Oracle® Solaris 管理：IP 服务

文件号码 E38856–03
2013 年 3 月

ORACLE

130701@25097

目录

前言

欢迎阅读《Oracle Solaris 管理：IP 服务》。本书是一套多卷丛书（全十四册）中的一册，该从书涵盖了 Oracle Solaris 系统管理的主要内容。本书假定您已经安装 Oracle Solaris OS。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。

在本文档中，这些与 x86 相关的术语表示以下含义：

- x86 泛指 64 位和 32 位的 x86 兼容产品系列。
- x64 特指 64 位的 x86 兼容 CPU。
- “32 位 x86”指出了有关基于 x86 的系统的特定 32 位信息。

有关支持的系统，请参见《Oracle Solaris OS: Hardware Compatibility Lists》（《Oracle Solaris OS：硬件兼容性列表》）。

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。

系统管理指南系列书籍的结构

下表列出了系统管理指南系列中各本书包含的主题。

书名	主题
《Oracle Solaris 管理：基本管理》	使用 Oracle Solaris 命令，引导和关闭系统，服务器和客户机支持，管理用户帐户和组，管理服务、系统信息、系统资源和系统性能，管理软件、控制台和终端，以及排除系统和软件问题
《系统管理指南：高级管理》	终端和调制解调器、系统资源（磁盘配额、记帐和 crontab）、系统进程以及 Oracle Solaris 软件问题故障排除
《System Administration Guide: Devices and File Systems》	可移除介质、磁盘和设备、文件系统以及备份和还原数据
《Oracle Solaris 管理：IP 服务》	TCP/IP 网络管理、IPv4 和 IPv6 地址管理、DHCP、IPsec、IKE、IP 过滤器、IP 网络多路径 (IP Network Multipathing, IPMP) 和 IPQoS
《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》	DNS、NIS 和 LDAP 命名和目录服务，包括从 NIS 转换到 LDAP 以及从 NIS+ 转换到 LDAP
《System Administration Guide: Naming and Directory Services (NIS+)》	NIS+ 命名和目录服务
《系统管理指南：网络服务》	Web 高速缓存服务器、与时间相关的服务、网络文件系统（NFS 和 Autofs）、邮件、SLP 和 PPP
《系统管理指南：Oracle Solaris Containers－资源管理和 Oracle Solaris Zones》	资源管理主题项目和任务、扩展记帐、资源控制、公平份额调度器 (fair share scheduler, FSS)、使用资源上限设置守护进程 (rcapd) 的物理内存控制，以及资源池；使用 Solaris Zones 软件分区技术和 `lx` 标记区域的虚拟功能
《系统管理指南：打印》	打印主题和任务，使用服务、工具、协议和技术来设置及管理打印服务和打印机
《System Administration Guide: Security Services》	审计、设备管理、文件安全、BART（基本审计和报告工具）、Kerberos 服务、PAM（可插拔验证模块）、加密框架、密钥管理、特权、RBAC（基于角色的访问控制）、SASL（简单身份认证和安全层）和安全 Shell
《Oracle Solaris ZFS 管理指南》	ZFS（Zettabyte 文件系统）存储池以及文件系统的创建和管理、快照、克隆、备份、使用访问控制列表 (Access Control List, ACL) 保护 ZFS 文件、在安装区域的 Solaris 系统中使用 ZFS、仿真卷以及故障排除和数据恢复
《Trusted Extensions 管理员规程》	特定于 Trusted Extensions 的系统管理
《Oracle Solaris Trusted Extensions 配置指南》	从 Solaris 10 5/08 发行版开始，介绍如何规划、启用及初始配置 Trusted Extensions

相关书籍

本书参考了以下普通书籍。

- 由 Stevens, W.Richard 编著的《TCP/IP Illustrated, Volume 1, The Protocols》。Addison Wesley 出版，1994。
- 由 Hunt Craig 编著的《TCP/IP Network Administration, 3rd Edition》。O'Reilly 出版社，2002。
- 由 Perkins, Charles E 编著的《Mobile IP Design Principles and Practices》。Massachusetts，Addison-Wesley Publishing Company 出版，1998。
- 由 Solomon, James D 编著的《Mobile IP: The Internet Unplugged》。New Jersey，Prentice-Hall, Inc. 出版，1998。
- 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。

相关的第三方 Web 站点引用

本文档引用了第三方 URL，这些 URL 提供了额外的相关信息。

Oracle Solaris 的 IP 过滤器功能源自开源 IP 过滤器软件。要查看 IP 过滤器的许可证条款、所有权和版权声明，缺省路径为 `/usr/lib/ipf/IPFILTER.LICENCE`。如果已经将 Oracle Solaris 操作系统 安装在非缺省的其他任何位置，请修改指定的路径，以便访问安装位置中的该文件。

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 `ls -a` 列出所有文件。 `machine_name% you have mail.`
`AaBbCc123`	用户键入的内容，与计算机屏幕输出的显示不同	`machine_name%` **`su`** `Password:`
aabbcc123	要使用实名或值替换的命令行占位符	删除文件的命令为 `rm` *filename*。
AaBbCc123	保留未译的新词或术语以及要强调的词	这些称为 *Class* 选项。 **注意**：有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	**高速缓存**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#`

第 1 部 分

系统管理介绍：IP 服务

本部分介绍 TCP/IP 协议套件及其如何在 Oracle Solaris 中实施。

◆ ◆ ◆ 第 1 章

Oracle Solaris TCP/IP 协议套件（概述）

本章介绍 Oracle Solaris 如何实施 TCP/IP 网络协议套件。所提供的信息适用于对 TCP/IP 基本概念不熟悉的系统管理员和网络管理员。本书的其余部分假定您熟悉这些概念。

本章包含以下信息：

本发行版新增功能

从 Solaris 10 5/08 开始，不再提供移动 IP 功能。Solaris 10 OS 8/07 发行版和早期发行版中提供了移动 IP 功能。

TCP/IP 协议套件介绍

本节详细介绍了 TCP/IP 中包括的协议。虽然是一些概念性的信息，但是您应该了解这些协议的名称，还应该了解每种协议的作用。

"TCP/IP" 是一组网络协议常用的首字母缩略词，这些协议组成了 *Internet* **协议套件**。许多文章都使用术语 "Internet" 来描述协议套件和全局广域网。在本书中，"TCP/IP" 专指 Internet 协议套件。"Internet" 是指广域网以及管理 Internet 的机构。

要将您的 TCP/IP 网络与其他网络互连，必须为您的网络获取一个唯一的 IP 地址。编写此内容时，您可以从 Internet 服务提供商 (Internet Service Provider, ISP) 获取该 IP 地址。

如果要将网络中的主机加入 Internet 域名系统 (Domain Name System, DNS)，则必须获取并注册唯一的域名。InterNIC 通过一系列全球注册机构来协调域名注册。有关 DNS 的更多信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

协议层和开放系统互连模型

大多数网络协议套件的结构都由一系列层组成，有时统称为**协议栈**。每一层都针对特定用途而设计，并且同时存在于发送系统和接收系统上。一个系统上的某个特定层发送或接收的对象与另一个系统上的**对等进程**发送或接收的对象完全相同。这些活动的发生与所考虑的层的上下层中的活动无关。实际上，系统上每一层的活动都独立于同一系统上的其他层，并且可与其他系统上的同一层并行执行操作。

OSI 参考模型

大多数网络协议套件的结构都由多个层组成。国际标准化组织 (International Organization for Standardization, ISO) 设计了使用结构化层的开放系统互连 (Open Systems Interconnection, OSI) 参考模型。OSI 模型介绍了网络活动的七层结构。每个层都与一个或多个协议关联。对于协作网络间所有类型的数据传送，这些层执行的数据传送操作都是相同的。

OSI 模型列出了从顶层（第 7 层）到底层（第 1 层）的协议层。下表显示了此模型。

表 1–1 开放系统互连参考模型

层编号	层名称	说明
7	应用	由所有人均可使用的标准通信服务和应用程序组成。
6	表示	确保将信息以系统可识别的形式传送到接收系统。
5	会话	管理协作系统之间的连接和终止。
4	传输	管理数据传输，同时还负责确保收到的数据与传送的数据完全相同。
3	网络	管理网络间的数据寻址和传送。
2	数据链路	处理网络介质间的数据传送。
1	物理	定义网络硬件的特征。

OSI 模型定义了并不特定于任何网络协议套件的概念性操作。例如，OSI 网络协议套件可实现 OSI 模型的所有七个层。TCP/IP 使用 OSI 模型的一些层，并且还会组合其他层。其他网络协议（如 SNA）会添加第八层。

TCP/IP 协议体系结构模型

OSI 模型通过一系列协议描述了理想的网络通信。TCP/IP 并不直接对应于此模型。TCP/IP 或者将几个 OSI 层组合为一个层，或者根本不使用某些层。下表显示了 Oracle Solaris 实现的 TCP/IP 层。该表列出了从最顶层（应用层）到最底层（物理网络层）的各层。

表 1–2 TCP/IP 协议栈

OSI 参考层编号	等效的 OSI 层	TCP/IP 层	TCP/IP 协议示例
5、6、7	应用层、会话层、表示层	应用	NFS、NIS、DNS、LDAP、`telnet`、`ftp`、`rlogin`、`rsh`、`rcp`、RIP、RDISC、SNMP 等
4	传输层	传输	TCP、UDP、SCTP
3	网络	Internet	IPv4、IPv6、ARP、ICMP
2	数据链路层	数据链路	PPP、IEEE 802.2
1	物理层	物理网络	以太网 (IEEE 802.3)、令牌环、RS-232、FDDI 等等

该表显示了 TCP/IP 协议层和 OSI 模型中的等效层。另外，还显示了可用于 TCP/IP 协议栈各级别的协议的示例。通信事务中涉及的每个系统都运行协议栈的唯一实施。

物理网络层

物理网络层指定要用于网络的硬件的特征。例如，物理网络层可指定通信介质的物理特征。TCP/IP 的物理层介绍了一些硬件标准，如作为以太网网络介质规范的 IEEE 802.3 以及作为标准管脚连接器规范的 RS-232。

数据链路层

数据链路层标识包的网络协议类型，在此实例中为 TCP/IP。数据链路层还提供了错误控制和“成帧”。数据链路层协议的示例包括以太网 IEEE 802.2 成帧和点对点协议 (Point-to-Point Protocol, PPP) 成帧。

Internet 层

Internet 层也称为**网络层**或 *IP* **层**，可为网络接受和传送包。该层包括功能强大的 Internet 协议 (Internet Protocol, IP)、地址解析协议 (Address Resolution Protocol, ARP) 和 Internet 控制消息协议 (Internet Control Message Protocol, ICMP)。

IP 协议

IP 协议及其关联的路由协议可能是整个 TCP/IP 套件中最重要的部分。IP 负责以下操作：

- **IP 寻址**－IP 寻址约定是 IP 协议的一部分。第 51 页中的 “设计 IPv4 寻址方案”介绍了 IPv4 寻址，第 67 页中的 “IPv6 寻址概述”介绍了 IPv6 寻址。
- **主机到主机通信**－根据接收系统的 IP 地址，IP 确定包必须采用的路径。
- **包格式设置**－IP 将包组装到称为**数据报**的单元中。第 41 页中的 “Internet 层：准备传送包的位置”中全面介绍了数据报。
- **分段**－如果包太大而无法通过网络介质进行传输，则发送系统上的 IP 会将包分为较小的段。然后，接收系统上的 IP 会将这些段重构为原始包。

Oracle Solaris 同时支持 IPv4 和 IPv6 寻址格式，这两种格式都在本书中进行了介绍。为避免对 Internet 协议进行寻址时出现混淆，请使用以下约定之一：

- 如果说明中使用了术语 "IP"，则此说明既适用于 IPv4 又适用于 IPv6。
- 如果说明中使用了术语 "IPv4"，则此说明仅适用于 IPv4。
- 如果说明中使用了术语 "IPv6"，则此说明仅适用于 IPv6。

ARP 协议

从概念上讲，地址解析协议 (Address Resolution Protocol, ARP) 位于数据链路层和 Internet 层之间。ARP 通过将以太网地址（长度为 48 位）映射到已知的 IP 地址（长度为 32 位），协助 IP 将数据报定向到相应的接收系统。

ICMP 协议

Internet 控制消息协议 (Internet Control Message Protocol, ICMP) 可检测并报告网络错误情况。ICMP 将报告以下情况：

- **丢弃的包**－到达速度太快而无法处理的包
- **连接故障**－无法连接到目标系统
- **重定向**－重定向发送系统，以便使用其他路由器

第 8 章，管理 TCP/IP 网络（任务）介绍了有关将 ICMP 用于错误检测的 Oracle Solaris 命令的更多信息。

传输层

TCP/IP **传输层**通过交换数据接收的确认信息并重新传送丢失的包，可确保包按顺序到达且不会出现错误。这种通信类型称为**端对端**。此级别的传输层协议栈包括传输控制协议 (Transmission Control Protocol, TCP)、用户数据报协议 (User Datagram Protocol, UDP) 以及流控制传输协议 (Stream Control Transmission Protocol, SCTP)。TCP 和 SCTP 可提供可靠的端对端服务。UDP 则会提供不可靠的数据报服务。

TCP 协议

TCP 使应用程序能够相互通信，就像它们通过物理电路连接一样。TCP 发送数据的形式类似于逐个字符进行传送，而不是以独立的包进行发送。这种传输由以下各项组成：

- 起始点，用于打开连接
- 按字节顺序进行的完整传输
- 结束点，用于关闭连接。

TCP 会向传送的数据中附加一个头。此头包含许多参数，可帮助发送系统上的进程连接到接收系统上的对等进程。

TCP 通过在发送主机和接收主机之间建立端对端连接，确认包是否已到达其目的地。因此，TCP 被视为一种“可靠的、面向连接的”协议。

SCTP 协议

SCTP 是一种可靠的、面向连接的传输层协议，它为应用程序提供的服务与 TCP 提供的服务相同。此外，SCTP 还可以支持在具有多个地址或**多宿主**系统之间建立连接。发送系统和接收系统之间的 SCTP 连接称为**关联**。关联中的数据组织成多个块。由于 SCTP 支持多宿主，因此，某些应用程序（尤其是电信行业使用的应用程序）需要通过 SCTP 而不是 TCP 运行。

UDP 协议

UDP 可提供数据报传送服务，并且不会检验接收主机和发送主机之间的连接。由于 UDP 不需要建立和验证连接，因此，发送少量数据的应用程序可使用 UDP。

应用层

应用层定义了任何用户均可使用的标准 Internet 服务和网络应用程序。这些服务与传输层协同工作以发送和接收数据。存在多种应用层协议。

以下列表显示了应用层协议的示例：

- 标准 TCP/IP 服务，如 `ftp`、`tftp` 和 `telnet` 命令
- UNIX "r" 命令，如 `rlogin` 和 `rsh`
- 名称服务，如 NIS 和域名系统 (Domain Name System, DNS)
- 目录服务 (LDAP)
- 文件服务，如 NFS 服务
- 简单网络管理协议 (Simple Network Management Protocol, SNMP)，用于启用网络管理
- 路由器搜索 (Router Discovery, RDISC) 服务器协议和路由信息协议 (Routing Information Protocol, RIP) 路由协议

标准 TCP/IP 服务

- **FTP 和匿名 FTP** – 文件传输协议 (File Transfer Protocol, FTP) 可以向远程网络以及从远程网络传输文件。此协议包括 `ftp` 命令和 `in.ftpd` 守护进程。使用 FTP，用户可以在本地主机的命令行中指定远程主机名和文件传送命令选项。然后，远程主机上的 `in.ftpd` 守护进程会处理来自本地主机的请求。与 `rcp` 不同，即使远程计算机没有运行基于 UNIX 的操作系统， `ftp` 仍会正常工作。除非远程系统已配置为允许匿名 FTP，否则，用户必须登录到远程系统以建立 `ftp` 连接。

  您可以从连接到 Internet 的**匿名 *FTP* 服务器**获取大量资料。大学和其他机构都设置了这些服务器，以便向公共域提供软件、研究论文和其他信息。登录到此类型的服务器时，您可以使用登录名 `anonymous`，因此就有了术语“匿名 FTP 服务器”。

使用匿名 FTP 以及设置匿名 FTP 服务器并不在本手册的介绍范围之内。但是，许多书籍（如《 *The Whole Internet User's Guide & Catalog* 》等）都详细介绍了匿名 FTP。有关使用 FTP 的说明，请参见《系统管理指南：网络服务》。`ftp(1)` 手册页介绍了通过命令解释程序调用的所有 `ftp` 命令选项。`ftpd(1M)` 手册页介绍了由 `in.ftpd` 守护进程提供的服务。

- **Telnet**－使用 Telnet 协议，终端和面向终端的进程可以在运行 TCP/IP 的网络上进行通信。此协议在本地系统上作为 `telnet` 程序实现，在远程计算机上则作为 `in.telnetd` 守护进程实现。Telnet 提供了一个用户界面，通过此界面两台主机可进行逐字符或逐行通信。Telnet 包括一组命令，`telnet(1)` 手册页对这些命令进行了全面介绍。
- **TFTP**－简单文件传输协议 (Trivial File Transfer Protocol, `tftp`) 可提供类似于 `ftp` 的功能，但此协议不会建立 `ftp` 的交互式连接。因此，用户无法列出目录内容或更改目录。用户必须知道要复制的文件的全名。`tftp(1)` 手册页介绍了 `tftp` 命令集。

UNIX "r" 命令

使用 UNIX "r" 命令，用户可以在其本地计算机上发出将在远程主机上运行的命令。

这些命令包括：

- `rcp`
- `rlogin`
- `rsh`

有关使用这些命令的说明，请参见 `rcp(1)`、`rlogin(1)` 和 `rsh(1)` 手册页。

名称服务

Oracle Solaris可提供下列名称服务：

- **DNS**－域名系统 (Domain Name System, DNS) 是 Internet 为 TCP/IP 网络提供的名称服务。DNS 为 IP 地址服务提供主机名，另外还可用作数据库进行邮件管理。有关此服务的完整说明，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。另请参见 `resolver(3RESOLV)` 手册页。
- **`/etc` 文件**－最初的基于主机的 UNIX 名称系统是为独立的 UNIX 计算机开发的，后来逐步演变为可以用于网络。许多旧的 UNIX 操作系统和计算机仍在使用此系统，但是此系统并不适用于大型的复杂网络。
- **NIS**－网络信息服务 (Network Information Service, NIS) 是独立于 DNS 开发的，并且其侧重点也稍有不同。DNS 侧重于使用计算机名称而不是数字 IP 地址来简化通信，而 NIS 侧重于对各种网络信息进行集中控制来更好地管理网络。NIS 存储有关计算机名称和地址、用户、网络本身以及网络服务的信息。NIS 名称空间信息存储在 NIS 映射中。有关 NIS 体系结构和 NIS 管理的更多信息，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

目录服务

Oracle Solaris 支持 LDAP（Lightweight Directory Access Protocol，轻量目录访问协议）与 Sun 开放网络环境 (Sun Open Net Environment, Sun ONE) Directory Server 和其他 LDAP 目录服务器一起使用。名称服务和目录服务之间的区别在于功能范围不同。目录服务不仅提供与名称服务相同的功能，而且还提供其他功能。请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

