPQoS 配置文件的 `filter` 子句中使用的选定器。定义过滤器时，应始终使用成功检索特定类的通信所需的最少选定器数目。您所定义的过滤器的数目会影响 IPQoS 性能。

下表列出了可用于 `ipgpc` 的选定器。

表 6-1 IPQoS 分类器的过滤选定器

选定器	参数	选定的信息
<code>saddr</code>	IP 地址号。	源地址。
<code>daddr</code>	IP 地址号。	目标地址。
<code>sport</code>	端口号或服务名，如 <code>/etc/services</code> 中所定义。	传出通信类的源端口。
<code>dport</code>	端口号或服务名，如 <code>/etc/services</code> 中所定义。	要将通信类送达的目标端口。

表 6-1 IPQoS 分类器的过滤选定器 (续)

选定器	参数	选定的信息
protocol	协议编号或协议名称，如 /etc/protocols 中所定义。	此通信类要使用的协议。
dsfield	值为 0 至 63 的 DS 代码点 (DS codepoint, DSCP)。	DSCP 用于定义要对包应用的任何转发行为。如果指定了此参数，则还必须指定 dsfield_mask 参数。
dsfield_mask	值为 0 至 255 的位掩码。	与 dsfield 选定器一起使用。dsfield_mask 将应用于 dsfield 选定器以确定要匹配的位。
if_name	接口名称。	用于特定类的传入或传出通信的接口。
user	要选择的 UNIX 用户 ID 号或用户名。如果包中没有用户 ID 或用户名，则使用缺省值 1。	提供给应用程序的用户 ID。
projid	要选择的项目 ID 号。	提供给应用程序的项目 ID。
priority	优先级编号。最低优先级为 0。	提供给此类包的优先级。优先级用于对同类过滤器的重要性进行排序。
direction	该参数可以为下列值之一： LOCAL_IN LOCAL_OUT FWD_IN FWD_OUT	IPQoS 计算机上包流的传输方向。 将本地通信输入到 IPQoS 系统。 将本地通信输出到 IPQoS 系统。 输入要转发的通信。 输出要转发的通信。
precedence	优先级值。最高优先级为 0。	优先级用于对具有相同优先级的过滤器进行排序。
ip_version	V4 或 V6	包使用的寻址方案 (IPv4 或 IPv6)。

## 计量器模块

计量器按包跟踪流的传输速率。然后，计量器确定包是否符合已配置的参数。计量器模块从一组操作中确定要对包执行的下一个操作，具体取决于包的大小、已配置的参数和流速率。

计量器由两个计量模块 tokenmt 和 tswtclmt 组成，您可以在 IPQoS 配置文件中配置它们。您可以为类配置一个或两个模块。

配置计量模块时，您可以定义两个速率参数：

- committed-rate 一针对特定类的包定义可接受的传输速率（以位/秒为单位）
- peak-rate 一针对特定类的包定义所允许的最大传输速率（以位/秒为单位）

对包执行计量操作将生成以下三种结果之一：

- green—包导致流保持在其承诺速率以内。
- yellow—包导致流超过其承诺速率但是没有超过其峰值速率。
- red—包导致流超过其峰值速率。

您可以在 IPQoS 配置文件中使用不同的操作来配置每种结果。承诺速率和峰值速率将在下一节中介绍。

## tokenmt 计量模块

tokenmt 模块使用令牌桶来度量流的传输速率。您可以将 tokenmt 配置为单速率或双速率计量器。tokenmt 操作实例维护两个令牌桶来确定通信流是否符合配置的参数。

tokenmt(7ipp) 手册页介绍了 IPQoS 如何实现令牌计量器模型。您可以在 Kalevi Kilkki 所著的《Differentiated Services for the Internet》和许多 Web 站点上找到有关令牌桶的更多常规信息。

tokenmt 的配置参数如下所示：

- committed\_rate—指定流的承诺速率（以位/秒为单位）。
- committed\_burst—指定承诺突发大小（以位为单位）。committed\_burst 参数定义可以以承诺速率向网络传送的特定类的传出包数目。
- peak\_rate—指定峰值速率（以位/秒为单位）。
- peak\_burst—指定峰值或超额突发大小（以位为单位）。peak\_burst 参数准许通信类具有超过承诺速率的峰值突发大小。
- color\_aware—打开 tokenmt 的识别模式。
- color\_map—定义一个将 DSCP 值映射到绿色、黄色或红色的整数数组。