文件服务

NFS 应用层协议可为 Oracle Solaris 提供文件服务。有关 NFS 服务的完整信息，请参见《系统管理指南：网络服务》。

网络管理

使用简单网络管理协议 (Simple Network Management Protocol, SNMP)，可以查看网络的布局和关键计算机的状态。使用 SNMP，还可以通过基于图形用户界面 (Graphical User Interface, GUI) 的软件获取复杂的网络统计信息。许多公司都提供了实现 SNMP 的网络管理软件包。

路由协议

路由信息协议 (Routing Information Protocol, RIP) 和路由器搜索 (Router Discovery, RDISC) 服务器协议是 TCP/IP 网络可用的两种路由协议。有关 Oracle Solaris 可用的路由协议的完整列表，请参阅表 5–1 和表 5–2。

TCP/IP 协议如何处理数据通信

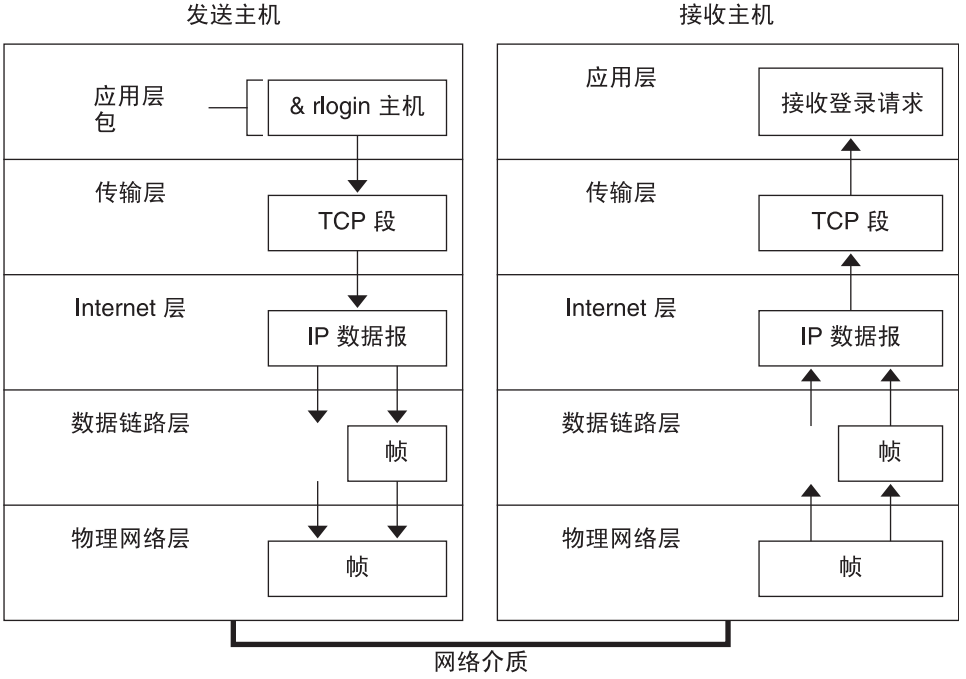
当用户发出使用 TCP/IP 应用层协议的命令时，即会启动一系列事件。用户命令或消息通过本地系统上的 TCP/IP 协议栈进行传递，然后，通过网络介质传递到远程系统上的协议。发送主机的每一层上的协议都会向原始数据添加信息。

发送主机的每一层上的协议还会与接收主机上的对等协议进行交互。图 1–1 显示了这种交互。

数据封装和 TCP/IP 协议栈

包是指通过网络传输的基本信息单元。基本的包由头（包含发送系统和接收系统的地址）和正文或**有效负荷**（包含要传送的数据）组成。当包经由 TCP/IP 协议栈时，每一层上的协议都会在基本头中添加或删除字段。当发送系统上的协议向包头中添加数据时，此过程即被称为**数据封装**。此外，每一层对于已更改的包都有不同的称呼，如下图中所示。

图 1–1 包如何经由 TCP/IP 栈



本节概述了包的生命周期。发出命令或发送消息时，生命周期即会开始。接收系统上的相应应用程序收到包时，生命周期即会完成。

应用层：通信的起源

当一个系统上的用户发送消息或发出必须访问远程系统的命令时，即会开始包的历史记录。应用协议会设置包的格式，以便相应的传输层协议（TCP 或 UDP）可以对包进行处理。

假定用户发出 `rlogin` 命令以登录到远程系统，如图 1–1 中所示。`rlogin` 命令会使用 TCP 传输层协议。TCP 希望以包含命令信息的字节流的形式接收数据。因此，`rlogin` 可将此数据作为 TCP 流进行发送。

传输层：数据封装开始的位置

当数据到达传输层时，该层上的协议即会开始数据封装过程。传输层会将应用程序数据封装到传输协议数据单元中。

传输层协议可在发送应用程序和接收应用程序（以传输端口号区分）之间创建虚拟数据流。端口号可标识**端口**，端口是内存中接收或发送数据的专用位置。此外，传输协议层可能还会提供其他服务，如可靠的、按顺序的数据传送。最终结果取决于是 TCP、SCTP 还是 UDP 处理信息。

TCP 段

TCP 通常称为“面向连接的”协议，这是因为 TCP 可确保将数据成功传送到接收主机。图 1–1 说明了 TCP 协议如何接收来自 `rlogin` 命令的流。然后，TCP 将从应用层收到的数据分成多个段，再向每个段中附加一个头。

段头包含发送端口和接收端口、段排序信息以及称为**校验和**的一个数据字段。两台主机上的 TCP 协议都会使用校验和数据来确定数据传送是否出现错误。

建立 TCP 连接

TCP 使用段来确定接收系统是否准备好接收数据。当用于发送的 TCP 需要建立连接时，TCP 会将一个称为 *SYN* 的段发送到接收主机上的 TCP 协议。用于接收的 TCP 将返回一个称为 *ACK* 的段以确认是否成功收到段。用于发送的 TCP 会发送另一个 ACK 段，然后继续发送数据。这种控制信息的交换称为**三次握手**。

UDP 包

UDP 是一种“无连接”协议。与 TCP 不同，UDP 不会检查数据是否已到达接收主机。相反，UDP 会将从应用层收到的消息的格式设置为 *UDP* **包**。UDP 会向每个包中附加一个头。此头包含发送端口和接收端口、包含包长度的字段以及校验和。

发送 UDP 进程会尝试将包发送到接收主机上的对等 UDP 进程。应用层将确定接收 UDP 进程是否会确认包的接收。UDP 不需要任何接收通知。UDP 不使用三次握手。

Internet 层：准备传送包的位置

传输协议 TCP、UDP 和 SCTP 会将其段和包向下传递到 Internet 层，IP 协议将在该位置处理这些段和包。IP 通过将这些段和包的格式设置为多个称为 *IP* **数据报**的单元，准备对其进行传送。然后，IP 会确定数据报的 IP 地址，以便将其高效地传送到接收主机。

IP 数据报

IP 除了向段或包的头中附加由 TCP 或 UDP 添加的信息之外，还会附加 *IP* **数据包头**。IP 数据包头中的信息包括发送主机和接收主机的 IP 地址、数据报长度以及数据报排序顺序。如果数据报超过网络包允许的字节大小而必须进行分段，则会提供此信息。

数据链路层：成帧位置

数据链路层协议（如 PPP）会将 IP 数据报的格式设置为**帧**。这些协议将附加第三个头和一个脚注，以便对数据报执行“成帧”操作。帧标题包括**循环冗余码校验** (cyclic redundancy check, CRC) 字段，用于检查帧经由网络介质时是否出现错误。然后，数据链路层会将帧传递到物理层。

物理网络层：帧的发送和接收位置

发送主机上的物理网络层会接收帧，并且将 IP 地址转换为适合网络介质的硬件地址。然后，物理网络层会通过网络介质将帧向外发送。

接收主机如何处理包

包到达接收主机时，其经过 TCP/IP 协议栈的顺序与发送时相反。图 1–1 说明了此路径。此外，接收主机上的每种协议还会删除头信息，该信息通过发送主机上的对等协议附加到包中。

将会发生以下过程：

1. 物理网络层接收帧格式的包。物理网络层会计算包的 CRC，然后将帧发送到数据链路层。
2. 数据链路层检验帧的 CRC 是否正确，然后删除帧标题和 CRC。最后，数据链路层将帧发送到 Internet 层。
3. Internet 层读取头中的信息以识别传输。然后，Internet 层将确定包是否为分段包。如果分段进行传输，则 IP 会将分段重新汇编成原始数据报。然后，IP 将删除 IP 数据包头并将数据报传递到传输层协议。
4. 传输层（TCP、SCTP 和 UDP）读取头以确定必须接收数据的应用层协议。然后，TCP、SCTP 或 UDP 将删除其相关的头。TCP、SCTP 或 UDP 将消息或流发送到接收应用程序。
5. 应用层接收消息。然后，应用层将执行发送主机所请求的操作。

TCP/IP 内部跟踪支持

TCP/IP 通过在 RST 包终止连接时记录 TCP 通信来提供内部跟踪支持。传送或接收 RST 包时，所传送的 10 个包中的信息将与连接信息一起记录。

有关 TCP/IP 和 Internet 的更多参考信息

有关 TCP/IP 和 Internet 的信息随处可见。如果您需要了解本文中没有涉及到的特定信息，或许可以在以下所列出的参考资料中找到感兴趣的内容。

有关 TCP/IP 的计算机书籍

本地库或计算机书库中提供了有关 TCP/IP 和 Internet 的多本公开发行的书籍。

以下两本书被视为 TCP/IP 方面的经典著作：

- 由 Craig Hunt 编著的《TCP/IP Network Administration》－该书包含有关管理异构 TCP/IP 网络的一些原理以及许多实用信息。
- 由 W.Richard Stevens 编著的《TCP/IP Illustrated, Volume I》－该书详细说明了 TCP/IP 协议。该书对于网络程序员以及要求具有 TCP/IP 技术背景的网络管理员来说，是理想的参考资料。

与 TCP/IP 和联网相关的 Web 站点

Internet 拥有大量致力于 TCP/IP 协议及其管理的 Web 站点和用户组。许多制造商（包括 Oracle Corporation）都为一般 TCP/IP 信息提供了基于 Web 的资源。以下是有关 TCP/IP 信息和一般系统管理信息的一些有用的 Web 资源。此表列出了相关的 Web 站点并说明这些站点提供的网络信息。

Web 站点	说明
Internet 工程任务组 (Internet Engineering Task Force, IETF) Web 站点 (`http://www.ietf.org/home.html`)	IETF 是负责 Internet 体系结构和管理的机构。IETF Web 站点包含有关该组织的各种活动的信息，另外还包括指向 IETF 主要出版物的链接。
Oracle Corporation 的 BigAdmin 门户网站 (`http://www.oracle.com/technetwork/systems/index.html`)	BigAdmin 提供有关管理 Sun 计算机的信息。此站点提供了常见问题解答、资源、讨论、文档链接以及其他与 Oracle Solaris 管理（包括联网）相关的材料。

RFC 与 Internet 草案

Internet 工程任务组 (Internet Engineering Task Force, IETF) 工作组发布了称为 *Requests for Comments* (RFC) 的标准文档。正在开发的标准会在 Internet 草案中发布。Internet 体系结构委员会 (Internet Architecture Board, IAB) 必须批准所有 RFC 之后，才能将其放入公共域中。通常，RFC 和 Internet 草案面向开发者和其他高级技术读者。但是，许多涉及 TCP/IP 主题的 RFC 都包含适用于系统管理员的重要信息。本书中多处引用了这些 RFC。

通常，仅供参考 (For Your Information, FYI) 文档显示为 RFC 的子集。FYI 包含的信息并不涉及 Internet 标准。FYI 包含更具一般性的 Internet 信息。例如，FYI 文档包括一个书目，其中列出了介绍性的 TCP/IP 书籍和论文。FYI 文档提供了与 Internet 相关的软件工具的详尽纲要。最后，FYI 文档还提供了 Internet 和通用网络术语词汇表。

在本指南以及 Oracle Solaris System Administrator Collection 中的其他书籍中，都存在对相关 RFC 的引用。

第 2 部分

TCP/IP 管理

本部分介绍配置、管理 TCP/IP 网络以及对其进行故障排除的任务和概念性信息。

◆ ◆ ◆ 第 2 章

规划 TCP/IP 网络（任务）

本章介绍了以有组织的成本效益方式创建网络时必须解决的问题。解决了这些问题之后，即可在将来配置和管理网络时制定一个网络计划。

本章包含以下信息：

有关配置网络的任务信息，请参阅第 5 章，配置 TCP/IP 网络服务和 IPv4 寻址（任务）。

网络规划（任务列表）

下表列出了各种配置网络的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
1. 规划您的硬件要求和网络拓扑	确定所需的设备类型以及此设备在站点上的布局。	■ 有关一般网络拓扑问题的信息，请参阅第 48 页中的 “确定网络硬件”。 ■ 有关 IPv6 的拓扑规划，请参阅第 78 页中的 “准备网络拓扑以支持 IPv6”。 ■ 有关特定设备类型的信息，请参阅设备制造商提供的文档。

任务	说明	参考
2. 为网络获取一个已注册的 IP 地址	如果计划与本地网络之外的网络进行通信（例如通过 Internet），则您的网络必须具有一个唯一的 IP 地址。	请参阅第 51 页中的 “获取网络的 IP 号”。
3. 基于 IPv4 网络前缀或 IPv6 站点前缀为系统设计一个 IP 地址寻址方案。	确定如何在站点上部署地址。	请参阅第 49 页中的 “确定网络的 IP 地址寻址格式”或第 82 页中的 “准备 IPv6 寻址计划”。
4. 创建一个包含网络中所有计算机 IP 地址和主机名的列表。	使用此列表可生成网络数据库	请参阅第 56 页中的 “网络数据库”
5. 确定要在网络中使用的名称服务。	决定是使用 NIS、LDAP、DNS，还是使用本地 `/etc` 目录中的网络数据库。	请参阅第 56 页中的 “选择名称服务和目录服务”
6. 在适用于网络的情况下建立管理细分	决定站点是否需要将网络划分为多次管理细分	请参阅第 57 页中的 “管理细分”
7. 在进行网络设计时确定路由器的放置位置。	如果网络足够大而需要路由器，请创建一个支持这些路由器的网络拓扑。	请参阅第 58 页中的 “为网络规划路由器”
8. 为子网制定策略（如果需要）。	您可能需要创建子网，以便管理 IP 地址空间或为用户提供更多 IP 地址。	有关 IPv4 子网规划的信息，请参阅第 211 页中的 “什么是子网划分？” 有关 IPv6 子网规划的信息，请参阅第 82 页中的 “为子网制定编号方案”

确定网络硬件

设计网络时，必须确定哪种网络最能满足组织的需要。必须制定的一些规划决定涉及以下网络硬件：

- 网络硬件的网络拓扑、布局以及连接
- 网络可支持的主机系统数量
- 网络支持的主机类型
- 可能需要的服务器类型
- 要使用的网络介质类型：以太网、令牌环、FDDI（Fiber Distributed Data Interface，光纤分布式数据接口）等
- 是否需要网桥或路由器扩展此介质或将本地网络连接到外部网络
- 除了内置接口之外，某些系统是否还需要另行购买的接口

根据以上因素，即可确定局域网大小。

注 – 如何规划网络硬件不在本手册的介绍范围内。有关帮助信息，请参阅硬件附带的手册。

确定网络的 IP 地址寻址格式

您期望支持的系统数量影响网络配置方式。您的组织可能需要由某一建筑的同一楼层中几十个独立系统所组成的一个小型网络。或者，您可能需要设置一个由分布在多个建筑中的 1,000 个以上系统所组成的网络。此设置要求您进一步将网络划分为多个称为**子网**的分支。

规划网络寻址方案时，请考虑以下因素：

- 要使用的 IP 地址类型：IPv4 或 IPv6
- 网络中可能需要的系统的数量
- 多宿主系统或路由器的数量，这需要多个具有自己的单独 IP 地址的网络接口卡 (Network Interface Card, NIC)
- 是否在网络中使用专用地址
- 是否需要管理 IPv4 地址池的 DHCP 服务器

自 1990 年以来，Internet 在全球范围内的增长已经导致可用的 IP 地址不足。为了改善这种状况，Internet 工程任务组 (Internet Engineering Task Force, IETF) 开发了许多备用 IP 地址寻址方案。

如果您的组织为网络指定了多个 IP 地址或者使用子网，请在组织内部指定一个中心权威机构来指定网络 IP 地址。此权威机构应该维持对已指定的网络 IP 地址池的控制，并且根据需要指定网络、子网和主机地址。为了避免出现问题，请确保在组织中不存在重复或随机的网络号。现在使用的 IP 地址类型包括以下几种：

IPv4 地址

这些 32 位地址是针对 TCP/IP 设计的原始 IP 地址寻址格式。最初，IP 网络包含三个类：A、B 和 C。分配给网络的**网络号**反映了这种类别的指定，另有 8 位或更多位表示主机。基于类的 IPv4 地址要求为网络号配置网络掩码。此外，要为本地网络中的系统提供更多地址，通常还需要将这些地址划分为多个子网。

现在，IP 地址称为 *IPv4* **地址**。虽然不能再从 ISP 获取基于类的 IPv4 网络号，但是很多现有网络仍拥有这些网络号。有关管理 IPv4 地址的更多信息，请参阅第 52 页中的“设计 IPv4 寻址方案”。

CIDR 格式的 IPv4 地址

IETF 开发了无类域间路由 (Classless Inter-Domain Routing, CIDR)，作为在短期到中期内解决 IPv4 地址不足的情况的方法。此外，还设计了 CIDR 格式作为解决全局 Internet 路由表容量匮乏状况的修正方法。使用 CIDR 表示的 IPv4 地址长度为 32 位，并且具有相同的点分十进制格式。但是，CIDR 在最右边的位之后添加了一个前缀标识，用于定义 IPv4 地址的网络部分。有关更多信息，请参阅第 54 页中的 “设计 CIDR IPv4 寻址方案”。

DHCP 地址

通过动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP)，系统可从 DHCP 服务器接收配置信息，其中包括作为引导进程的一部分的 IP 地址。DHCP 服务器维护 IP 地址池，通过该地址池可为 DHCP 客户机指定地址。使用 DHCP 的站点所用的 IP 地址池小于为所有客户机指定永久性 IP 地址时所需的 IP 地址池。您可以设置 DHCP 服务来管理站点的 IP 地址或部分地址。有关更多信息，请参阅第 12 章，关于 DHCP（概述）。

IPv6 地址

IETF 部署了 128 位 IPv6 地址作为可用的 IPv4 地址不足的长期解决方案。IPv6 地址提供的地址空间比 IPv4 可用的地址空间更大。Oracle Solaris 通过使用双堆栈 TCP/IP，可在同一主机上支持 IPv4 和 IPv6 寻址。与 CIDR 格式的 IPv4 地址一样，IPv6 地址也没有网络类或网络掩码的概念。采用 CIDR 格式时，IPv6 地址会使用前缀指定定义站点网络的地址部分。有关 IPv6 的介绍，请参阅第 67 页中的 “IPv6 寻址概述”。

专用地址和文档前缀

IANA 保留了用于专用网络的一个 IPv4 地址块和一个 IPv6 站点前缀。您可以在一个企业网络内的系统上部署这些地址，但是请注意，不能在 Internet 中路由具有专用地址的包。有关专用地址的更多信息，请参阅第 55 页中的 “使用专用 IPv4 地址”。

注 – 用于文档的专用 IPv4 地址也会保留。本书中的示例使用专用 IPv4 地址和保留的 IPv6 文档前缀。

获取网络的 IP 号

IPv4 网络通过 IPv4 网络号加上网络的掩码或**网络掩码**的组合来定义。IPv6 网络通过其**站点前缀**定义；如果划分为子网，则通过其**子网前缀**定义。

除非将网络规划为永久性专用网络，否则本地用户很可能需要在本地网络之外进行通信。因此，必须从适当的组织为网络获取一个已注册的 IP 号，然后才能与外部进行通信。此地址会成为 IPv4 寻址方案的网络号或 IPv6 寻址方案的站点前缀。

Internet 服务提供商可为网络提供 IP 地址，其定价基于不同的服务级别。了解各个 ISP 可确定哪一个提供商可提供最好的网络服务。ISP 通常向企业提供动态分配的地址或静态 IP 地址。某些 ISP 同时提供 IPv4 和 IPv6 地址。

如果您的站点是一个 ISP，则可通过您语言环境的 Internet 注册机构 (Internet Registry, IR) 获取用户的 IP 地址块。Internet 编号分配机构 (Internet Assigned Numbers Authority, IANA) 最终负责将注册的 IP 地址授予世界各地的 IR。每个 IR 都拥有由其提供服务的语言环境的注册信息和模板。有关 IANA 及其 IR 的信息，请参阅 IANA 的 IP 地址服务页面 (`http://www.iana.org/ipaddress/ip-addresses.htm`)。

注 – 请勿随意为网络指定 IP 地址，即使当前未将网络连接到外部 TCP/IP 网络也是如此。相反，应按照第 55 页中的 “使用专用 IPv4 地址”中所述使用专用地址。

设计 IPv4 寻址方案

注 – 有关 IPv6 地址规划的信息，请参阅第 82 页中的 “准备 IPv6 寻址计划”。

本节将概述 IPv4 寻址，以帮助您设计 IPv4 寻址计划。有关 IPv6 地址的信息，请参见第 67 页中的 “IPv6 寻址概述”。有关 DHCP 地址的信息，请参见第 12 章，关于 DHCP（概述）。

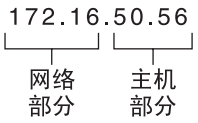
每个基于 IPv4 的网络都必须包含以下内容：

- 由 ISP、IR 指定的唯一网络号；对于旧网络，则为通过 IANA 注册的唯一网络号。如果计划使用专用地址，则设计的网络号在组织中必须是唯一的。
- 网络中每个系统接口的唯一 IPv4 地址。
- 网络掩码。

IPv4 地址是在系统上唯一标识一个网络接口的 32 位数字，如第 55 页中的 “IP 地址如何应用于网络接口”中所述。IPv4 地址以十进制数字表示，分为四个用句点分隔的 8 位字段。每个 8 位字段表示 IPv4 地址的一个字节。这种表示 IPv4 地址字节的形式通常称为**点分十进制格式**。

下图显示了 IPv4 地址 172.16.50.56 的组成部分。

图 2–1 IPv4 地址格式



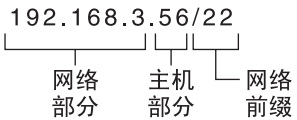
172.16 注册的 IPv4 网络号。在基于类的 IPv4 表示法中，此编号还定义了 IP 网络类，在本示例中为 B 类，应已通过 IANA 进行了注册。

50.56 IPv4 地址的主机部分。主机部分唯一标识网络系统上的一个接口。请注意，对于本地网络中的每个接口，地址的网络部分相同，但是主机部分一定不同。

如果计划将基于类的 IPv4 网络划分为多个子网，则需要定义子网掩码或**网络掩码**，如第 211 页中的 “netmasks 数据库”所述。

以下示例说明了 CIDR 格式的地址 192.168.3.56/22

图 2–2 CIDR 格式的 IPv4 地址



192.168.3 网络部分，由从 ISP 或 IR 收到的 IPv4 网络号组成。

56 主机部分，此部分是为系统接口指定的部分。

/22 网络前缀，定义包含网络号的地址的位数。网络前缀还为 IP 地址提供子网掩码。网络前缀也是由 ISP 或 IR 指定的。

基于 Oracle Solaris 的网络可以将标准 IPv4 地址、CIDR 格式的 IPv4 地址、DHCP 地址、IPv6 地址和专用 IPv4 地址结合起来。

设计 IPv4 寻址方案

本节介绍了标准 IPv4 地址组织而成的类。虽然 IANA 不再分配基于类的网络号，但是这些网络号仍然在许多网络中使用。您也许需要使用基于类的网络号来管理站点的地址空间。有关 IPv4 网络类的完整介绍，请参阅第 223 页中的 “网络类”。

下表说明了标准 IPv4 地址如何划分为网络地址空间和主机地址空间。对于每个类，"Range"（范围）指定网络号第一个字节的十进制值的范围。"Network Address"（网络地址）指示专用于地址的网络部分的 IPv4 地址的字节数。每个字节都由 *xxx* 表示。"Host Address"（主机地址）指示专用于地址的主机部分的字节数。例如，在 A 类网络地址中，第一个字节专用于网络，最后三个字节专用于主机。对于 C 类网络，则应用相反的指定。

表 2–1 IPv4 类的划分

类	字节范围	网络号	主机地址
A	0–127	*xxx*	*xxx.xxx. xxx*
B	128–191	*xxx.xxx*	*xxx.xxx*
C	192–223	*xxx.xxx. xxx*	*xxx*

IPv4 地址的首个字节中的数字定义此网络是 A 类、B 类还是 C 类。其余三个字节范围介于 0–255。编号 0 和 255 为保留编号。您可以为每个字节指定编号 1 至 254，具体取决于 IANA 指定给网络的网络类。

下表显示了为您指定的 IPv4 地址字节。该表还显示了可用于为主机指定的每个字节中的编号范围。

表 2–2 可用 IPv4 类的范围

网络类	字节 1 的范围	字节 2 的范围	字节 3 的范围	字节 4 的范围
A	0–127	1–254	1–254	1–254
B	128–191	由 IANA 预先指定	1–254	1–254
C	192–223	由 IANA 预先指定	由 IANA 预先指定	1–254

IPv4 子网号

有时会将包含大量主机的本地网络划分为多个子网。如果将 IPv4 网络号划分为子网，则需要为每个子网指定一个网络标识符。您可以通过将 IPv4 地址主机部分的某些位用作网络标识符，从而最大程度地提高 IPv4 地址空间的效率。用作网络标识符时，地址的指定部分会成为子网号。可以使用网络掩码来创建子网号，该网络掩码是选择 IPv4 地址的网络和子网部分的位掩码。有关详细信息，请参阅第 212 页中的 “为 IPv4 地址创建网络掩码”。

设计 CIDR IPv4 寻址方案

最初构成 IPv4 的网络类不再在全局 Internet 上使用。现在，IANA 可将无类别 CIDR 格式的地址分发给其世界各地的注册机构。所有从 ISP 获取的 IPv4 地址都采用 CIDR 格式，如图 2–2 所示。

CIDR 地址的网络前缀表示有多少 IPv4 地址可用于网络上的主机。请注意，这些主机地址指定给主机上的接口。如果主机有多个物理接口，则需要为每个正在使用的物理接口指定一个主机地址。

CIDR 地址的网络前缀还定义了子网掩码的长度。大多数 Oracle Solaris 命令可识别网络子网掩码的 CIDR 前缀标识。但是，Oracle Solaris 安装程序和 `/etc/netmask file` 要求使用点分十进制表示法来设置子网掩码。在这两种情况下，请使用 CIDR 网络前缀的点分十进制表示法，如下表所示。

表 2–3 CIDR 前缀及其等效的十进制值

CIDR 网络前缀	可用 IP 地址	等效的点分十进制表示的子网
/19	8,192	255.255.224.0
/20	4,096	255.255.240.0
/21	2,048	255.255.248.0
/22	1024	255.255.252.0
/23	512	255.255.254.0
/24	256	255.255.255.0
/25	128	255.255.255.128
/26	64	255.255.255.192
/27	32	255.255.255.224

有关 CIDR 地址的更多信息，请参阅以下内容：

- 有关 CIDR 的技术详细信息，请参阅 《RFC 1519, Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy》 (`http://www.ietf.org/rfc/rfc1519.txt?number=1519`)（《无类别域间选路：地址分配和聚合策略》）。
- 有关 CIDR 的更多常规信息，请参阅 Pacific Bell Internet 的 《Classless Inter-Domain Routing (CIDR) Overview》 (`http://www.wirelesstek.com/cidr.htm`)（《无类别域间选路概述》）。
- 有关 CIDR 的其他概述信息，请参阅 Wikipedia 文章， 《Classless inter-domain routing》 (`http://en.wikipedia.org/wiki/Classless_inter-domain_routing`)（《无类别域间选路》）。

使用专用 IPv4 地址

IANA 为公司保留了用于其专用网络的三个 IPv4 地址块。这些地址在 RFC 1918, Address Allocation for Private Internets (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`) 中有定义。您可以在公司内联网的本地网络系统上使用这些**专用地址**（也称为 1918 地址）。但是，专用地址在 Internet 上是无效的。请勿在必须与本地网络之外的网络通信的系统上使用它们。

下表列出了 IPv4 专用地址的范围及其相应的掩码。

IPv4 地址范围	网络掩码
10.0.0.0 - 10.255.255.255	10.0.0.0
172.16.0.0 - 172.31.255.255	172.16.0.0
192.168.0.0 - 192.168.255.255	192.168.0.0

IP 地址如何应用于网络接口

要连接到网络，系统必须至少有一个**物理网络接口**。每个网络接口都必须具有其唯一的 IP 地址。安装 Oracle Solaris 的过程中，必须为安装程序找到的第一个接口提供 IP 地址。通常，此接口的名称为 *device-name*0，例如 `eri0` 或 `hme0`。此接口会被视为**主网络接口**。

如果向主机添加第二个网络接口，则此接口也必须具有自己唯一的 IP 地址。添加第二个网络接口后，主机会成为**多宿主主机**。与此相反，向主机添加第二个网络接口并启用 IP 转发时，该主机会成为路由器。有关说明，请参见第 105 页中的 “配置 IPv4 路由器”。

每个网络接口都有设备名称、设备驱动程序以及位于 `/devices` 目录中的关联设备文件。网络接口的设备名称可能为 `eri` 或 `smc0`，这两个设备名称用于两个常用的以太网接口。

有关与接口相关的信息和任务，请参阅第 6 章，管理网络接口（任务）。

注 – 本书假定您的系统具有以太网接口。如果计划使用不同的网络介质，请参阅网络接口附带的手册以获取配置信息。

命名网络中的实体

收到分配的网络 IP 地址并使用此 IP 地址配置系统的所有 NIC 后，下一个任务即是为主机指定名称。然后，必须确定如何处理网络中的名称服务。这些名称最初是在设置网络时使用，然后是在通过路由器、网桥或 PPP 扩展网络时使用。

TCP/IP 协议使用系统的 IP 地址在网络中查找系统。但是，如果使用的是可识别的名称，则可以轻松地标识系统。因此，TCP/IP 协议（和 Oracle Solaris）要求 IP 地址和主机名唯一标识系统。

从 TCP/IP 角度来说，网络是一组命名的实体。主机是具有名称的实体，路由器是具有名称的实体，网络也是具有名称的实体。也可以为安装有网络的组或部门指定名称，这与可为部门、区域或公司指定名称一样。理论上，可用于标识网络的名称分层结构实际没有限制。域名可标识一个**域**。

管理主机名

许多站点允许用户为其计算机选择主机名。服务器也需要至少一个与其主网络接口的 IP 地址关联的主机名。

作为系统管理员，必须确保域中的每个主机名都是唯一的。也就是说，网络中不能有两台计算机都命名为 "fred"。但是，名为 "fred" 的计算机可以有多个 IP 地址。

规划网络时，请创建一个包含 IP 地址及其关联的主机名的列表，以便在设置过程中轻松访问它们。此列表可以帮助检验所有主机名是否唯一。

选择名称服务和目录服务

通过 Oracle Solaris 可以使用三种类型的名称服务：本地文件、NIS 和 DNS。名称服务可维护有关网络中计算机的关键信息，如主机名、IP 地址、以太网地址等。Oracle Solaris 还允许您选择是同时使用 LDAP 目录服务和名称服务，还是使用 LDAP 目录服务而非名称服务。有关 Oracle Solaris 上名称服务的介绍，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》中的第 I 部分, “关于名称和目录服务”。

网络数据库

安装操作系统时，作为此过程的一部分，需要提供服务器、客户机或独立系统的主机名和 IP 地址。Oracle Solaris 安装程序将该信息添加到 `hosts` 和 `ipnodes` 网络数据库（对于 Solaris 10 11/06 及早期 Solaris 10 发行版）。该数据库是包含 TCP/IP 在网络中运行所必需的信息的一组网络数据库的一部分。为网络选择的名称服务可读取这些数据库。

网络数据库的配置非常关键。因此，需要决定作为网络规划过程一部分要使用的名称服务。此外，决定是否使用名称服务还会影响是否将网络组织为管理域。第 214 页中的 “网络数据库和 `nsswitch.conf` 文件”提供了一组网络数据库的详细信息。

使用 NIS 或 DNS 作为名称服务

NIS 和 DNS 名称服务可维护网络上多台服务器中的网络数据库。《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》介绍了这些名称服务并说明如何配置数据库。此外，该指南还详细说明了“名称空间”和“管理域”的概念。

使用本地文件作为名称服务

如果未实现 NIS、LDAP 或 DNS，网络会使用**本地文件**提供名称服务。“本地文件”一词是指 `/etc` 目录中由网络数据库使用的一系列文件。除非另行指明，否则，本书中的过程将假定您使用本地文件作为名称服务。

注 – 如果决定使用本地文件作为网络的名称服务，则以后可以设置其他名称服务。

域名

许多网络会将其主机和路由器组织为管理域的分层结构。如果使用 NIS 或 DNS 名称服务，则必须为组织选择一个全球唯一的域名。要确保域名是唯一的，应向 InterNIC 注册此域名。如果计划使用 DNS，则也需要向 InterNIC 注册域名。

域名结构是分层的。新域通常位于现有的相关域的下方。例如，分公司的域名可位于父公司域名的下方。如果域名没有其他关系，则该组织可以直接将其域名放置在其中一个现有顶级域的下方。

以下是一些顶级域的示例：

- `.com` － 商业公司（国际范围）
- `.edu` － 教育机构（国际范围）
- `.gov` － 美国政府机构
- `.fr` － 法国

您可以选择标识自己组织的名称，并且规定该名称必须是唯一的。

管理细分

管理细分的问题即是对大小和控制进行处理。网络中包含的主机和服务器越多，管理任务就会越复杂。您可能需要通过设置其他管理划分来处理此类情况。如添加特殊的网络类，将现有网络划分为多个子网。

有关为网络设置管理细分的决定由以下因素确定：

- **网络的大小。**

  一次管理划分可以处理包含数百台主机的单一网络，这些主机位于同一物理位置并且需要相同的管理服务。但是，有时您应该建立多次管理细分。如果您有一个包含子网的小型网络，并且该网络分散在一个广泛的地理区域，则这种细分将非常有用。

- **网络中的用户是否有类似需求？**

  例如，您的网络可能局限于单个建筑，并且支持数量相对较少的计算机。这些计算机划分在多个子网中。每个子网都支持有不同需求的用户组。在本示例中，您可能需要针对每个子网使用一次管理细分。

为网络规划路由器

回顾使用 TCP/IP 的情形，网络中有两种类型的实体： 主机和路由器。所有网络都必须包含主机，不过并非所有网络都需要路由器。网络的物理拓扑可确定是否需要路由器。本节介绍了网络拓扑和路由的概念。在决定将其他网络添加到现有网络环境中时，这些概念非常重要。

注 – 有关在 IPv4 网络上配置路由器的完整详细信息和任务，请参阅第 100 页中的 “IPv4 网络上的包转发和路由”。有关在 IPv6 网络上配置路由器的完整详细信息和任务，请参阅第 154 页中的 “配置 IPv6 路由器”。

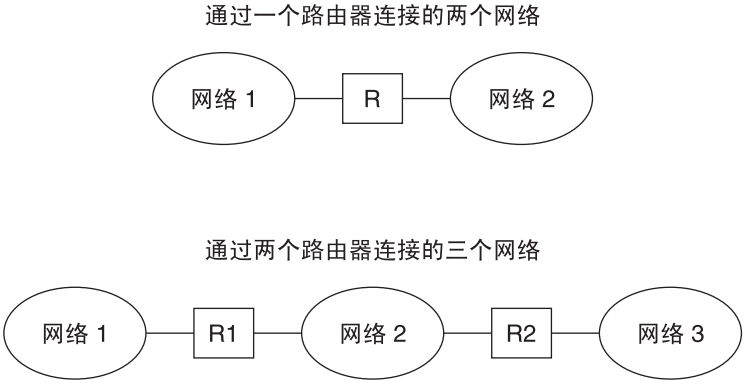
网络拓扑概述

网络拓扑描述了网络如何结合在一起。路由器是指将网络相互连接的实体。路由器是指任何一台具有两个或更多网络接口并实现 IP 转发的计算机。但是，只有正确配置，系统才能用作路由器，如第 105 页中的 “配置 IPv4 路由器”中所述。

路由器可连接两个或更多网络以形成更大的互联网络。必须配置路由器，使其能在两个相邻网络间传送包。路由器还应该可以将包传送到相邻网络以外的网络。

下图显示了网络拓扑的基本部分。第一个图例显示由单个路由器连接的两个网络的简单配置。第二个图例显示由两个路由器互连的三个网络的配置。在第一个示例中，路由器 R 将网络 1 和网络 2 连接成一个大型互联网络。在第二个示例中，路由器 R1 连接网络 1 和 2。路由器 R2 连接网络 2 和 3。这些连接形成了一个包括网络 1、2 和 3 的网络。

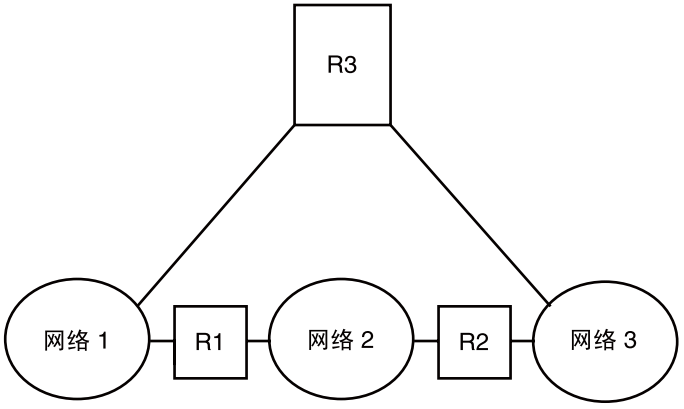
图 2–3 基本网络拓扑



除了将网络连接成互联网络之外，路由器还会在基于目标网络地址的网络间路由包。随着互联网络的日益复杂，每个路由器制定的有关包目标地址的决定也越来越多。

下图显示了一种更复杂的情况。路由器 R3 直接连接网络 1 和 3。冗余性提高了可靠性。如果网络 2 关闭，则路由器 R3 仍会在网络 1 和 3 之间提供路由。您可以互连许多网络。但是，这些网络必须使用相同的网络协议。