## 将 tokenmt 配置为单速率计量器

要将 tokenmt 配置为单速率计量器，请不要在 IPQoS 配置文件中为 tokenmt 指定 peak\_rate 参数。要将单速率 tokenmt 实例配置为具有红色、绿色或黄色的结果，必须指定 peak\_burst 参数。如果不使用 peak\_burst 参数，可以将 tokenmt 配置为只有红色或绿色的结果。有关具有两种结果的单速率 tokenmt 的示例，请参见示例 3-3。

tokenmt 作为单速率计量器运行时，peak\_burst 参数实际为超额突发大小。committed\_rate 以及 committed\_burst 或 peak\_burst 必须为非零正整数。

## 将 tokenmt 配置为双速率计量器

要将 tokenmt 配置为双速率计量器，请在 IPQoS 配置文件中为 tokenmt 操作指定 peak\_rate 参数。双速率 tokenmt 始终具有三种结果，即红色、黄色和绿色结果。committed\_rate、committed\_burst 和 peak\_burst 参数必须为非零正整数。



## 将 tokenmt 配置为可识别颜色

要将双速率 tokenmt 配置为可识别颜色，必须添加参数以专门添加“颜色识别”功能。以下是将 tokenmt 配置为可识别颜色的操作语句示例。

示例 6-1 针对 IPQoS 配置文件的可识别颜色 tokenmt 操作

```
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}
  }
}
```

您可以通过将 `color_aware` 参数设置为 `true` 来打开颜色识别。作为可识别颜色的计量器，`tokenmt` 假设先前的某个 `tokenmt` 操作已将包标记为红色、黄色或绿色。除使用双速率计量器的参数外，可识别颜色的 `tokenmt` 还使用包头中的 DSCP 来评估包。

`color_map` 参数包含包头中的 DSCP 要映射到的数组。请看以下 `color_map` 数组：

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

DSCP 为 0 至 20 和 22 的包映射到绿色。DSCP 为 21 和 23 至 42 的包映射到红色。DSCP 为 43 至 63 的包映射到黄色。`tokenmt` 保持缺省颜色映射。但是，您可以根据需要使用 `color_map` 参数来更改缺省设置。

在 `color_action_name` 参数中，可以指定 `continue` 以完成对包的处理。或者，可以添加一个参数以将包发送到标记器操作，例如 `yellow_action_name mark22`。

## tswtclmt 计量模块

`tswtclmt` 计量模块使用基于时间的**速率估计器**来估测通信类的平均带宽。`tswtclmt` 始终作为三重结果计量器运行。速率估计器可估测流的到达速率。此速率应接近通信流在特定时间段（即其**时间窗口**）内的平均传送带宽。速率估计算法可从 RFC 2859 "A Time Sliding Window Three Colour Marker" 中获取。

可以使用以下参数配置 `tswtclmt`：

- `committed_rate`—指定承诺速率（以位/秒为单位）
- `peak_rate`—指定峰值速率（以位/秒为单位）
- `window`—定义保持平均带宽历史记录的时间窗口（以毫秒为单位）

有关 `tswtclmt` 的详细技术信息，请参阅 [tswtclmt\(7ipp\)](#) 手册页。有关类似 `tswtclmt` 的码率整形器的一般信息，请参见 [RFC 2963, A Rate Adaptive Shaper for Differentiated Services](http://www.ietf.org/rfc/rfc2963.txt?number=2963) (<http://www.ietf.org/rfc/rfc2963.txt?number=2963>) (RFC 2963, 区分服务的码率整形器)。

## 标记器模块

IPQoS 包含两个标记器模块，`dscpmk` 和 `dlcosmk`。本节包含有关使用这两个标记器的信息。通常情况下，应使用 `dscpmk`，因为 `dlcosmk` 仅可用于具有 VLAN 设备的 IPQoS 系统。

有关 `dscpmk` 的技术信息，请参阅 [dscpmk\(7ipp\)](#) 手册页。有关 `dlcosmk` 的技术信息，请参阅 [dlcosmk\(7ipp\)](#) 手册页。