图 2–4 在网络间提供其他路径的网络拓扑



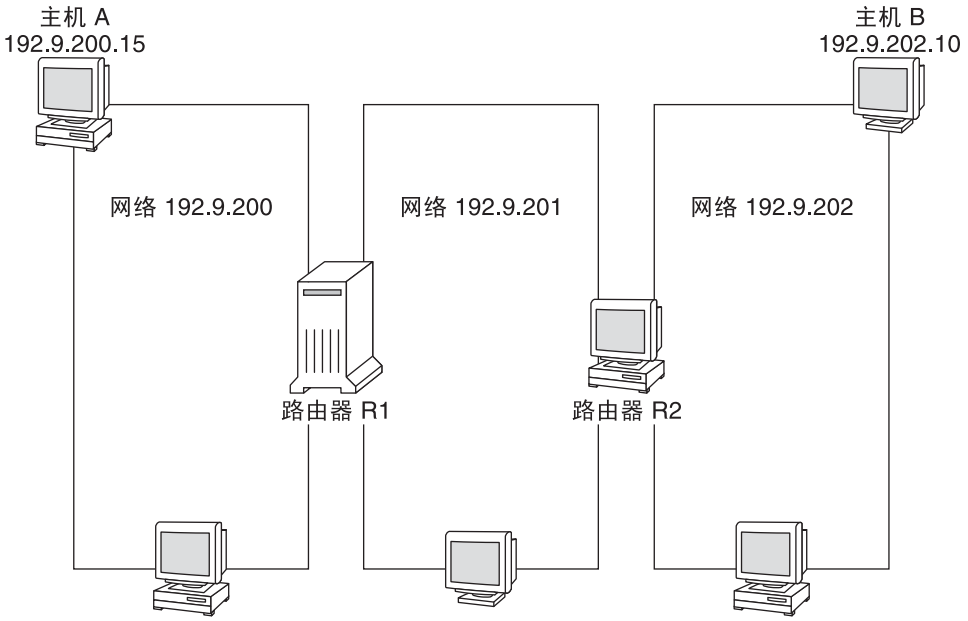
路由器如何传送包

接收者的 IP 地址（包头的一部分）将确定如何对包进行路由。如果此地址包含本地网络的网络号，则包会直接传送到具有此 IP 地址的主机。如果网络号不是指本地网络，则包将传送到本地网络中的路由器。

路由器在**路由表**中维护路由信息。这些表包含路由器连接到的网络中的主机和路由器的 IP 地址。该表还包含指向这些网络的链接。路由器收到包后即会检查路由表，以确定该表是否在标题中列出了目标地址。如果该表不包含目标地址，则路由器会将此包转发到其路由表中列出的其他路由器。有关路由器的详细信息，请参阅第 105 页中的“配置 IPv4 路由器”。

下图中显示了由两个路由器连接的三个网络的网络拓扑。

图 2–5 具有三个互连网络的网络拓扑



路由器 R1 连接网络 `192.9.200` 和 `192.9.201` 。路由器 R2 连接网络 `192.9.201` 和 `192.9.202`。

如果网络 192.9.200 中的主机 A 向网络 192.9.202 中的主机 B 发送消息，则会发生以下事件：

1. 主机 A 通过网络 192.9.200 发送出一个包。包头中包含接收主机 B 的 IPv4 地址 192.9.202.10。
2. 网络 192.9.200 中没有 IPv4 地址为 192.9.202.10 的计算机。因此，路由器 R1 会接受此包。
3. 路由器 R1 检查其路由表。网络 192.9.201 中没有地址为 192.9.202.10 的计算机。但是，路由表确实列出了路由器 R2。
4. R1 随后会选择 R2 作为“下一个跃点”路由器。R1 会将包发送到 R2。
5. 因为 R2 将网络 192.9.201 与 192.9.202 连接，所以 R2 有主机 B 的路由信息。路由器 R2 随后将包转发到网络 192.9.202，主机 B 在此网络中接受包。

◆ ◆ ◆ 第 3 章

IPv6 介绍（概述）

本章概述了 Oracle Solaris Internet 协议版本 6 (Internet Protocol version 6, IPv6) 的实现。此实现包括支持 IPv6 地址空间的相关守护进程和实用程序。

IPv6 和 IPv4 地址可以在 Oracle Solaris 联网环境中共存。配置了 IPv6 地址的系统将保留其原有的 IPv4 地址。涉及 IPv6 地址的操作不会对 IPv4 操作造成不利影响，IPv4 操作也不会对 IPv6 操作造成不利影响。

本章主要讨论以下主题：

有关 IPv6 的更多详细信息，请查阅以下各章：

- IPv6 网络规划－第 4 章，规划 IPv6 网络（任务）
- 与 IPv6 相关的任务－第 7 章，配置 IPv6 网络（任务）和第 8 章，管理 TCP/IP 网络（任务）。
- IPv6 详细信息－第 11 章，IPv6 详解（参考）

IPv6 的主要特征

与 IPv4 相比，IPv6 的显著特征是具有更大的地址空间。IPv6 还在许多方面改进了 Internet 功能，本节将概述这些方面。

扩展的寻址功能

IP 地址大小从 IPv4 中的 32 位增加到 IPv6 中的 128 位，从而可以支持更多层的寻址分层结构。另外，IPv6 提供了更多的可寻址 IPv6 系统。有关更多信息，请参见第 67 页中的 “IPv6 寻址概述”。

地址自动配置和相邻节点搜索

IPv6 **相邻节点搜索** *(Neighbor Discovery, ND)* 协议简化了 IPv6 地址的自动配置。**自动配置**是 IPv6 主机的一个功能，可用来自动生成其自身的 IPv6 地址，从而简化地址管理并缩短管理时间。有关更多信息，请参见第 73 页中的 “IPv6 地址自动配置”。

相邻节点搜索协议对应于以下 IPv4 协议的组合： 地址解析协议 (Address Resolution Protocol, ARP)、Internet 控制消息协议 (Internet Control Message Protocol, ICMP)、路由器搜索 (Router Discovery, RDISC) 和 ICMP 重定向。IPv6 路由器使用相邻节点搜索来通告 IPv6 站点前缀。IPv6 主机使用相邻节点搜索来实现各种目的，包括从 IPv6 路由器请求前缀。有关更多信息，请参见第 72 页中的 “IPv6 相邻节点搜索协议概述”。

简化了包头的格式

IPv6 包头格式要么删除某些 IPv4 包头字段，要么将这些字段设为可选。尽管地址大小增加了，但这种更改却最大程度地减少了 IPv6 包头所占用的带宽。虽然 IPv6 地址长度是 IPv4 地址长度的四倍，但是 IPv6 包头的大小只是 IPv4 包头大小的两倍。

改进了对 IP 数据包头选项的支持

更改了 IP 数据包头选项的编码方式，从而提高了转发效率。而且，对 IPv6 选项长度的限制也不那么严格。这种更改为以后引入新选项提供了更大的灵活性。

对 IPv6 寻址提供了应用程序支持

许多关键的 Oracle Solaris 网络服务都能够识别和支持 IPv6 地址，例如：

- DNS、LDAP 和 NIS 等名称服务。有关这些名称服务对 IPv6 的支持的更多信息，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。
- IP 安全体系结构 (IPsec) 和 Internet 密钥交换 (Internet Key Exchange, IKE) 等验证和保密性应用程序。有关更多信息，请参见第 4 部分。
- 由 IP 服务质量 (IP Quality of Service, IPQoS) 提供的区别服务。有关更多信息，请参见第 6 部分。
- 由 IP 网络多路径 (IP Network Multipathing, IPMP) 提供的故障转移检测。有关更多信息，请参见第 5 部分。

其他 IPv6 资源

除本部分外，您还可以从以下两节列出的资源中获取有关 IPv6 的信息。

IPv6 RFC 与 Internet 草案

有许多与 IPv6 有关的 RFC。下表列出了本书截稿时已有的主要 IPv6 文章及其 Internet 工程任务组 (Internet Engineering Task Force, IETF) Web 的位置。

表 3–1 与 IPv6 相关的 RFC 和 Internet 草案

RFC 或 Internet 草案	主题	位置
RFC 2461，Neighbor Discovery for IP Version 6 (IPv6)	描述 IPv6 相邻节点搜索协议的特征和功能	http://www.ietf.org/rfc/rfc2461.txt$number=2461 (http://www.ietf.org/rfc/rfc2461.txt?number-2461)
RFC 3306，Unicast-Prefix-Based IPv6 Multicast Addresses	描述 IPv6 多点传送地址的格式和类型	ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt (ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt)
RFC 3484， Default Address Selection for Internet Protocol version 6 (IPv6)	描述用于选择 IPv6 缺省地址的算法	http://www.ietf.org/rfc/rfc3484?number=3484 (http://www.ietf.org/rfc/rfc3484.txt?number=3484)
RFC 3513，Internet Protocol version 6 (IPv6) Addressing Architecture	包含有关 IPv6 地址类型的完整详细信息，并提供了很多示例	http://www.ietf.org/rfc/rfc3513.txt?number=3513 (http://www.ietf.org/rfc/rfc3513.txt?number=3513)
RFC 3587，IPv6 Global Unicast Address Format	定义 IPv6 单播地址的标准格式	http://www.ietf.org/rfc/rfc3587.txt?number=3587 (http://www.ietf.org/rfc/rfc3587.txt?number=3587)

Web 站点

下面的 Web 站点提供有关 IPv6 的有用信息。

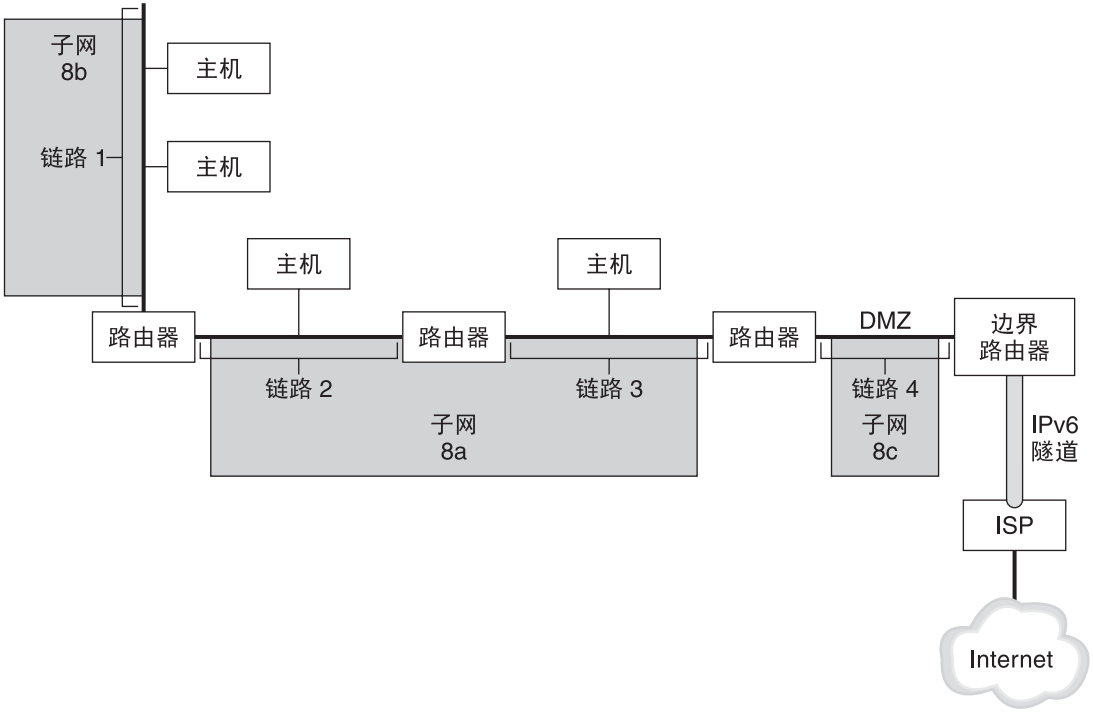
表 3–2 IPv6 相关的 Web 站点

Web 站点	说明	位置
IPv6 论坛	该社区的 Web 站点提供世界各地与 IPv6 相关的演示文稿、活动、课程和实现的链接	http://www.ipv6forum.com
Internet 工程任务组 IPv6 工作组	此 IETF 工作组的主页提供所有相关 IPv6 RFC 和 Internet 草案的链接	http://www.ietf.org/html.charters/ipv6-charter.html

IPv6 网络概述

本节介绍 IPv6 网络拓扑的基本术语。下图显示了 IPv6 网络的基本部分。

图 3–1 IPv6 网络的基本组件



该图描述 IPv6 网络以及它与 ISP 的连接。内部网络由链路 1、链路 2、链路 3 和链路 4 组成。每个链路上安装若干台主机，末端连接一台路由器。链路 4 是网络的 DMZ，它的一端连接边界路由器。边界路由器使用 IPv6 隧道与 ISP 相连，从而为网络提供 Internet 连通性。链路 2 和 3 作为子网 8a 进行管理。子网 8b 仅包含链路 1 上的系统。子网 8c 与链路 4 上的 DMZ 相接。

如图 3–1 中所示，IPv6 网络与 IPv4 网络具有几乎完全相同的组件。但是，IPv6 术语与 IPv4 术语稍有不同。下面是用在 IPv6 上下文中的网络组件的常见术语。

node（节点） 具有 IPv6 地址且接口配置为支持 IPv6 的任何系统。该专业术语适用于主机和路由器。

IPv6 router（IPv6 路由器） 用来转发 IPv6 包的节点。路由器必须至少有一个接口配置为支持 IPv6。IPv6 路由器还可以通过内部网络通告企业的已注册 IPv6 站点前缀。

IPv6 host（IPv6 主机）	具有 IPv6 地址的节点。IPv6 主机可以有多个配置为支持 IPv6 的接口。与 IPv4 主机一样，IPv6 主机也不转发包。
link（链路）	单一且连续的网络介质，其两端均连接有路由器。
neighbor（相邻节点）	与本地节点在同一个链路上的 IPv6 节点。
IPv6 subnet（IPv6 子网）	IPv6 网络的管理段。与 IPv4 子网的组件一样，IPv6 子网的组件也可以直接对应于链路上的所有节点。必要时，可以在单独的子网中对链路上的节点进行管理。另外，IPv6 还支持多链路子网，在多链路子网上，多个链路上的节点可以是同一个子网的组件。图 3–1 中的链路 2 和链路 3 是多链路子网 8a 的组件。
IPv6 tunnel（IPv6 隧道）	在一个 IPv6 节点和另一个 IPv6 节点端点之间提供虚拟的点对点路径的隧道。IPv6 支持可手动配置的隧道和 6to4 自动隧道。
boundary router（边界路由器）	位于网络边界的路由器，是通往本地网络外部端点的 IPv6 隧道的一个端点。此路由器必须至少有一个连接到内部网络的 IPv6 接口。对于外部网络，此路由器可以有一个 IPv6 接口或一个 IPv4 接口。

IPv6 寻址概述

因为一个节点可以有多个接口，所以应将 IPv6 地址指定给接口，而非节点。此外，可以为一个接口指定多个 IPv6 地址。

注 – 有关 IPv6 地址格式的完整技术信息，请参阅 RFC 2374，IPv6 Global Unicast Address Format (`http://www.ietf.org/rfc/rfc2374.txt?number=2374`)

IPv6 定义了以下三种地址类型：

单播 标识单个节点的接口。

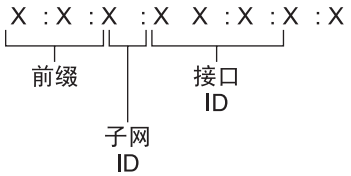
多播 标识一组通常位于不同节点上的接口。发送到多播地址的包将传递到**多播组**的所有成员。

任播 标识一组通常位于不同节点上的接口。发送到任播地址的包将传递到**任播组**中物理位置最接近发送者的成员节点。

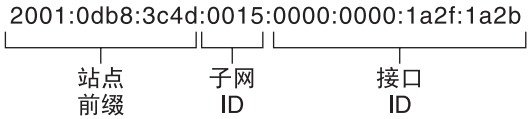
IPv6 地址的各个部分

IPv6 地址的长度为 128 位，由八个 16 位字段组成，相邻字段用冒号分隔。IPv6 地址中的每个字段都必须包含一个十六进制数字，而 IPv4 地址则以点分十进制表示法表示。在下图中，x 表示十六进制数字。

图 3–2 IPv6 地址的基本格式



示例：



最左侧的三个字段（48 位）包含**站点前缀**。站点前缀描述通常由 ISP 或区域 Internet 注册机构 (Regional Internet Registry, RIR) 分配给您的站点的**公共拓扑**。

下一个字段是您（或其他管理员）为您的站点分配的 16 位**子网** *ID*。子网 ID 描述**专用拓扑**（也称为**站点拓扑**），因为它是您的站点的内部 ID。

最右边的四个字段（64 位）包含**接口** *ID*，也称为**标记**。接口 ID 可以从接口的 MAC 地址自动配置，也可以采用 EUI-64 格式手动配置。

请再看一下图 3–2 中的地址：

```
2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b
```

此示例显示了 IPv6 地址的全部 128 位。前 48 位 `2001:0db8:3c4d` 包含表示公共拓扑的站点前缀。随后的 16 位 `0015` 包含代表站点专用拓扑的子网 ID。低阶（最右边的 64 位 `0000:0000:1a2f:1a2b`）包含接口 ID。

缩短 IPv6 地址

大多数 IPv6 地址都不会占用全部 128 位，这可能会导致一些字段会被零填充或仅包含零。

IPv6 寻址体系结构允许您使用两个冒号 (: :) 表示法来表示连续的 16 位零字段。例如，可以通过将接口 ID 中两个连续的零字段替换为两个冒号来缩短图 3–2 中的 IPv6 地址。替换后的地址为 `2001:0db8:3c4d:0015::1a2f:1a2b`。其他零字段可以表示为单个 0。还可以省略字段中的前导零，如将 `0db8` 更改为 `db8`。

因此，地址 `2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b` 可以缩短为 `2001:db8:3c4d:15::1a2f:1a2b`。

可以使用两个冒号替代 IPv6 地址中任意连续的全零字段。例如，IPv6 地址 `2001:0db8:3c4d:0015:0000:d234::3eee:0000` 可以缩短为 `2001:db8:3c4d:15:0:d234:3eee::`。