### 使用 `dscpmk` 标记器转发包

标记器将在分类器或计量模块处理了通信流之后接收这些流。标记器将使用转发行为为标记通信。此转发行为是在流离开 IPQoS 系统之后要对流执行的操作。要对通信类执行的转发行为在**单跳行为** (per-hop behavior, PHB) 中定义。PHB 为通信类指定优先级，指明该类流相对于其他类流的优先级。PHB 仅管理 IPQoS 系统的连续网络中的转发行为。有关 PHB 的更多信息，请参阅第 18 页中的“单跳行为”。

**包转发**是指在网络中将特定类的通信发送到下一个目的地的过程。对于诸如 IPQoS 系统的主机，会将包从主机转发到本地网络流。对于 Diffserv 路由器，会将包从本地网络转发到路由器的下一个跃点。

标记器使用 IPQoS 配置文件中定义的已知转发行为来标记包头中的 DS 字段。此后，IPQoS 系统和后续可识别 Diffserv 的系统便会按照 DS 字段中的指示转发通信，直到更改标记。要指定 PHB，IPQoS 系统应在包头的 DS 字段中标记值。此值称为区分服务代码点 (differentiated service codepoint, DSCP)。Diffserv 体系结构定义两种转发行为是 EF 和 AF，它们使用不同的 DSCP。有关 DSCP 的概述信息，请参阅第 18 页中的“DS 代码点”。

IPQoS 系统读取通信流的 DSCP，并评估此流相对于其他传出通信流的优先级。然后，IPQoS 系统设置所有并发通信流的优先级，并根据优先级将每个流释放到网络中。

Diffserv 路由器接收传出通信流，并读取包头中的 DS 字段。使用 DSCP，路由器可以对并发通信流设置优先级并进行调度。路由器将按照 PHB 指示的优先级转发每个流。请注意，PHB 不能在网络的边界路由器之外应用，除非后续跃点上的可识别 Diffserv 的系统也可识别同一 PHB。

## 加速转发 (Expedited Forwarding, EF) PHB

**加速转发** (Expedited forwarding, EF) 保证具有推荐 EF 代码点 46 (101110) 的包在释放到网络中时得到最佳处理。通常将加速转发比喻为租用线路。确保所有 Diffserv 路由器保证会优先处理具有 46 (101110) 代码点的包，将其路由到其目的地。有关 EF 的技术信息，请参阅 "RFC 2598 *An Expedited Forwarding PHB*" (RFC 2598, 加速转发 PHB)。

## 保证转发 (Assured Forwarding, AF) PHB

**保证转发** (Assured forwarding, AF) 提供四种可以指定给标记器的不同转发行为类。下表显示了这些类、每个类的三个丢弃优先级以及与每个优先级关联的推荐 DSCP。每个 DSCP 都分别用其 AF 值、十进制值和二进制值表示。

表 6-2 保证转发代码点

	类 1	类 2	类 3	类 4
低丢弃优先级	AF11 =	AF21 =	AF31 =	AF41 =
	10 (001010)	18 (010010)	26 (011010)	34 (100010)
中丢弃优先级	AF12 =	AF22 =	AF32 =	AF42 =
	12 (001100)	20 (010100)	28 (011100)	36 (100100)
高丢弃优先级	AF13 =	AF23 =	AF33 =	AF43 =
	14 (001110)	22 (010110)	30 (011110)	38 (100110)

任何可识别 Diffserv 的系统均可将 AF 代码点用作向不同通信类提供区分转发行为的指南。

当这些包到达 Diffserv 路由器时，路由器便会评估包的代码点以及队列中其他通信的 DSCP。然后，路由器会转发或丢弃包，具体取决于可用带宽以及包的 DSCP 所指定的优先级。请注意，以 EF PHB 为标记的包保证比以各种 AF PHB 为标记的包优先使用带宽。

协调网络中所有 IPQoS 系统和 Diffserv 路由器之间的包标记，确保包按预期方式转发。例如，假定网络中的 IPQoS 系统使用 AF21 (010010)、AF13 (001110)、AF43 (100110) 和 EF (101110) 代码点标记包。这样，便需要将 AF21、AF13、AF43 和 EF DSCP 添加到 Diffserv 路由器上的相应文件中。

有关 AF 代码点表的技术说明，请参阅 RFC 2597。路由器制造商 Cisco Systems 和 Juniper Networks 在其 Web 站点上提供了有关设置 AF PHB 的详细信息。您可以使用此信息来为 IPQoS 系统和路由器定义 AF PHB。此外，路由器制造商的文档包含在其设备上设置 DS 代码点的说明。

## 为标记器提供 DSCP

DSCP 的长度为 6 位。DS 字段的长度为 1 字节。定义 DSCP 时，标记器将使用 DS 代码点来标记包头的前 6 个重要的位。其余的 2 个次要的位将不会使用。

要定义 DSCP，您可以在标记器操作语句中使用以下参数：

```
dscp_map{0-63:DS_codepoint}
```

`dscp_map` 参数是包含 64 个元素的数组，此数组将使用 (DSCP) 值进行填充。`dscp_map` 用于将传入 DSCP 映射到 `dscpmk` 标记器所要应用的传出 DSCP。

必须采用十进制表示法为 `dscp_map` 指定 DSCP 值。例如，必须将 EF 代码点 101110 转换为十进制值 46，结果为 `dscp_map{0-63:46}`。对于 AF 代码点，必须将表 6-2 中所示的各种代码点转换为十进制表示法，才能在 `dscp_map` 中使用。

## 将 dlcosmk 标记器用于 VLAN 设备

`dlcosmk` 标记器模块在数据报的 MAC 头中标记转发行为。只能在具有 VLAN 接口的 IPQoS 系统上使用 `dlcosmk`。

`dlcosmk` 会向 MAC 头中添加称为 **VLAN 标记** 的四个字节。VLAN 标记包括由 IEEE 801.D 标准定义的 3 位用户优先级值。了解 VLAN 的可识别 Diffserv 的交换机可以读取数据报中的用户优先级字段。801.D 用户优先级值实现服务类 (class-of-service, CoS) 标记，这些标记为商业交换机所熟知和了解。

您可以通过定义下表中列出的服务类标记，在 `dlcosmk` 标记器操作中使用用户优先级值。

表 6-3 801.D 用户优先级值

服务类	定义
0	尽力服务
1	后台
2	备用
3	出色服务
4	受控负载
5	少于 100 ms 延迟的视频
6	少于 10 ms 延迟的视频
7	网络控制

有关 `dlcosmk` 的更多信息，请参阅 `dlcosmk(7ipp)` 手册页。

## 具有 VLAN 设备的系统的 IPQoS 配置

本节介绍一个简单网络方案，说明如何在具有 VLAN 设备的系统上实现 IPQoS。此方案包括两个由交换机连接的 IPQoS 系统：machine1 和 machine2。machine1 上的 VLAN 设备的 IP 地址为 10.10.8.1。machine2 上的 VLAN 设备的 IP 地址为 10.10.8.3。

以下用于 machine1 的 IPQoS 配置文件说明了标记通过交换机转到 machine2 的通信的简单解决方案。

示例 6-2 具有 VLAN 设备的系统的 IPQoS 配置文件

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

    filter {
        name myfilter2
        daddr 10.10.8.3
        class myclass
    }

    class {
        name myclass
        next_action mark4
    }
}

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

在此配置中，所有来自 machine1 并且目标为 machine2 上的 VLAN 设备的通信都将被传送到 dlcosmk 标记器。mark4 标记器操作指示 dlcosmk 向 CoS 为 4 的 myclass 类数据报中添加 VLAN 标记。用户优先级值 4 指示两台计算机之间的交换机应该为来自 machine1 的 myclass 通信流提供受控负载转发。

## flowacct 模块

IPQoS flowacct 模块记录有关通信流的信息，此过程称为**流记帐**。流记帐将生成可用于向用户收费或评估到特定类的通信量的数据。

流记帐为可选过程。flowacct 通常是已计量或已标记的通信流在释放到网络流之前可能遇到的最后一个模块。有关 flowacct 在 Diffserv 模型中的位置的说明，请参见图 1-1。有关 flowacct 的详细技术信息，请参阅 flowacct(7ipp) 手册页。

要启用流记帐，除需要使用 `flowacct` 外，还需要使用 Oracle Solaris `exacct` 记帐功能和 `acctadm` 命令。有关设置流记帐的所有步骤的信息，请参阅第 73 页中的“设置流记帐（任务列表）”。