IPv6 中的前缀

IPv6 地址最左边的字段包含用来路由 IPv6 包的前缀。IPv6 前缀具有以下格式：

prefix/*length in bits*

前缀长度以无类域间路由 (Classless Inter-Domain Routing, CIDR) 表示法声明。CIDR 表示法在地址末尾有一个斜杠，斜杠后跟前缀长度（以位为单位）。有关 CIDR 格式的 IP 地址的信息，请参阅第 54 页中的 “设计 CIDR IPv4 寻址方案”。

IPv6 地址的**站点前缀**最多占用 IPv6 地址最左侧的 48 位。例如，IPv6 地址 `2001:db8:3c4d:0015:0000:0000:1a2f:1a2b/48` 的站点前缀包含在最左边的 48 位 `2001:db8:3c4d` 中。此前缀可使用如下形式（将零省略掉）来表示：

```
2001:db8:3c4d::/48
```

注 – 前缀 `2001:db8::/32` 是专用于文档示例的特殊 IPv6 前缀。

您还可以指定**子网前缀**，该前缀用来定义连接到路由器的网络的内部拓扑。示例 IPv6 地址具有以下子网前缀：

```
2001:db8:3c4d:15::/64
```

子网前缀总是包含 64 位。这些位中有 48 位用于站点前缀，还有 16 位用于子网 ID。

下列前缀已留作特殊用途：

`2002::/16` 指示后跟 6to4 路由前缀。

`fe80::/10` 指示后跟链路本地地址。

`ff00::/8` 指示后跟多播地址。

单播地址

IPv6 包括两种不同的单播地址指定方式：

- 全局单播地址
- 链路本地地址

单播地址的类型由地址中最左边（高阶）的连续位（其中包含前缀）来确定。

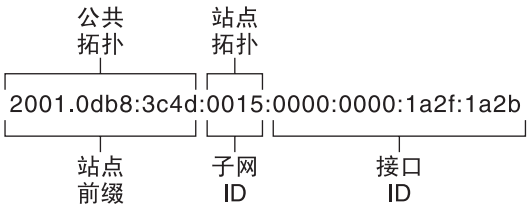
单播地址的格式按以下分层结构进行组织：

- 公共拓扑
- 站点（专用）拓扑
- 接口 ID

全局单点传送地址

全局单播地址在 Internet 中保持全局唯一。第 69 页中的 “IPv6 中的前缀”中的示例 IPv6 地址是全局单播地址。下图显示全局单播地址的范围，它们对应于 IPv6 地址的相应部分。

图 3–3 全局单播地址的各个部分



公共拓扑

站点前缀定义从网络到路由器的**公共拓扑**。企业的站点前缀可以从 ISP 或区域 Internet 注册机构 (Regional Internet Registry, RIR) 获取。

站点拓扑和 IPv6 子网

在 IPv6 中，**子网** *ID* 定义网络的管理子网，它的最大长度为 16 位。可以在配置 IPv6 网络的过程中指定子网 ID。**子网前缀**通过指定已分配了子网的特定链路来定义路由器的站点拓扑。

IPv6 子网在概念上与 IPv4 子网相同，因为每个子网通常都与一个硬件链路相关联。但是，IPv6 子网 ID 用十六进制表示法表示，而不是用点分十进制表示法表示。

接口 ID

接口 *ID* 用来标识特定节点的接口。接口 ID 必须在子网内唯一。IPv6 主机可以使用相邻节点搜索协议自动生成其自身的接口 ID。相邻节点搜索协议基于主机接口的 MAC 地址或 EUI-64 地址自动生成接口 ID。也可以手动指定接口 ID，建议对 IPv6 路由器和启用了 IPv6 的服务器采用这种方式。有关如何创建手动 EUI-3513 地址的说明，请参阅 RFC 3513，Internet Protocol Version 6 (IPv6) Addressing Architecture。

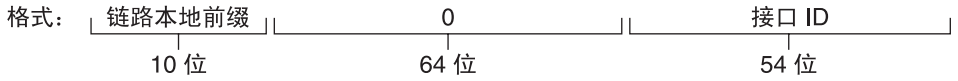
过渡型全局单播地址

为了进行过渡，IPv6 协议提供在 IPv6 地址中嵌入 IPv4 地址这一功能。这种类型的 IPv4 地址便于借助现有的 IPv4 网络隧道传送 IPv6 包。6to4 地址就是一种过渡型全局单播地址。有关 6to4 寻址的更多信息，请参阅第 254 页中的 “6to4 自动隧道”。

链路本地单播地址

链路本地单播地址只能用在本地网络链路上。在企业外部，链路本地地址不但无效，而且无法识别。以下示例显示了链路本地地址的格式。

示例 3–1 链路本地单播地址的各个部分



示例：fe80::123e:456d

链路本地前缀具有以下格式：

`fe80::`*interface-ID*`/10`

下面是链路本地地址的示例：

```
fe80::23a1:b152
```

`fe80` 10 位二进制前缀 1111111010 的十六进制表示形式。此前缀用来将 IPv6 地址的类型标识为链路本地地址。

interface-ID 接口的十六进制地址，通常从 48 位 MAC 地址派生而来。

如果在安装 Oracle Solaris 的过程中启用了 IPv6，则会使用链路本地地址配置本地计算机上编号最小的接口。每个接口都至少需要一个链路本地地址，以便将该节点与本地链路上的其他节点区分开。因此，您需要为节点的其他接口手动配置链路本地地址。完成配置后，该节点会使用其链路本地地址进行自动地址配置和相邻节点搜索。

多点传送地址

IPv6 支持使用多播地址。多播地址用来标识**多播组**，多播组是一组通常位于不同节点上的接口。一个接口可以属于任意数量的多播组。如果 IPv6 地址的前 16 位是 `ff00`*n*，则说明该地址是多播地址。

多播地址用来向定义为多播组成员的所有接口发送信息或服务。例如，使用多播地址与本地链路上的所有 IPv6 节点进行通信。

在创建某个接口的 IPv6 单播地址时，内核会自动使该接口成为某些多播组的成员。例如，内核会使每个节点都成为 "Solicited Node"（请求节点）多播组的成员，相邻节点搜索协议使用该组来检测可访问性。内核还自动使节点成为 "All-Nodes"（所有节点）或 "All Routers"（所有路由器）多播组的成员。

有关多播地址的详细信息，请参阅第 227 页中的 “IPv6 多点传送地址详解”。有关技术信息，请参见 RFC 3306, Unicast-Prefix-based IPv6 Multicast Addresses (`ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt`)，其中介绍了多播地址的格式。有关正确使用多播地址和组的更多信息，请参见 RFC 3307, Allocation Guidelines for IPv6 Multicast Addresses (`ftp://ftp.rfc-editor.org/in-notes/rfc3307.txt`)。

任意点传送地址和组

IPv6 任播地址用来标识一组位于不同 IPv6 节点上的接口。每组接口都称作一个**任播组**。当包发送到任播地址时，任播组中物理位置最接近发送者的成员将收到包。

注 – Oracle Solaris 实现的 IPv6 不支持创建任播地址和任播组。但是，Oracle Solaris IPv6 节点可以将包发送到任播地址。有关更多信息，请参见第 256 页中的 “6to4 中继路由器隧道的注意事项”。

IPv6 相邻节点搜索协议概述

IPv6 引入了相邻节点搜索协议，该协议使用消息传递作为处理相邻节点间交互的方式。**相邻节点**是指在同一链路上的 IPv6 节点。例如，通过发出与相邻节点搜索相关的消息，节点可以获知相邻节点的链路本地地址。

相邻节点搜索控制 IPv6 本地链路上的以下主要活动：

- **路由器搜索** – 帮助主机查找本地链路上的路由器。
- **地址自动配置** – 使节点能够为其接口自动配置 IPv6 地址。
- **前缀搜索** – 使节点能够搜索已分配给链路的已知子网前缀。节点使用前缀来区分位于本地链路上的目标和那些只能通过路由器来访问的目标。

- **地址解析** – 帮助节点确定相邻节点的链路本地地址（如果只给定目标的 IP 地址）。
- **确定下一个跃点** – 使用某种算法来确定本地链路之外的包接受者的跃点的 IP 地址。下一个跃点可以是路由器或目标节点。
- **相邻节点无法访问检测** – 帮助节点确定相邻节点是否不再可以访问。对于路由器和主机，可以重复进行地址解析。
- **重复地址检测** – 使节点能够确定其要使用的地址是否尚未被使用。
- **重定向** – 使路由器能够通知主机要用于到达特定目标的较好的第一个跃点节点。

相邻节点搜索使用下列类型的 ICMP 消息在链路上的节点之间进行通信：

- 路由器请求
- 路由器通告
- 相邻节点请求
- 相邻节点通告
- 重定向

有关相邻节点搜索消息和其他相邻节点搜索协议主题的详细信息，请参阅第 244 页中的 “IPv6 相邻节点搜索协议”。有关相邻节点搜索的技术信息，请参见 RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (`http://www.ietf.org/rfc/rfc2461.txt?number=2461`)。

IPv6 地址自动配置

IPv6 的一个主要特征就是允许主机自动配置接口。通过相邻节点搜索，主机可以在本地链路上查找 IPv6 路由器并请求站点前缀。在自动配置过程中，主机将执行以下操作：

- 为每个接口创建链路本地地址，该操作不要求链路上有路由器。
- 检验地址在链路上是否唯一，该操作不要求链路上有路由器。
- 确定全局地址是应通过无状态机制、有状态机制还是这两种机制来获取。（要求链路上有路由器。）

无状态自动配置概述

无状态自动配置不需要手动配置主机，只需对路由器进行很少的配置（如果需要），而且不需要其他服务器。无状态机制允许主机生成其本身的地址。无状态机制使用本地信息以及由路由器通告的非本地信息来生成地址。

可以为接口实现临时地址，临时地址也是自动配置的。可以为主机上的一个或多个接口启用临时地址标记。但是，与自动配置的标准 IPv6 地址不同，临时地址由站点前缀和一个随机生成的 64 位数字组成。这个随机数将成为 IPv6 地址的接口 ID 部分。临时地址作为接口 ID 时，不会生成链路本地地址。

路由器将通告链路上已指定的所有前缀。IPv6 主机使用相邻节点搜索从本地路由器获取子网前缀。主机通过合并子网前缀和从接口的 MAC 地址生成的接口 ID 来自动生成 IPv6 地址。如果没有路由器，主机可以只生成链路本地地址。链路本地地址只能用于和同一链路上的节点进行通信。

注 – 请不要使用无状态自动配置功能来创建服务器的 IPv6 地址。主机会在自动配置过程中基于特定于硬件的信息来自动生成接口 ID。如果用新接口替换现有的接口，当前的接口 ID 可能会变得无效。

IPv6 隧道概述

对于大多数企业，必须以循序渐进的方式在现有的 IPv4 网络中分步引入 IPv6。Oracle Solaris 双栈网络环境既支持 IPv4 功能又支持 IPv6 功能。因为大多数网络都使用 IPv4 协议，所以 IPv6 网络目前需要一种在其边界外部进行通信的方法。IPv6 网络可以使用隧道来实现这一目的。

在大多数 IPv6 隧道传送方案中，外发 IPv6 包封装在 IPv4 包内部。IPv6 网络的边界路由器可以设置经由各种 IPv4 网络到达目标 IPv6 网络的边界路由器的点对点隧道。包通过隧道到达目标网络的边界路由器，该边界路由器将对包取消封装，然后将单独的 IPv6 包转发到目标节点。

Oracle Solaris IPv6 实现支持下列隧道传送方案：

- 从一个 IPv6 网络经由 IPv4 网络到达另一个 IPv6 网络的手动配置的隧道。IPv4 网络可以是 Internet，也可以是企业内部的本地网络。
- 从一个 IPv4 网络经由 IPv6 网络（通常位于企业内部）到达另一个 IPv4 网络的手动配置的隧道。
- 从一个 IPv6 网络经由企业内的 IPv4 网络或 Internet 到达另一个 IPv6 网络的动态配置的自动 6to4 隧道。

有关 IPv6 隧道的详细信息，请参阅第 250 页中的 “IPv6 隧道”。有关 IPv4 到 IPv4 隧道和 VPN 的信息，请参阅第 438 页中的 “虚拟专用网络和 IPsec”。

◆ ◆ ◆ 第 4 章

规划 IPv6 网络（任务）

在新网络或现有网络上部署 IPv6 需要进行许多规划工作。本章包含在您的站点配置 IPv6 之前必须执行的规划任务。对于现有的网络，IPv6 部署应逐步分阶段进行。本章中的主题有助于您在原本仅使用 IPv4 的网络上分阶段采用 IPv6。

本章讨论以下主题：

有关 IPv6 概念的介绍，请参阅第 3 章，IPv6 介绍（概述）。有关详细信息，请参阅第 11 章，IPv6 详解（参考）。

IPv6 规划（任务列表）

请依次完成以下任务列表中的任务，以完成 IPv6 部署所必须执行的规划任务。

下表列出了各种配置 IPv6 网络的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

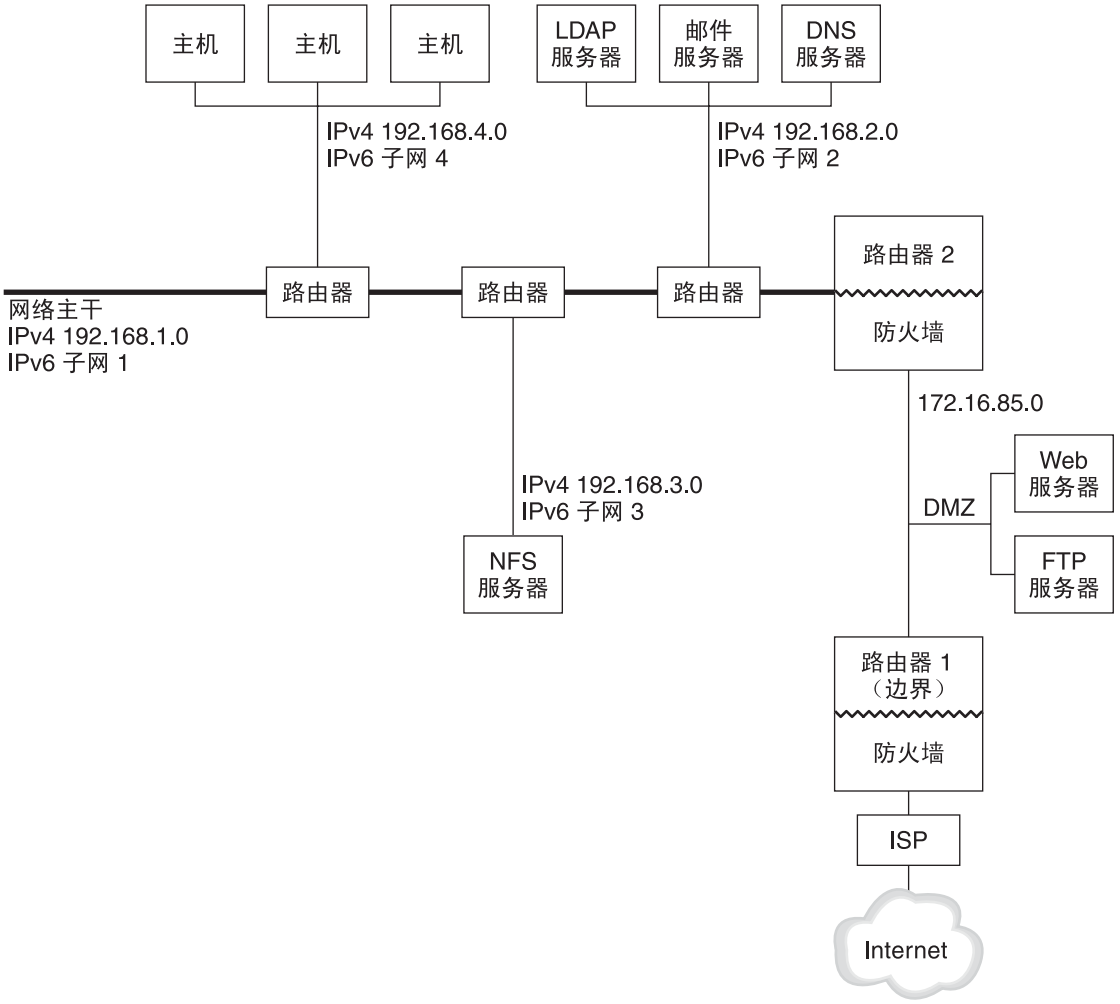
任务	说明	参考
1. 准备硬件以支持 IPv6。	确保硬件可以升级到 IPv6。	第 78 页中的 “准备网络拓扑以支持 IPv6”
2. 找到支持 IPv6 的 ISP。	确保当前的 ISP 支持 IPv6。否则，请寻找可以支持 IPv6 的 ISP。可以使用两个 ISP，一个 ISP 用于 IPv6 通信，另一个用于 ISP IPv4 通信。	
3. 确保应用程序能够支持 IPv6。	验证应用程序是否可以在 IPv6 环境中运行。	第 79 页中的 “如何准备网络服务以支持 IPv6”

任务	说明	参考
4. 获取站点前缀。	从 ISP 或最近的 RIR 获取您站点的 48 位站点前缀。	第 82 页中的 “获取站点前缀”
5. 制定子网寻址计划。	必须先整体规划 IPv6 网络拓扑和寻址方案，然后才能在网络中的各个节点上配置 IPv6。	第 82 页中的 “为子网制定编号方案”
6. 制定隧道使用计划。	确定应当使用哪些路由器来建立与其他子网或外部网络连接的隧道。	第 81 页中的 “在网络拓扑中规划隧道”
7. 为网络上的实体制定寻址计划。	在配置 IPv6 之前，应当先制定好服务器、路由器和主机的寻址计划。	第 83 页中的 “为节点制定 IPv6 寻址计划”
8. 制定 IPv6 安全策略。	在制定 IPv6 安全策略时，需要弄清楚 IP 过滤器、IP 安全体系结构 (IP security architecture, IPsec)、Internet 密钥交换 (Internet Key Exchange, IKE) 和其他 Oracle Solaris 安全功能。	第 4 部分
9.（可选）设置 DMZ。	出于安全方面的考虑，在配置 IPv6 之前，需要为 DMZ 及其实体制定寻址计划。	第 81 页中的 “IPv6 实现的安全注意事项”
10. 使节点能够支持 IPv6。	在所有的路由器和主机上配置 IPv6。	第 154 页中的 “IPv6 路由器配置（任务列表）”
11. 打开网络服务。	确保现有的服务器能够支持 IPv6。	第 177 页中的 “主要的 TCP/IP 管理任务（任务列表）”
12. 更新名称服务器以支持 IPv6。	确保使用新 IPv6 地址更新 DNS、NIS 和 LDAP 服务器。	第 172 页中的 “针对 IPv6 配置名称服务支持”

IPv6 网络拓扑方案

本章中的任务旨在说明如何在典型企业网络上规划 IPv6 服务。下图显示本章中所谈及的网络。建议的 IPv6 网络可能包括该图中显示的部分或全部网络链路。

图 4–1 IPv6 网络拓扑方案



企业网络方案由五个具有现有 IPv4 地址的子网组成。网络的链路直接对应于管理子网。四个内部网络以 RFC 1918 样式的专用 IPv4 地址表示，这是在缺少 IPv4 地址时的常见解决方案。下面是这些内部网络的寻址方案：

- 子网 1 是内部网络主干 `192.168.1`。
- 子网 2 是具有 LDAP、`sendmail` 和 DNS 服务器的内部网络 `192.168.2`。
- 子网 3 是具有企业 NFS 服务器的内部网络 `192.168.3`。
- 子网 4 是包含企业员工主机的内部网络 `192.168.4`。

外部的公共网络 172.16.85 充当企业的 DMZ（隔离区）。此网络中包含 Web 服务器、匿名 FTP 服务器以及企业为外界提供的其他资源。路由器 2 使用防火墙并将公共网络 172.16.85 与内部主干分开。在 DMZ 的另一端，路由器 1 使用防火墙并充当企业的边界服务器。

在图 4–1 中，公共 DMZ 具有 RFC 1918 专用地址 172.16.85。在实际应用中，公共 DMZ 必须具有已注册的 IPv4 地址。大多数 IPv4 站点都使用公共地址和 RFC 1918 专用地址的组合。但是，在引入 IPv6 时，公共地址和专用地址的概念发生了变化。因为 IPv6 具有大得多的地址空间，所以，可以将公共 IPv6 地址同时用于专用网络和公共网络。

准备现有的网络以支持 IPv6

注 – Oracle Solaris 双协议栈支持同时执行 IPv4 操作和 IPv6 操作。在网络上部署 IPv6 期间以及之后，可以成功运行 IPv4 相关的操作。

IPv6 在现有的网络中引入了其他功能。因此，在首次部署 IPv6 时，必须确保不会中断正在使用 IPv4 的任何操作。本节中的主题介绍如何在现有的网络中分步引入 IPv6。

准备网络拓扑以支持 IPv6

IPv6 部署中的第一步就是评估网络上现有的哪些实体能够支持 IPv6。大多数情况下，在实现 IPv6 时，网络拓扑（电缆、路由器和主机）可以保持不变。但是，在实际为网络接口配置 IPv6 地址之前，可能必须针对 IPv6 准备现有的硬件和应用程序。

检验网络上的哪个硬件可以升级到 IPv6。例如，可以就下列类别的硬件，查阅制造商的文档，确定是否已经针对 IPv6 做好准备：

- 路由器
- 防火墙
- 服务器
- 交换机

注 – 本部分中的所有过程都假定您的设备（尤其是路由器）可以升级到 IPv6。

某些型号的路由器无法升级到 IPv6。有关更多信息和解决方法，请参阅第 203 页中的 “IPv4 路由器无法升级到 IPv6”。

准备网络服务以支持 IPv6

在当前的 Oracle Solaris 发行版中，下列典型的 IPv4 网络服务可以支持 IPv6：

- `sendmail`
- NFS
- HTTP（Apache 2.x 或 Orion）
- DNS
- LDAP

IMAP（Internet 消息访问协议）邮件服务仅适用于 IPv4。

针对 IPv6 配置的节点可以运行 IPv4 服务。在打开 IPv6 时，并非所有的服务都能够接受 IPv6 连接。已经移植到 IPv6 的服务将能够接受连接。尚未移植到 IPv6 的服务将使用 IPv4 协议栈。

在将服务升级到 IPv6 之后，可能会出现一些问题。有关详细信息，请参见第 203 页中的 “将服务升级到 IPv6 之后遇到的问题”。

准备服务器以支持 IPv6

因为服务器被视为 IPv6 主机，所以，在缺省情况下，服务器的 IPv6 地址会由相邻节点搜索协议自动配置。但是，许多服务器会有多个网络接口卡 (Network Interface Card, NIC)，您可能希望将它们换出以进行维修或更换。更换 NIC 后，相邻节点搜索会自动为新 NIC 生成一个新的接口 ID。对于特定的服务器，可能不支持此行为。

因此，请考虑为服务器的每个接口手动配置 IPv6 地址的接口 ID 部分。有关说明，请参阅第 161 页中的 “如何配置用户指定的 IPv6 标记”。以后需要更换现有的 NIC 时，已经配置的 IPv6 地址可应用于更换后的 NIC。

▼ 如何准备网络服务以支持 IPv6

1 **更新以下网络服务以支持 IPv6：**

- 邮件服务器
- NIS 服务器
- NFS

> **注 –** LDAP 无需执行特定于 IPv6 的配置任务即可支持 IPv6。

2 **检验防火墙硬件是否能够支持 IPv6。**

有关说明，请参阅与防火墙有关的文档。

3. **检验网络上的其他服务是否已移植到 IPv6。**
   有关更多信息，请参阅软件的营销宣传材料和相关文档。

4. **如果您的站点部署了下列服务，请确保已经针对这些服务采取了相应的措施：**
   - 防火墙
     考虑增强面向 IPv4 的策略以支持 IPv6。有关更多的安全注意事项，请参见第 81 页中的 “IPv6 实现的安全注意事项”。
   - 邮件
     在 DNS 的 MX（邮件交换）记录中，考虑添加邮件服务器的 IPv6 地址。
   - DNS
     有关特定于 DNS 的注意事项，请参见第 80 页中的 “如何准备 DNS 以支持 IPv6”。
   - IPQoS
     在主机上使用先前用于 IPv4 的同一 Diffserv 策略。有关更多信息，请参见第 715 页中的 “分类器模块”。

5. **在将某个节点转换为支持 IPv6 以前，审计由该节点提供的任何网络服务。**

▼ 如何准备 DNS 以支持 IPv6

当前的 Oracle Solaris 发行版在客户端和服务器端均支持 DNS 解析。要使 DNS 服务支持 IPv6，请执行以下准备工作。

有关与准备 DNS 以支持 IPv6 相关的更多信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

1. **确保执行递归名称解析的 DNS 服务器是双栈（IPv4 和 IPv6）服务器或者仅包含 IPv4。**

2. **在 DNS 服务器上，使用转发区域中相关的 IPv6 数据库 AAAA 记录填充 DNS 数据库。**

   注 – 需要特别注意那些运行多个关键服务的服务器。确保网络正常工作，还要确保所有的关键服务都已经移植到 IPv6。然后，将服务器的 IPv6 地址添加到 DNS 数据库中。

3. **向反向区域中添加与 AAAA 记录相关联的 PTR 记录。**

4. **向描述区域的 NS 记录中仅添加 IPv4 数据或者同时添加 IPv6 和 IPv4 数据。**

在网络拓扑中规划隧道

在将网络迁移到 IPv4 和 IPv6 的混合网络时，IPv6 实现支持将许多隧道配置作为转换机制。隧道可以使隔离的 IPv6 网络能够进行通信。因为大多数 Internet 都运行 IPv4，所以，来自您的站点的 IPv6 包需要借助于通往目标 IPv6 网络的隧道在 Internet 上传播。

下面是在 IPv6 网络拓扑中使用隧道的一些主要方案：

- 从其购买 IPv6 服务的 ISP 允许您建立一个从您的站点的边界路由器到 ISP 网络的隧道。图 4–1 显示了这样的隧道。在这种情况下，需要建立 IPv6 over IPv4 手动隧道。
- 管理具有 IPv4 连通性的大型分布式网络。要连接使用 IPv6 的分布式站点，可以从每个子网的边界路由器建立 6to4 自动隧道。
- 有时，基础结构中的某个路由器无法升级到 IPv6。在这种情况下，可以建立将两个 IPv6 路由器作为端点且经由 IPv4 路由器的手动隧道。

有关配置隧道的过程，请参阅第 164 页中的 “针对 IPv6 支持配置隧道所需的任务（任务列表）”。有关隧道的相关概念信息，请参阅第 250 页中的 “IPv6 隧道”。

IPv6 实现的安全注意事项

在现有网络中引入 IPv6 时，必须注意不要危及站点的安全性。在分阶段实现 IPv6 时，需要注意以下安全问题：

- 对于 IPv6 包和 IPv4 包，需要相同的过滤量。
- 通常，IPv6 包通过防火墙进行隧道传送。因此，您应当实现下列任一方案：
  - 让防火墙在隧道内部执行内容检查。
  - 在隧道的另一个端点设置一个具有相似规则的 IPv6 防火墙。
- 在 IPv4 隧道上存在某些使用 IPv6 over UDP 的转换机制。这些机制能够绕过防火墙，因此被认为存在危险。
- IPv6 节点可从企业网络外部进行全局访问。如果安全策略禁止公共访问，则必须为防火墙制定更严格的规则。例如，考虑配置有状态的防火墙。

本书包括可用在 IPv6 实现中的安全功能。

- IP 安全体系结构 (IPsec) 功能允许您为 IPv6 包提供加密保护。有关更多信息，请参阅第 19 章，IP 安全体系结构（概述）。
- Internet 密钥交换 (Internet Key Exchange, IKE) 功能允许您针对 IPv6 包使用公钥验证。有关更多信息，请参阅第 22 章，Internet 密钥交换（概述）。

准备 IPv6 寻址计划

从 IPv4 转换到 IPv6 的主要任务包括制定寻址计划。此任务涉及到进行以下准备：

- 第 82 页中的 “获取站点前缀”
- 第 82 页中的 “制定 IPv6 编号方案”

获取站点前缀

在配置 IPv6 之前，必须获取站点前缀。站点前缀用于派生 IPv6 实现中所有节点的 IPv6 地址。有关站点前缀的介绍，请参阅第 69 页中的 “IPv6 中的前缀”。

支持 IPv6 的任何 ISP 都可以为贵工作单位提供 48 位 IPv6 站点前缀。如果当前的 ISP 仅支持 IPv4，则可以使用另一个 ISP 来支持 IPv6，同时保留当前的 ISP 来支持 IPv4。在这种情况下，您可以使用多种解决方法之一。有关更多信息，请参见第 203 页中的 “当前的 ISP 不支持 IPv6”。

如果贵工作单位是 ISP，则可以从相应的 Internet 注册机构获取客户的站点前缀。有关更多信息，请参见 Internet Assigned Numbers Authority (IANA) (`http://www.iana.org`)（Internet 编号分配机构）。

制定 IPv6 编号方案

除非建议的 IPv6 网络是全新的网络，否则请将现有的 IPv4 拓扑用作 IPv6 编号方案的基础。

为子网制定编号方案

在制定编号方案时，应首先将现有的 IPv4 子网映射到等效的 IPv6 子网。例如，请考虑图 4–1 中所示的子网。子网 1–4 除了用数字 1–4 来指示子网以外，还使用所指定的 RFC 1918 IPv4 专用地址作为其地址的前 16 位。为了进行说明，假定已将 IPv6 前缀 `2001:db8:3c4d/48` 指定给该站点。

下表说明了如何将专用的 IPv4 前缀映射到 IPv6 前缀。

IPv4 子网前缀	等效的 IPv6 子网前缀
`192.168.1.0/24`	`2001:db8:3c4d:1::/64`
`192.168.2.0/24`	`2001:db8:3c4d:2::/64`
`192.168.3.0/24`	`2001:db8:3c4d:3::/64`
`192.168.4.0/24`	`2001:db8:3c4d:4::/64`

为节点制定 IPv6 寻址计划

对于大多数主机，采用无状态自动配置为其接口配置 IPv6 地址是恰当的省时策略。当主机从离其最近的路由器接收到站点前缀时，相邻节点搜索会自动为主机上的每个接口生成 IPv6 地址。

服务器需要具有稳定的 IPv6 地址。如果您未手动配置服务器的 IPv6 地址，那么，更换服务器上的 NIC 卡时，系统会自动配置一个新的 IPv6 地址。在为服务器创建地址时，请记住以下提示：

- 为服务器提供有意义的稳定接口 ID。一个策略就是针对接口 ID 使用连续编号方案。例如，图 4–1 中 LDAP 服务器的内部接口可能会变成 `2001:db8:3c4d:2::2`。
- 或者，如果您不定期为 IPv4 网络重新编号，请考虑使用路由器和服务器现有的 IPv4 地址作为其接口 ID。在图 4–1 中，假定路由器 1 的 DMZ 接口具有 IPv4 地址 `123.456.789.111`。可以将 IPv4 地址转换为十六进制地址，并将结果用作接口 ID。新的接口 ID 将为 `::7bc8:156F`。

  只有当您拥有已注册的 IPv4 地址（而不是从 ISP 获取的地址）时，才使用此方法。如果使用由 ISP 提供给您的 IPv4 地址，则会产生依赖性，而这在更换 ISP 时会造成问题。

由于 IPv4 地址的数量有限，因此，在过去，网络设计者必须考虑在何处使用全局已注册地址和专用 RFC 1918 地址。但是，全局和专用 IPv4 地址的概念并不适用于 IPv6 地址。可以在网络的所有链路（包括公共 DMZ）上使用全局单播地址（包括站点前缀）。

◆ ◆ ◆ 第 5 章

配置 TCP/IP 网络服务和 IPv4 寻址（任务）

TCP/IP 网络管理包括两个阶段。第一个阶段是装配硬件。第二个阶段是配置实现 TCP/IP 协议的守护进程、文件和服务。

本章介绍如何在实现 IPv4 寻址和服务的网络上配置 TCP/IP。

注 – 本章中的许多任务同时适用于仅启用了 IPv4 的网络和启用了 IPv6 的网络。如果这两种寻址格式的配置任务是不同的，则 IPv4 配置步骤在本章中介绍。本章中的任务将交叉引用第 7 章，配置 IPv6 网络（任务）中的与此相当的 IPv6 任务。

本章包含以下信息：

本章新增内容

在 Solaris 10 8/07 中，进行了以下更改：

- 作为使用 `routeadm` 命令的替代方法，可以通过服务管理工具 (Service Management Facility, SMF) 配置和管理路由。有关说明，请参阅第 100 页中的 “IPv4 网络上的包转发和路由”中的过程和示例以及 `routeadm(1M)` 手册页。
- `/etc/inet/ipnodes` 文件已过时。只能对早期 Solaris 10 发行版使用 `/etc/inet/ipnodes`，如以下各个过程中所述。

配置 IPv4 网络之前（ 任务列表 ）

在配置 TCP/IP 之前，请完成下表中列出的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
1. 设计网络拓扑。	确定网络的物理布局。	第 58 页中的 “网络拓扑概述”和第 103 页中的 “IPv4 自治系统拓扑”
2. 从 ISP 或区域 Internet 注册机构 (Regional Internet Registry, RIR) 获取网络号。	获取已注册的网络号，站点上的系统可以使用它与外部进行通信。	第 52 页中的 “设计 IPv4 寻址方案”。
3. 为网络规划 IPv4 寻址方案。如果适用，则包括子网寻址。	将网络号用作寻址计划的基础。	第 52 页中的 “设计 IPv4 寻址方案”。
4. 根据网络拓扑来装配网络硬件。保证硬件正常工作。	设置在网络拓扑设计中概述的系统、网络介质、路由器、交换机、集线器和网桥。	硬件手册和第 58 页中的 “网络拓扑概述”。
5. 将 IPv4 地址和主机名指定给网络中的所有系统。	在 Oracle Solaris 安装过程中或安装后，在适当的文件中指定 IPv4 地址。	第 52 页中的 “设计 IPv4 寻址方案”和第 96 页中的 “如何更改 IPv4 地址和其他网络配置参数”
6. 运行网络接口和路由器所需的配置软件（如果适用）。	配置路由器和多宿主主机。	第 58 页中的 “为网络规划路由器”和第 105 页中的 “配置 IPv4 路由器”（以了解有关路由器的信息）。
7. 确定网络使用的名称服务或目录服务： NIS、LDAP、DNS 或本地文件。	配置选定的名称服务和/或目录服务。	《系统管理指南： 名称和目录服务（DNS、NIS 和 LDAP）》。
8. 选择网络的域名（如果适用）。	选择网络的域名，并向 InterNIC 注册。	《系统管理指南： 名称和目录服务（DNS、NIS 和 LDAP）》

确定主机配置模式

作为网络管理员，您可以将 TCP/IP 配置为在主机和路由器上运行（如果适用）。可以将这些系统配置为从本地系统上的文件或者从网络中其他系统上的文件获取配置信息。

需要以下配置信息：

- 每个系统的主机名
- 每个系统的 IP 地址

- 每个系统所属的域名
- 缺省路由器
- 在每个系统的网络中使用的 IPv4 网络掩码

从本地文件获取 TCP/IP 配置信息的系统在**本地文件模式**下运行。从远程网络服务器获取 TCP/IP 配置信息的系统在**网络客户机模式**下运行。

应该以本地文件模式运行的系统

要以本地文件模式运行，系统必须有 TCP/IP 配置文件的本地副本。第 205 页中的“TCP/IP 配置文件”中介绍了这些文件。系统应有自己的磁盘，尽管严格说来此建议并非必需。

大多数服务器应该以本地文件模式运行。此要求涉及以下服务器：

- 网络配置服务器
- NFS 服务器
- 提供 NIS、LDAP 或 DNS 服务的名称服务器
- 邮件服务器

此外，路由器应该以本地文件模式运行。

专门用作打印服务器的系统无需以本地文件模式运行。单台主机是否应该以本地文件模式运行，取决于网络的规模。

如果运行的网络规模很小，则在单台主机上维护这些文件所涉及的工作量是易于管理的。如果网络为数百台主机服务，则该任务将变得很困难，即使将网络划分为许多管理子域也是如此。因此，对于大型网络，使用本地文件模式通常效率较低。但是，由于路由器和服务器必须是独立的，因此应该以本地文件模式配置它们。

网络配置服务器

网络配置服务器是为配置为网络客户机模式的主机提供 TCP/IP 配置信息的服务器。这些服务器支持以下三种引导协议：

- RARP－反向地址解析协议 (Reverse Address Resolution Protocol, RARP) 将以太网地址（48 位）映射到 IPv4 地址（32 位），它与 ARP 相反。当您在网络配置服务器上运行 RARP 时，以网络客户机模式运行的主机将从服务器获取其 IP 地址和 TCP/IP 配置文件。`in.rarpd` 守护进程启用 RARP 服务。有关详细信息，请参阅 `in.rarpd(1M)` 手册页。
- TFTP－简单文件传输协议 (Trivial File Transfer Protocol, TFTP) 是在远程系统之间传输文件的应用程序。`in.tftpd` 守护进程执行 TFTP 服务，从而允许在网络配置服务器及其网络客户机之间传输文件。有关详细信息，请参阅 `in.tftpd(1M)` 手册页。
- Bootparams－Bootparams 协议提供关闭网络引导的客户机所需的引导参数。`rpc.bootparamd` 守护进程执行这些服务。有关详细信息，请参阅 `bootparamd(1M)` 手册页。

网络配置服务器还可以用作 NFS 文件服务器。

如果要将任何主机配置为网络客户机，还必须将网络上的至少一个系统配置为网络配置服务器。如果网络划分为多个子网，则包含网络客户机的每个子网必须配有至少一个网络配置服务器。