## flowacct 参数

`flowacct` 模块将有关流的信息收集在流表中，该表由流记录组成。表中的每项都包含一个流记录。无法显示流表。

在 IPQoS 配置文件中，您可以定义以下 `flowacct` 参数以度量流记录并将记录写入流表中：

- `timer`—定义将已超时的流从流表中删除和写入由 `acctadm` 创建的文件中的时间间隔（以毫秒为单位）
- `timeout`—定义一个时间间隔（以毫秒为单位），它指定了在流超时之前包流必须保持不活动状态的时间

---

注—您可以将 `timer` 和 `timeout` 配置为具有不同的值。

---

- `max_limit`—针对可存储在流表中的流记录数设置上限

有关 `flowacct` 参数在 IPQoS 配置文件中如何应用的示例，请参阅第 60 页中的“如何在 IPQoS 配置文件中配置流控制”。

## 流表

`flowacct` 模块维护着一个流表，此表记录了 `flowacct` 实例发现的所有包流。

流由以下参数标识，其中包括 `flowacct` 8 元组：

- 源地址
- 目标地址
- 源端口
- 目标端口
- DSCP
- 用户 ID
- 项目 ID
- 协议编号

如果流的 8 元组的所有参数保持不变，则流表仅包含一个项。`max_limit` 参数确定流表可以包含的项数。

将按照在 IPQoS 配置文件中为 `timer` 参数指定的时间间隔扫描流表。缺省值是 15 秒。如果 IPQoS 系统在 IPQoS 配置文件指定的 `timeout` 时间间隔内未发现流的包，则表示此流已“超时”。缺省超时时间间隔为 60 秒。已超时的项将被写入使用 `acctadm` 命令创建的记帐文件中。

## flowacct 记录

flowacct 记录包含下表中介紹的属性。

表 6-4 flowacct 记录的属性

属性名称	属性内容	类型
<i>src-addr-address-type</i>	始发者的源地址。 <i>address-type</i> 为 v4 (对于 IPv4) 或 v6 (对于 IPv6), 如 IPQoS 配置文件中所指定。	基本
<i>dest-addr-address-type</i>	包的目标地址。 <i>address-type</i> 为 v4 (对于 IPv4) 或 v6 (对于 IPv6), 如 IPQoS 配置文件中所指定。	基本
<i>src-port</i>	传出流的源端口。	基本
<i>dest-port</i>	要将此流送达的目标端口号。	基本
<i>protocol</i>	流的协议编号。	基本
<i>total-packets</i>	流中的包数。	基本
<i>total-bytes</i>	流中的字节数。	基本
<i>action-name</i>	记录此流的 flowacct 操作的名称。	基本
<i>creation-time</i>	flowacct 首次发现该流的包的时间。	仅扩展
<i>last-seen</i>	上次发现该流的包的时间。	仅扩展
<i>diffserv-field</i>	此流的传出包头中的 DSCP。	仅扩展
<i>user</i>	从应用程序中获取的 UNIX 用户 ID 或用户名。	仅扩展
<i>projid</i>	从应用程序中获取的项目 ID。	仅扩展

## 将 acctadm 用于 flowacct 模块

可以使用 `acctadm` 命令创建用于存储 flowacct 生成的各种流记录的文件。`acctadm` 可与扩展记帐功能一起使用。有关 `acctadm` 的技术信息, 请参阅 [acctadm\(1M\)](#) 手册页。

flowacct 模块将监视流, 并使用流记录填充流表。然后, flowacct 将在 `timer` 指定的时间间隔内评估其参数和属性。如果在 `last_seen` 值与 `timeout` 值之和对应的时间内未发现包, 则表示包已超时。所有超时项都将从流表中删除。每经过 `timer` 参数指定的时间间隔, 都会将这些项写入记帐文件中。

要调用 `acctadm` 以用于 flowacct 模块, 请使用以下语法:

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

`acctadm -e` 调用带有 `-e` 选项的 `acctadm`。`-e` 表示后面跟有资源列表。

*file-type* 指定要收集的属性。*file-type* 必须用 `basic` 或 `extended` 替换。有关每种文件类型中的属性列表, 请参阅表 6-4。

- f *file-name*      创建文件 *file-name* 来保存流记录。
- flow                指示 acctadm 与 IPQoS 一起运行。