配置为网络客户机的系统

从网络配置服务器获取其配置信息的任何主机都以网络客户机模式运行。已配置为网络客户机的系统不需要 TCP/IP 配置文件的本地副本。

网络客户机模式简化了大型网络的管理。网络客户机模式最大限度地减小了在单台主机上执行的配置任务的数目。网络客户机模式保证网络中的所有系统都遵循相同的配置标准。

可以在所有类型的计算机上配置网络客户机模式。例如，可以在独立系统上配置网络客户机模式。

混合配置

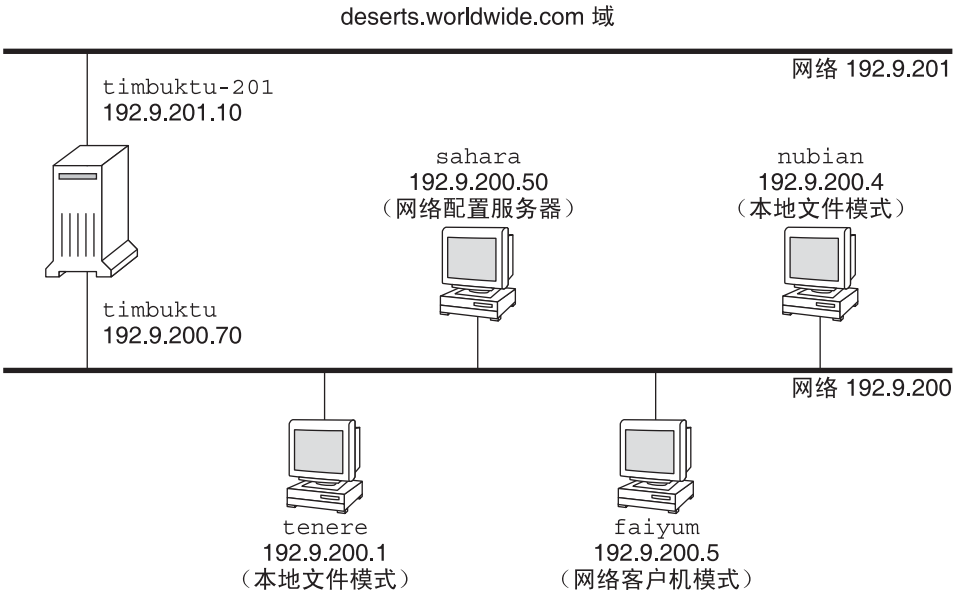
配置并不仅限于纯本地文件模式或纯网络客户机模式。路由器和服务器应该始终以本地文件模式进行配置。对于主机，可以使用本地文件模式和网络客户机模式的任何组合。

IPv4 网络拓扑方案

图 5–1 显示了网络号为 192.9.200的虚构网络上的主机。该网络有一个网络配置服务器，称为 sahara。主机 tenere 和 nubian 有自己的磁盘并以本地文件模式运行。主机 faiyum 也有磁盘，但是此系统以网络客户机模式运行。

最后，系统 timbuktu 被配置为路由器。该系统包括两个网络接口。第一个接口名为 timbuktu。此接口属于网络 192.9.200。第二个接口名为 timbuktu-201。此接口属于网络 192.9.201。这两个网络都位于组织域 deserts.worldwide.com 中。该域将本地文件用作其名称服务。

图 5–1 IPv4 网络拓扑方案中的主机



将子网添加到网络（任务列表）

如果要将不使用子网的网络更改为使用子网的网络，请执行以下任务列表中的任务。

注 – 本节中的信息仅适用于 IPv4 子网。有关规划 IPv6 子网的信息，请参阅第 78 页中的“准备网络拓扑以支持 IPv6”和第 82 页中的“为子网制定编号方案”。

下表列出了各种用于在当前网络中添加子网的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
1. 确定网络拓扑是否需要子网。	确定新子网的拓扑，其中包括路由器和主机在子网中的位置。	第 58 页中的“为网络规划路由器”、第 211 页中的“什么是子网划分？”和第 223 页中的“网络类”
2. 将含有新子网号的 IP 地址指定给要成为子网成员的系统。	在 Oracle Solaris 安装过程中或安装之后，在 `/etc/hostname.`*interface* 文件中配置使用新子网号的 IP 地址。	第 49 页中的“确定网络的 IP 地址寻址格式”

任务	说明	参考
3. 在子网中的所有预期系统上配置子网的网络掩码。	如果要手动配置网络客户机，请修改 /etc/inet/netmasks 文件。或者，将网络掩码提供给 Oracle Solaris 安装程序。	第 211 页中的 “netmasks 数据库”和第 212 页中的 “为 IPv4 地址创建网络掩码”
4. 用子网中所有系统的新 IP 地址编辑网络数据库。	在所有主机上修改 /etc/inet/hosts（对于 Solaris 10 11/06 和早期发行版，则修改 /etc/inet/ipnodes），以反映新主机地址。	第 207 页中的 “hosts 数据库”
5. 重新引导所有系统。		

网络配置任务列表

下表列出了从无子网的网络配置转变到使用子网的网络后还需要执行的额外任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
以本地文件模式配置主机	涉及到编辑 nodename、hostname、hosts、defaultdomain、defaultrouter 和 netmasks 文件。	第 91 页中的 “如何以本地文件模式配置主机”
设置网络配置服务器	涉及到打开 in.tftp 守护进程以及编辑 hosts、ethers 和 bootparams 文件	第 93 页中的 “如何设置网络配置服务器”
以网络客户机模式配置主机	涉及到创建 hostname 文件、编辑 hosts 文件以及删除 nodename 和 defaultdomain 文件（如果它们存在）	第 95 页中的 “如何以网络客户机模式配置主机”
为网络客户机指定路由策略	涉及到确定在主机上使用静态路由还是动态路由。	第 116 页中的 “如何在单接口主机上启用静态路由”和第 118 页中的 “如何在单接口主机上启用动态路由”。
修改现有网络配置	涉及到更改主机名、IP 地址以及在安装时设置的或稍后配置的其他参数。	第 96 页中的 “如何更改 IPv4 地址和其他网络配置参数”

配置本地网络中的系统

网络软件与操作系统软件一起安装。此时，必须将某些 IP 配置参数存储在适当文件中，以便可以在引导时读取它们。

网络配置过程涉及到创建或编辑网络配置文件。如何使配置信息可用于系统内核是有条件的。是否可用取决于这些文件是在本地存储（本地文件模式）还是从网络配置服务器获取（网络客户机模式）。

在网络配置过程中提供的参数如下：

- 每个系统上每个网络接口的 IP 地址。
- 网络中每个系统的主机名。可以在本地文件或名称服务数据库中键入主机名。
- 系统所驻留的 NIS、LDAP 或 DNS 域名（如果适用）。
- 缺省路由器地址。如果在一个简单的网络拓扑中，每个网络仅连接有一个路由器，则可以提供此信息。如果路由器不运行路由协议，如路由器搜索 (Router Discovery, RDISC) 服务器协议或路由器信息协议 (Router Information Protocol, RIP)，则也可以提供此信息。有关缺省路由器的更多信息，请参阅第 100 页中的 “IPv4 网络上的包转发和路由”。有关 Oracle Solaris 支持的路由协议的列表，请参见表 5–1。
- 子网掩码（只有包含子网的网络需要）。

如果 Oracle Solaris 安装程序检测到系统上有多个接口，则可以选择在安装过程中配置其他接口。有关完整说明，请参见《Oracle Solaris 10 1/13 安装指南：基本安装》。

本章包含有关创建和编辑本地配置文件的信息。有关使用名称服务数据库的信息，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

▼ 如何以本地文件模式配置主机

使用此过程可以在以本地文件模式运行的主机上配置 TCP/IP。

有关在 Solaris 10 11/06 及后续发行版中手动配置接口的步骤，请参阅第 129 页中的 “如何在安装系统后配置物理接口”。

1 承担主管理员角色或者成为超级用户

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 转到 `/etc` 目录。

3 **验证在 /etc/nodename 文件中设置的主机名是否正确。**

在 Oracle Solaris 安装过程中指定系统的主机名时，该主机名将输入到 /etc/nodename 文件中。确保节点名称项是系统的正确主机名。

4 **检验系统上的每个网络接口是否存在对应的 /etc/hostname.*interface*文件。**

有关 /etc/hostname.*interface* 文件的文件语法和基本信息，请参阅第 126 页中的 “管理物理接口的基础知识”。

Oracle Solaris 安装程序要求您在安装过程中至少配置一个接口。您配置的第一个接口将自动成为**主网络接口**。安装程序会为主网络接口和在安装时选择配置的任何其他接口创建 /etc/hostname.*interface* 文件。

如果在安装过程中配置了其他接口，请验证每个接口是否有对应的 /etc/hostname.*interface* 文件。在 Oracle Solaris 安装过程中，无需配置多个接口。但是，如果稍后要将更多接口添加到系统中，则必须手动配置它们。

有关在 Solaris 10 11/06 及后续发行版中手动配置接口的步骤，请参阅第 129 页中的 “如何在安装系统后配置物理接口”。

5 **对于 Solaris 10 11/06 和更早的发行版，验证 /etc/inet/ipnodes 文件中的项是否是最新的。**

Solaris 10 安装程序会创建 /etc/inet/ipnodes 文件。此文件包含在安装过程中配置的每个接口的节点名称和 IPv4 地址以及 IPv6 地址（如果适用）。

对 /etc/inet/ipnodes 文件中的项使用以下格式：

IP-address node-name nicknames...

nicknames 是接口的其他名称。

6 **验证 /etc/inet/hosts 文件中的项是否最新。**

Oracle Solaris 安装程序为主网络接口、回送地址和在安装过程中配置的任何其他接口（如果适用）创建项。

a. **确保 /etc/inet/hosts 中的现有项是最新的。**

b. **（可选）为安装后添加到本地主机的任何网络接口添加 IP 地址和对应名称。**

c. **（可选）如果 /usr 文件系统是 NFS 挂载的，则添加文件服务器的一个或多个 IP 地址。**

7 **在 /etc/defaultdomain文件中键入主机的全限定域名。**

例如，假定主机 tenere 是域 deserts.worldwide.com 的一部分。因此应在 /etc/defaultdomain 中键入 deserts.worldwide.com。有关更多信息，请参见第 207 页中的 “/etc/defaultdomain 文件”。

8. **在 /etc/defaultrouter 文件中键入路由器的名称。**
   有关此文件的信息，请参见第 207 页中的 “/etc/defaultrouter 文件”。

9. **在 /etc/inet/hosts 文件中键入缺省路由器的名称及其 IP 地址。**
   还可以使用其他路由选项，如第 95 页中的 “如何以网络客户机模式配置主机”中所述。可以将这些选项应用于本地文件模式配置。

10. **为网络添加网络掩码（如果适用）：**
    - 如果主机从 DHCP 服务器获取其 IP 地址，则不必指定网络掩码。
    - 如果已经在此客户机所在的网络上设置 NIS 服务器，则可以将 netmask 信息添加到该服务器上的相应数据库中。
    - 对于所有其他情况，请执行以下操作：

    a. **在 /etc/inet/netmasks 文件中键入网络号和网络掩码。**
       使用以下格式：

       ```
       network-number netmask
       ```

       例如，对于 C 类网络号 192.168.83，请键入：

       ```
       192.168.83.0    255.255.255.0
       ```

       对于 CIDR 地址，将网络前缀转换为等效的用点分十进制表示法表示的项。网络前缀及其点分十进制等效项可以在表 2–3 中找到。例如，使用以下内容可以表示 CIDR 网络前缀 192.168.3.0/22。

       ```
       192.168.3.0 255.255.252.0
       ```

    b. **在 /etc/nsswitch.conf 中更改网络掩码的查找顺序，以便首先搜索本地文件：**

       ```
       netmasks:   files nis
       ```

11. **重新引导系统。**

▼ 如何设置网络配置服务器

有关如何设置安装服务器和引导服务器的信息，请参见《Oracle Solaris 10 1/13 安装指南：基本安装》。

1. **承担主管理员角色，或成为超级用户。**
   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **转到预期的网络配置服务器的根 (/) 目录。**

3. **通过创建目录 `/tftpboot` 打开 `in.tftpd` 守护进程：**

   ```
   # mkdir /tftpboot
   ```

   此命令将系统配置为 TFTP、bootparams 和 RARP 服务器。

4. **创建指向目录的符号链接。**

   ```
   # ln -s /tftpboot/. /tftpboot/tftpboot
   ```

5. **在 `/etc/inetd.conf` 文件中启用 `tftp` 行。**

   检查该项是否如下所示：

   ```
   tftp dgram udp6 wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
   ```

   此行可防止 `in.tftpd` 检索除位于 `/tftpboot` 中的文件之外的任何文件。

6. **编辑 `hosts` 数据库。**

   添加网络中每个客户机的主机名和 IP 地址。

7. **编辑 `ethers` 数据库。**

   为网络中以网络客户机模式运行的每台主机创建项。

8. **编辑 `bootparams` 数据库。**

   请参见第 219 页中的 “`bootparams` 数据库”。使用通配符项，或者为以网络客户机模式运行的每台主机创建项。

9. **将 `/etc/inetd.conf` 项转换为服务管理工具 (Service Management Facility, SMF) 服务清单，并启用生成的服务：**

   ```
   # /usr/sbin/inetconv
   ```

10. **验证 `in.tftpd` 是否正常工作。**

    ```
    # svcs network/tftp/udp6
    ```

    应该看到与如下所示类似的输出：

    ```
    STATE          STIME    FMRI
    online         18:22:21 svc:/network/tftp/udp6:default
    ```

更多信息 管理 `in.tftpd` 守护进程

`in.tftpd` 守护进程由服务管理工具管理。可以使用 `svcadm` 命令对 `in.tftpd` 执行管理操作（如启用、禁用或重新启动）。启动和重新启动此服务的职责已委托给 `inetd`。使用 `inetadm` 命令可以进行配置更改以及查看 `in.tftpd` 的配置信息。使用 `svcs` 命令可以查询服务的状态。有关服务管理工具的概述，请参阅《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。

配置网络客户机

网络客户机从网络配置服务器接收其配置信息。因此，在将主机配置为网络客户机之前，必须确保至少为网络设置了一个网络配置服务器。

▼ 如何以网络客户机模式配置主机

在要以网络客户机模式配置的每台主机上，执行以下过程。

1 **承担主管理员角色，或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **在 `/etc` 目录中搜索 `nodename` 文件。**

如果存在这样的文件，则删除它。

删除 `/etc/nodename` 会导致系统使用 `hostconfig` 程序从网络配置服务器获取主机名、域名和路由器地址。请参见第 91 页中的 “配置本地网络中的系统”。

3 **如果 `/etc/hostname.`*interface* 文件不存在，则创建它。**

确保该文件为空。`/etc/hostname.`*interface* 文件为空会导致系统从网络配置服务器获取 IPv4 地址。

4 **确保 `/etc/inet/hosts` 文件仅包含回送网络接口的 `localhost` 名称和 IP 地址。**

```
# cat /etc/inet/hosts
# Internet host table
#
127.0.0.1       localhost
```

IPv4 回送接口的 IP 地址为 `127.0.0.1`.

有关更多信息，请参见第 208 页中的 “回送地址”。该文件不应包含本地主机（主网络接口）的 IP 地址和主机名。

5 **检查是否存在 `/etc/defaultdomain` 文件。**

如果存在这样的文件，则删除它。

`hostconfig` 程序自动设置域名。要覆盖由 `hostconfig` 设置的域名，请在 `/etc/defaultdomain` 文件中键入替代域名。

6 **确保客户机的 `/etc/nsswitch.conf` 文件中的搜索路径反映网络的名称服务需求。**

▼ 如何更改 IPv4 地址和其他网络配置参数

此过程说明如何在以前安装的系统上修改 IPv4 地址、主机名和其他网络参数。使用此过程可以修改服务器或联网独立系统的 IP 地址。此过程不适用于网络客户机或设备。这些步骤创建一个在重新引导后继续存在的配置。

注 – 此操作说明仅适用于更改主网络接口的 IPv4 地址。要为系统添加其他接口，请参阅第 129 页中的 “如何在安装系统后配置物理接口”。

在几乎所有情况下，以下步骤都使用传统的 IPv4 点分十进制表示法指定 IPv4 地址和子网掩码。另外，在此过程中也可以使用 CIDR 表示法在所有适用文件中指定 IPv4 地址。有关 CIDR 表示法的简介，请参见第 50 页中的 “CIDR 格式的 IPv4 地址”。

1. **承担主管理员角色，或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **（仅适用于 Solaris 10 11/06 及早期发行版）在 `/etc/inet/ipnodes` 文件或等效 `ipnodes` 数据库中修改 IP 地址。**

   对于要添加到系统的每个 IP 地址，使用以下语法：

   IP-address host-name, nicknames
   IP-address interface-name, nicknames

   第一项应该包含主网络接口的 IP 地址和系统的主机名。可以选择添加主机名的别名。将其他物理接口添加到系统时，在 `/etc/inet/ipnodes` 中为这些接口的 IP 地址和关联名称创建项。

3. **如果必须更改系统的主机名，请在 `/etc/nodename` 文件中修改主机名项。**

4. **在 `/etc/inet/hosts` 文件或等效 `hosts` 数据库中修改 IP 地址和主机名（如果适用）。**

5. **在 `/etc/hostname.`*interface* 文件中修改主网络接口的 IP 地址。**

   可以在 `/etc/hostname.`*interface* 文件中将以下任何项用作主网络接口的项：

   - 用传统的点分十进制格式表示的 IPv4 地址

     使用以下语法：

     IPv4 address subnet mask

     网络掩码项是可选的。如果不指定它，则假定为缺省网络掩码。

     以下是一个示例：

     ```
     # vi hostname.eri0
     10.0.2.5 netmask 255.0.0.0
     ```

- 用 CIDR 表示法表示的 IPv4 地址（如果适合网络配置）。

  IPv4 address/network prefix

  以下是一个示例：

  ```
  # vi hostname.eri0
  10.0.2.5/8
  ```

  CIDR 前缀指定适合 IPv4 地址的网络掩码。例如，上面的 /8 指示网络掩码 255.0.0.0。
- 主机名。

  要在 /etc/hostname.*interface* 文件中使用系统的主机名，请确保主机名和关联的 IPv4 地址也在 hosts 数据库中。

6 如果已更改子网掩码，请在以下文件中修改子网项：

- /etc/netmasks
- （可选）/etc/hostname.*interface*

7 如果已更改子网地址，请在 /etc/defaultrouter 中将缺省路由器的 IP 地址更改为新子网缺省路由器的 IP 地址。

8 重新引导系统。

```
# reboot -- -r
```

示例 5–1 修改要在重新引导后继续存在的 IPv4 地址和其他网络参数

此示例说明如何更改已移动到其他子网的系统的以下网络参数：

- 主网络接口 eri0 的 IP 地址从 10.0.0.14 更改为 192.168.55.14。
- 主机名从 myhost 更改为 mynewhostname。
- 网络掩码从 255.0.0.0 更改为 255.255.255.0。
- 缺省路由器地址更改为 192.168.55.200。

查看系统的当前状态：

```
# hostname
myhost
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
```

接下来，在适当的文件中更改系统的主机名和 eri0 的 IP 地址：

```
# vi /etc/nodename
mynewhostname
```

仅在 Oracle Solaris 10 11/06 以及早期 Oracle Solaris 10 发行版中执行以下操作：

```
# vi /etc/inet/ipnodes
192.168.55.14   mynewhostname      #moved system to 192.168.55 net

# vi /etc/inet/hosts
#
# Internet host table
#
127.0.0.1       localhost
192.168.55.14   mynewhostname        loghost
# vi /etc/hostname.eri0
192.168.55.14   netmask  255.255.255.0
```