## IPQoS 配置文件

本节包含有关 IPQoS 配置文件各部分的完整详细信息。IPQoS 引导时激活的策略存储在文件 `/etc/inet/ipqosinit.conf` 中。尽管您可以编辑此文件，但是对于新 IPQoS 系统而言，最佳做法是创建具有不同名称的配置文件。有关应用和调试 IPQoS 配置的任务，请参见第 3 章，[创建 IPQoS 配置文件（任务）](#)。

示例 6-3 中显示了 IPQoS 配置文件的语法。

此示例使用以下约定：

- `computer-style type`—用于介绍配置文件各部分的语法信息。您无法键入任何在计算机样式类型中出现的文本。
- **bold type**—您必须在 IPQoS 配置文件中键入的文字文本。例如，您必须始终以 `fmt_version` 来开始 IPQoS 配置文件。
- *italic type*—您使用有关配置的说明性信息来替换的变量文本。例如，您必须始终用与配置有关的信息来替换 `action-name` 或 `module-name`。

示例 6-3 IPQoS 配置文件的语法

```
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
```



示例 6-3 IPQoS 配置文件的语法 (续)

```

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

```

剩余部分介绍 IPQoS 配置文件的各个主要部分。

## action 语句

您可以使用 action 语句来调用第 77 页中的“IPQoS 体系结构和 Diffserv 模型”中介绍的各种 IPQoS 模块。

当您创建 IPQoS 配置文件时，必须始终以版本号开始。然后，您必须添加以下 action 语句来调用分类器：

```

fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
}

```

分类器 action 语句后接 params 子句或 class 子句。

对于所有其他 action 语句，请使用以下语法：

```

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

```

name *action\_name* 为操作指定名称。

module *module\_name* 标识要调用的 IPQoS 模块，此模块必须为表 6-5 中的模块之一。

<i>params_clause</i>	可以为要处理的分类器参数，例如全局统计信息或者要处理的下一个操作。
<i>cf_clauses</i>	一组零个或多个 <code>class</code> 子句或 <code>filter</code> 子句。

## 模块定义

模块定义指示由哪个模块处理 `action` 语句中的参数。IPQoS 配置文件可以包括以下模块。

表 6-5 IPQoS 模块

模块名称	定义
<code>ipgpc</code>	IP 分类器
<code>dscpmk</code>	用于在 IP 包中创建 DSCP 的标记器
<code>dlcosmk</code>	用于 VLAN 设备的标记器
<code>tokenmt</code>	令牌桶计量器
<code>tswtclmt</code>	时间滑动窗口计量器
<code>flowacct</code>	流记帐模块

## class 子句

您可以为每个通信类定义一个 `class` 子句。

可以使用以下语法定义 IPQoS 配置中的其余类：

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

要为特定类启用统计信息收集，必须先在 `ipgpc.classify action` 语句中启用全局统计功能。有关更多信息，请参阅第 89 页中的“[action 语句](#)”。

当需要针对某一类打开统计信息收集时，请使用 `enable_stats TRUE` 语句。如果不需要收集类的统计信息，则可以指定 `enable_stats FALSE`。或者，可以删除 `enable_stats` 语句。

未专门定义的启用了 IPQoS 的网络中的通信将归入缺省类中。

## filter 子句

**过滤器**由多个用于将通信流分类的选定器构成。这些选定器具体定义了属于类子句中创建的类的通信所要应用的条件。如果包与最高优先级的过滤器的所有选定器相匹配，则此包被视为此过滤器类的成员。有关可以与 `ipgpc` 分类器一起使用的选定器的完整列表，请参阅表 6-1。

可以使用 `filter` 子句在 IPQoS 配置文件中定义过滤器，此子句的语法如下：

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

## params 子句

`params` 子句包含操作语句中定义的模块的处理指令。可以针对 `params` 子句使用以下语法：

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

在 `params` 子句中，可以使用适用于模块的参数。

`params` 子句中的 `params-stats` 值可以是 `global_stats TRUE` 或 `global_stats FALSE`。`global_stats TRUE` 指令将针对调用全局统计信息的 `action` 语句启用 UNIX 样式统计信息。可以使用 `kstat` 命令来查看该统计信息。按类启用统计功能之前，必须先为 `action` 语句启用统计功能。

## ipqosconf 配置实用程序

您可以使用 `ipqosconf` 实用程序来读取 IPQoS 配置文件，并在 UNIX 内核中配置 IPQoS 模块。`ipqosconf` 可执行以下操作：

- 将配置文件应用于 IPQoS 内核模块 (`ipqosconf -a filename`)
- 列出当前驻留在内核中的 IPQoS 配置文件 (`ipqosconf -l`)
- 确保每次重新引导计算机时会读取和应用当前的 IPQoS 配置 (`ipqosconf -c`)
- 刷新当前 IPQoS 内核模块 (`ipqosconf -f`)

有关技术信息，请参阅 `ipqosconf(1M)` 手册页。



# 索引

---

## A

acctadm 命令,用于流记帐, 16, 75, 87  
action 语句, 89

## C

class 子句,在 IPQoS 配置文件中, 46  
class 子句,在 IPQoS 配置文件中, 90

## D

Diffserv 模型  
    IPQoS 实现, 14-18, 16-17, 17  
    标记器模块, 16  
    分类器模块, 14-15  
    计量器模块, 15-16  
    流示例, 17  
dlcosmk 标记器, 16  
    VLAN 标记, 84  
    规划数据报转发, 35  
    用户优先级值, 表, 84  
DS 代码点 (DS codepoint, DSCP), 16, 18  
    AF 转发代码点, 19, 83  
    dscp\_map 参数, 84  
    EF 转发代码点, 19, 83  
    PHB 和 DSCP, 18  
    定义,在 IPQoS 配置文件中, 49  
    规划,在 QoS 策略中, 35  
    配置,在 diffserv 路由器上, 63, 82  
    颜色识别配置, 81

dscpmk 标记器, 16  
    调用,在标记器 action 语句中, 49, 53, 59, 61  
    规划包转发, 34  
    用于包转发的 PHB, 82-84

## F

filter 子句,在 IPQoS 配置文件中, 47, 91  
flowacct 模块, 16-17, 85  
    acctadm 命令,用于创建流记帐文件, 87  
    flowacct 的 action 语句, 51  
    参数, 86  
    流记录, 74  
    流记录表, 86  
    流记录的属性, 87

## I

ipgpc 分类器,请参见分类器模块  
IPQoS, 9  
    Diffserv 模型实现, 14-18  
    IPQoS 网络上的路由器, 63  
    QoS 策略规划, 27  
    VLAN 设备支持, 84-85  
    错误消息, 68  
    功能, 10  
    配置规划, 23  
    配置示例, 37-39  
    配置文件, 42, 88  
        action 语句语法, 89  
        class 子句, 46

## IPQoS, 配置文件 (续)

- filter 子句, 47
  - IPQoS 模块列表, 90
  - 标记器 action 语句, 49
  - 初始 action 语句, 89
  - 初始 action 语句, 45
  - 语法, 88
  - 生成统计信息, 76
  - 手册页, 11
  - 通信管理功能, 13, 14
  - 网络示例, 42
  - 相关 RFC, 11
  - 消息日志, 67
  - 支持的网络拓扑, 24, 25, 26
- IPQoS 错误消息, 68
- IPQoS 的 syslog.conf 文件日志, 67
- IPQoS 的网络示例, 42
- IPQoS 的网络拓扑, 24
- 包含启用了 IPQoS 的防火墙的 LAN, 26
  - 包含启用了 IPQoS 的服务器场的 LAN, 24
  - 包含启用了 IPQoS 的主机的 LAN, 24
- 配置示例, 37
- IPQoS 配置文件示例
- VLAN 设备配置, 85
  - 高级 Web 服务器, 43
  - 尽力服务 Web 服务器, 44
  - 颜色识别段, 81
  - 应用服务器, 54
- IPQoS 统计信息
- 启用基于类的统计信息, 90
  - 启用全局统计功能, 46, 90
  - 生成, 通过 kstat 命令, 76
- IPQoS 网络中的虚拟 LAN (virtual LAN, VLAN) 设备, 84-85
- ipqosconf, 42
- ipqosconf 命令
- 列出当前配置, 67
  - 命令选项, 91
  - 应用配置, 66, 67

## K

kstat 命令, 用于 IPQoS, 76

## P

- params 子句
- 定义全局统计功能, 46, 91
  - 用于 flowacct action, 51
  - 用于标记器 action, 49
  - 用于计量 action, 61
  - 语法, 91

## Q

- QoS 策略, 12
- 策略组织模板, 27
  - 创建过滤器, 30
  - 规划任务列表, 27
  - 实现, 在 IPQoS 配置文件中, 41

## T

- tokenmt 计量器, 15
- 单速率计量器, 80
  - 计量速率, 80-81
  - 双速率计量器, 80
  - 速率参数, 80
  - 颜色识别配置, 16, 81
- tswtclmt 计量器, 15, 81-82
- 计量速率, 81

## W

- Web 服务器
- 为 IPQoS 配置, 44, 52, 53
- web 服务器, 为 IPQoS 配置, 43

## 保

- 保证转发 (assured forwarding, AF), 19, 83
- AF 代码点表, 83
  - 用于标记器 action 语句, 49

## 标

- 标记器模块, 16
  - 另请参见 dlcsmk 标记器
  - 另请参见 dscpmk 标记器
- PHB, 用于 IP 包转发, 18
- 支持 VLAN 设备, 84–85
- 指定 DS 代码点, 84

## 带

- 带宽控制, 13
  - 规划, 在 QoS 策略中, 30

## 单

- 单跳行为 (per-hop behavior, PHB), 18
  - AF 转发, 19
  - EF 转发, 19
  - 定义, 在 IPQoS 配置文件中, 62
  - 使用, 通过 dscpmk 标记器, 82–84

## 分

- 分类器模块, 14–15
  - action 语句, 45
  - 分类器的功能, 78

## 服

- 服务级别协议 (service-level agreement, SLA), 12
  - 服务类, 15
  - 客户机记帐, 基于流记帐, 74
  - 提供不同的服务类, 13
- 服务类, 请参见类
- 服务类 (class of service, CoS) 标记, 16
- 服务质量 (quality of service, QoS)
  - QoS 策略, 12
  - 任务, 10

## 负

- 负载均衡, 在启用了 IPQoS 的网络中, 25

## 过

- 过滤器, 15
  - filter 子句语法, 91
  - 创建, 在 IPQoS 配置文件中, 53, 57
  - 规划, 在 QoS 策略中, 30
  - 选定器, 列表, 78

## 计

- 计量模块
  - 另请参见 tokenmt 计量器
  - 另请参见 tswtclmt 计量器
  - 调用, 在 IPQoS 配置文件中, 60
  - 计量结果, 16, 80
  - 介绍, 15–16

## 加

- 加速转发 (expedited forwarding, EF), 19, 83
  - 定义, 在 IPQoS 配置文件中, 50

## 可

- 可识别 Diffserv 的路由器
  - 规划, 28
  - 评估 DS 代码点, 83

## 类

- 类, 15
  - class 子句的语法, 90
  - 定义, 在 IPQoS 配置文件中, 52, 56
  - 选定器, 列表, 78

## 流

- 流记帐, 74-75, 85
  - 流记录表, 86
- 流控制, 通过计量模块, 15

## 启

- 启用了 IPQoS 的网络的硬件, 24

## 请

- 请求注解文档 (Requests for Comments, RFC), IPQoS, 11

## 区

- 区分服务, 9-10
  - 区分服务模型, 14-18
  - 提供不同的服务类, 13
  - 网络拓扑, 24

## 任

- 任务列表
  - IPQoS
    - QoS 策略规划, 27
    - 流记帐设置, 73
    - 配置规划, 23
    - 配置文件创建, 41

## 通

- 通信管理
  - 规划网络拓扑, 24
  - 控制带宽, 13
  - 控制流, 15-16
  - 设置通信流的优先级, 13-14
  - 转发通信, 18, 19, 20
- 通信一致性
  - 定义, 60

## 通信一致性 (续)

- 规划
  - QoS 策略中的结果, 33
  - QoS 策略中的速率, 33
- 结果, 16, 80
- 速率参数, 79, 80

## 选

- 选定器, 15
  - IPQoS 5 元组, 14
  - 规划, 在 QoS 策略中, 30
  - 选定器, 列表, 78

## 颜

- 颜色识别, 16, 81

## 应

- 应用服务器, 为 IPQoS 配置, 54

## 用

- 用户优先级值, 16

## 转

- 转发通信
  - IP 包转发, 使用 DSCP, 18
  - PHB 对包转发的影响, 82-84
  - 规划, 在 QoS 策略中, 29
  - 数据报转发, 84-85
  - 通过 Diffserv 网络的通信流, 19