最后，更改缺省路由器的网络掩码和 IP 地址。

```
# vi /etc/netmasks
...
192.168.55.0    255.255.255.0

# vi /etc/defaultrouter
192.168.55.200        #moved system to 192.168.55 net
#
```

进行这些更改后，重新引导系统。

```
# reboot -- -r
```

验证在重新引导后是否保持刚设置的配置：

```
# hostname
mynewhostname
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.55.14 netmask ffffff00 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
```

示例 5–2 为当前会话更改 IP 地址和主机名

此示例说明如何仅为当前会话更改主机名、主网络接口的 IP 地址和子网掩码。如果重新引导系统，则系统会恢复到其以前的 IP 地址和子网掩码。主网络接口 `eri0` 的 IP 地址从 `10.0.0.14` 更改为 `192.168.34.100`。

```
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# ifconfig eri0 192.168.34.100 netmask 255.255.255.0 broadcast + up
```

```
# vi /etc/nodename
mynewhostname

# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.34.100 netmask ffffff00 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# hostname
mynewhostname
```

示例 5–3 使用 CIDR 表示法为当前会话更改 IPv4 地址

此示例说明如何使用 CIDR 表示法，仅为当前会话更改主机名和 IP 地址。如果重新引导系统，则系统会恢复到其以前的 IP 地址和子网掩码。主网络接口 `eri0` 的 IP 地址从 `10.0.0.14` 更改为 `192.168.6.25/27`。

```
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# ifconfig eri0 192.168.6.25/27 broadcast + up
# vi /etc/nodename
mynewhostname
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.06.25 netmask ffffffe0 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# hostname
mynewhostname
```

对 IPv4 地址使用 CIDR 表示法时，不必指定网络掩码。`ifconfig` 使用网络前缀标识来确定网络掩码。例如，对于 `192.168.6.0/27` 网络，`ifconfig` 设置网络掩码 `ffffffe0`。如果使用了更常见的 `/24` 前缀标识，则生成的网络掩码是 `ffffff00`。使用 `/24` 前缀标识相当于在配置新 IP 地址时为 `ifconfig` 指定网络掩码 `255.255.255.0`。

另请参见 要更改主网络接口以外的接口的 IP 地址，请参阅《Oracle Solaris 管理：基本管理》和第 129 页中的 “如何在安装系统后配置物理接口”。

IPv4 网络上的包转发和路由

本节包含说明如何在 IPv4 网络上为路由器和主机配置转发和路由的过程和示例。

包转发是在网络上的系统之间共享信息的基本方法。包在源接口和目标接口（通常位于两个不同的系统上）之间进行传送。当您发出命令或将消息发送到非本地接口时，系统将那些包转发到本地网络上。然后，具有包头中所指定目标 IP 地址的接口将从本地网络检索包。如果目标地址不在本地网络上，则将包转发到下一个相邻网络或**跃点**。缺省情况下，在安装 Oracle Solaris 时自动配置包转发。

路由是系统确定要向何处发送包的过程。系统上的路由协议“搜索”本地网络中的其他系统。当源系统和目标系统位于同一本地网络中时，包在它们之间传送的路径称为**直接路由**。如果包必须至少传送到源系统之外的一个跃点，则源系统和目标系统之间的路径称为**间接路由**。路由协议获知目标接口的路径，并将有关已知路由的数据保留在系统的**路由表**中。

路由器是特别配置的系统，具有用于将路由器连接到多个本地网络的多个物理接口。因此，路由器可以将包转发到主 LAN 之外，而不管路由器是否运行路由协议。有关路由器如何转发包的更多信息，请参阅第 58 页中的 “为网络规划路由器”。

路由协议处理系统上的路由活动，并通过与其他主机交换路由信息，维护到远程网络的已知路由。路由器和主机都可以运行路由协议。主机上的路由协议与其他路由器和主机上的路由选择守护进程进行通信。这些协议有助于主机确定向何处转发包。启用网络接口后，系统自动与路由选择守护进程进行通信。这些守护进程监视网络中的路由器，并将路由器的地址通告本地网络中的所有主机。某些路由协议（虽然不是全部）还维护可以用于衡量路由性能的统计信息。与包转发不同，必须在 Oracle Solaris 系统上显式配置路由。

本节介绍在 IPv4 路由器和主机上管理包转发和路由的任务。有关在启用了 IPv6 的网络中路由的信息，请参阅第 154 页中的 “配置 IPv6 路由器”。

Oracle Solaris 支持的路由协议

路由协议分为内部网关协议 (Interior Gateway Protocol, IGP)、外部网关协议 (Exterior Gateway Protocol, EGP) 或这两者的组合。**内部网关协议**通过常见的管理控制在网络中的路由器之间交换路由信息。在图 5–3 所示的网络拓扑中，路由器运行 IGP 以交换路由信息。通过**外部网关协议**，将本地互联网络连接到外部网络的路由器可以与外部网络中的其他路由器交换信息。例如，将公司网络连接到 ISP 的路由器运行 EGP，以便与 ISP 上的相应路由器交换路由信息。边界网关协议 (Border Gateway Protocol, BGP) 是常见的 EGP，用于在不同的组织和 IGP 之间传送路由信息。

下表提供有关 Oracle Solaris 路由协议以及每个协议的关联文档的位置的信息。

表 5–1 Oracle Solaris 路由协议

协议	关联的守护进程	说明	参考
路由信息协议 (Routing Information Protocol, RIP)	`in.routed`	用于路由 IPv4 包和维护路由表的 IGP	第 106 页中的 “如何配置 IPv4 路由器”
Internet 控制消息协议 (Internet Control Message Protocol, ICMP) 路由器搜索	`in.routed`	由主机用来搜索网络上存在的路由器	第 116 页中的 “如何在单接口主机上启用静态路由”和第 118 页中的 “如何在单接口主机上启用动态路由”
下一代路由信息协议 (Routing Information Protocol next generation, RIPng)	`in.ripngd`	用于路由 IPv6 包和维护路由表的 IGP	第 155 页中的 “如何配置启用了 IPv6 的路由器”
相邻节点搜索 (Neighbor Discovery, ND) 协议	`in.ndpd`	通告存在 IPv6 路由器并搜索网络中存在的 IPv6 主机	第 149 页中的 “配置 IPv6 接口”

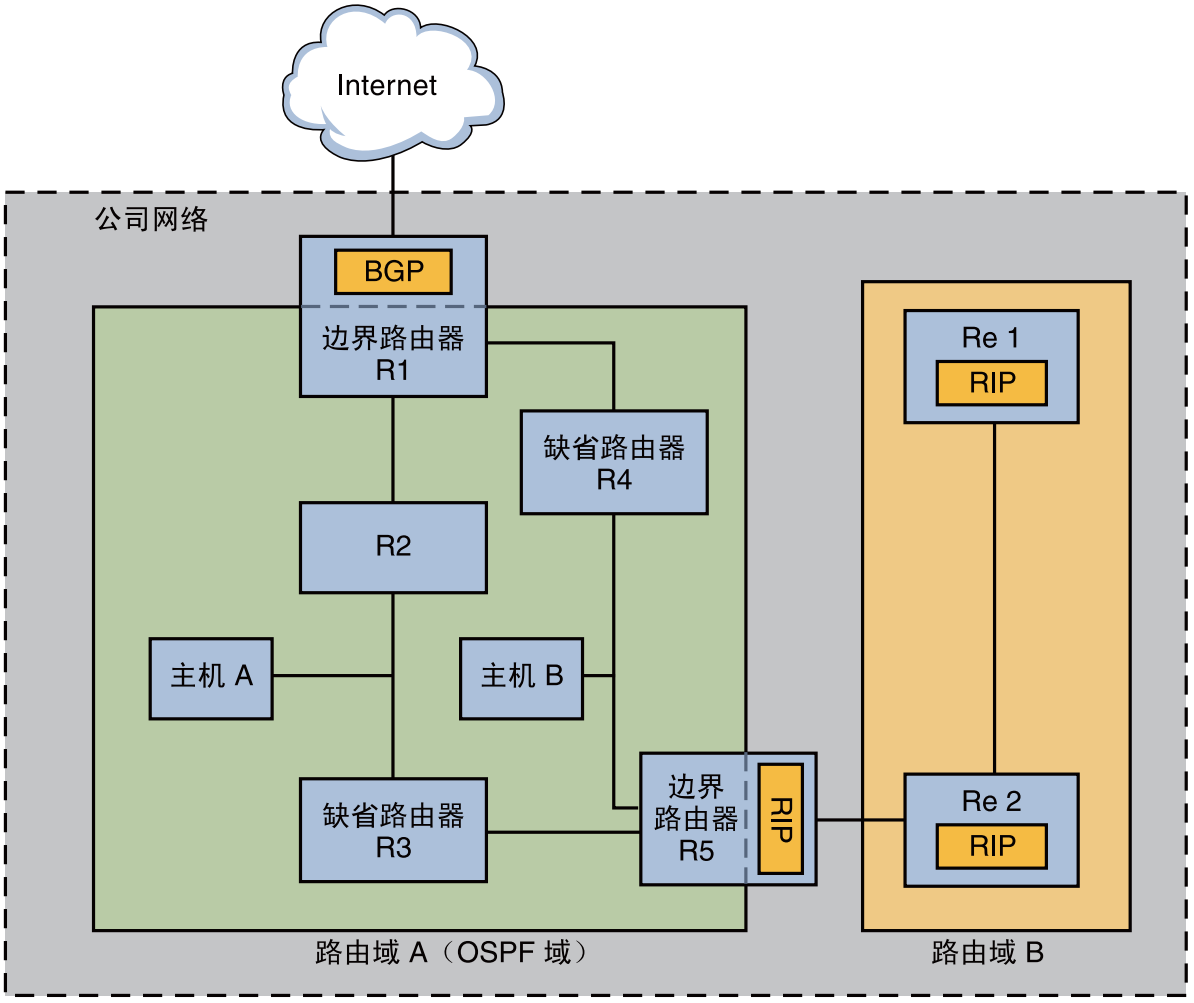
Oracle Solaris 也支持开放源代码 Quagga 路由协议套件。这些协议可以从 SFW 集合磁盘获取，尽管它们不是 Oracle Solaris 主发行版的一部分。下表列出了 Quagga 协议。

表 5–2 Open Source Quagga 协议

协议	守护进程	说明
RIP 协议	`ripd`	路由 IPv4 包并将其路由表通告相邻节点的 IPv4 距离向量 IGP。
RIPng	`ripngd`	IPv6 距离向量 IGP。路由 IPv6 包和维护路由表。
开放最短路径优先 (Open Shortest Path First, OSPF) 协议	`ospfd`	用于包路由和高可用性联网的 IPv4 链路状态 IGP
边界网关协议 (Border Gateway Protocol, BGP)	`bgpd`	用于在管理域之间路由的 IPv4 和 IPv6 EGP。

下图显示使用 Quagga 路由协议的自治系统。

图 5–2 运行 Quagga 协议的公司网络



该图显示了已分为两个路由域（A 和 B）的公司网络自治系统。**路由域**是一个使用统一的路由策略的互联网络（出于管理目的或因为该域使用单个路由协议）。图中的两个域都运行 Quagga 协议套件中的路由协议。

路由域 A 是通过单个 OSPF 域 ID 管理的 OSPF 域。此域中的所有系统都将 OSPF 作为其内部网关协议运行。除了内部主机和路由器外，域 A 还包括两个边界路由器。

边界路由器 R1 将公司网络连接到 ISP 并最终连接到 Internet。为便于公司网络和外界之间的通信，R1 通过其面向外部的网络接口运行 BGP。边界路由器 R5 将域 A 和域 B 连

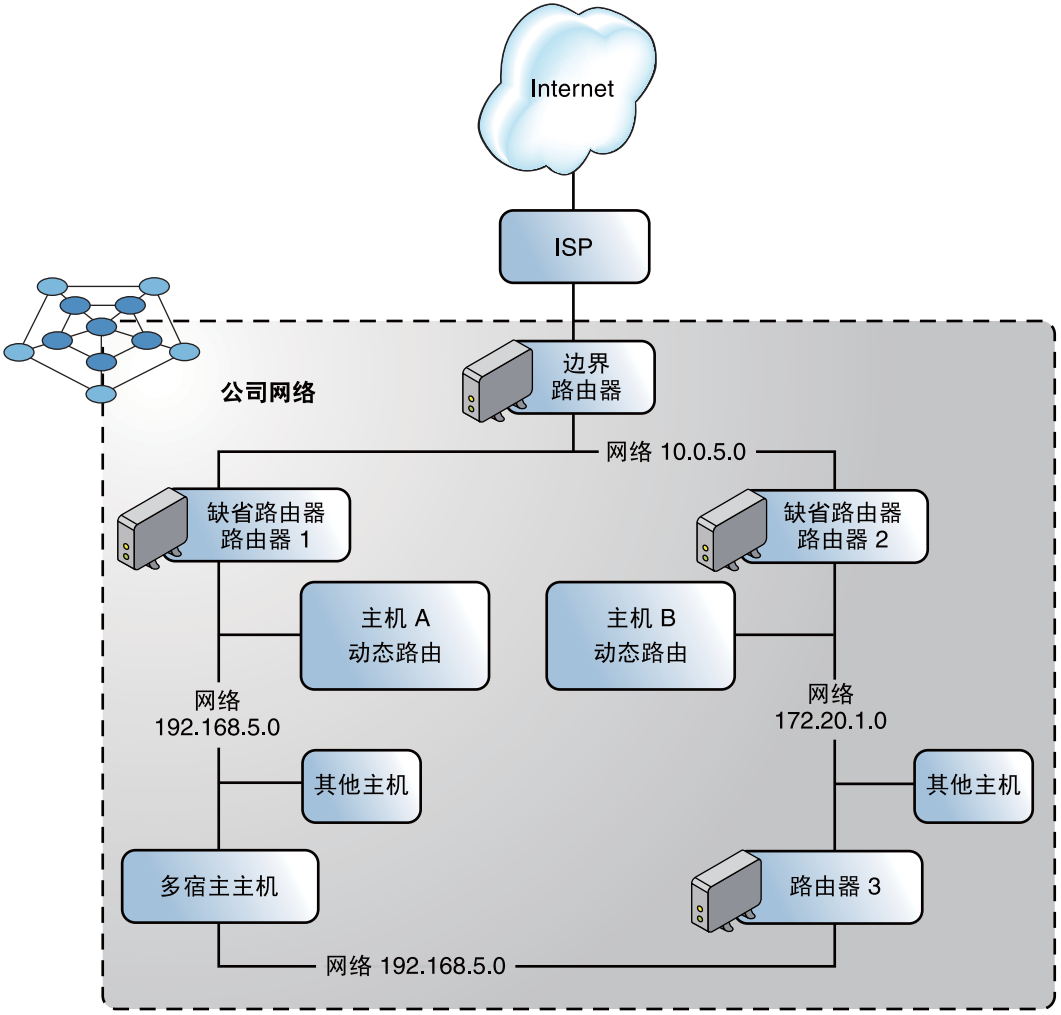
接在一起。域 B 上的所有系统都使用 RIP 进行管理（作为其内部网关协议）。因此，边界路由器 R5 在面向域 A 的接口上必须运行 OSPF，在面向域 B 的接口上必须运行 RIP。

有关 Quagga 协议的更多信息，请访问 Quagga Routing Suite 网站：`http://www.nongnu.org/quagga/index.html`.

IPv4 自治系统拓扑

具有多个路由器和网络的站点通常将其网络拓扑作为单个路由域或**自治系统** *(Autonomous System, AS)* 进行管理。下图显示了一个将被视为小型 AS 的典型网络拓扑。在贯穿本节的示例中引用的就是此拓扑。

图 5–3 具有多个 IPv4 路由器的自治系统



该图显示了一个已划分为三个本地网络（即 10.0.5.0、172.20.1.0 和 192.168.5）的 AS。四个路由器分担包转发和路由职责。AS 包括以下类型的系统：

- **边界路由器**将 AS 连接到外部网络，如 Internet。边界路由器与在本地 AS 上运行的 IGP 的外部网络互连。边界路由器可以运行 EGP，如边界网关协议 (Border Gateway Protocol, BGP)，以与外部路由器（例如 ISP 上的路由器）交换信息。在图 5–3 中，边界路由器的接口连接到内部网络 10.0.5.0 以及服务提供商的高速路由器。

  有关配置边界路由器的信息，请参阅适用于 BGP 的 开放源代码 Quagga 文档 (http://www.quagga.net/docs/docs-info.php#SEC72)。

如果计划使用 BGP 将 AS 连接到 Internet，则应该从适用于您语言环境的 Internet 注册机构获取自治系统编号 (Autonomous System Number, ASN)。区域注册机构，如美国 Internet 编号注册机构 (American Registry for Internet Numbers, ARIN)，提供了有关如何获取 ASN 的指导。例如，《ARIN Number Resource Policy Manual》(https://www.arin.net/policy/nrpm.html#five)（《ARIN 数字资源政策手册》）包含有关在美国和加拿大获取自治系统的 ASN 的说明。或者，您的 ISP 也许能够为您获取 ASN。

- **缺省路由器**维护有关本地网络中所有系统的路由信息。这些路由器通常运行 IGP，如 RIP。在图 5–3 中，路由器 1 的接口连接到内部网络 `10.0.5.0` 和内部网络 `192.168.5`。路由器 1 还充当 `192.168.5` 的缺省路由器。路由器 1 维护 `192.168.5` 中所有系统的路由信息并路由到其他路由器（如边界路由器）。路由器 2 的接口连接到内部网络 `10.0.5.0` 和内部网络 `172.20.1`。

  有关配置缺省路由器的示例，请参阅示例 5–4。
- **包转发路由器**转发包但不运行路由协议。此类型的路由器从其连接到单个网络的接口之一接收包。然后，这些包通过路由器上的其他接口转发到其他本地网络。在图 5–3 中，路由器 3 是连接到网络 `172.20.1` 和 `192.168.5` 的包转发路由器。
- **多宿主主机**具有连接到同一网络段的两个或更多个接口。多宿主主机可以转发包，这是运行 Oracle Solaris 的所有系统的缺省行为。图 5–3 显示了一个多宿主主机，它的两个接口都连接到网络 `192.168.5`。有关配置多宿主主机的示例，请参阅示例 5–6。
- **单接口主机**不仅在包转发方面而且在接收重要配置信息方面依赖于本地路由器。图 5–3 包含 `192.168.5` 网络中实现动态路由的主机 A 和 `172.20.1` 网络中实现静态路由的主机 B。要将主机配置为运行动态路由，请参阅第 118 页中的 “如何在单接口主机上启用动态路由”。要将主机配置为运行静态路由，请参阅第 116 页中的 “如何在单接口主机上启用静态路由”。

配置 IPv4 路由器

本节包含配置 IPv4 路由器的过程和示例。要配置启用了 IPv6 的路由器，请参阅第 155 页中的 “如何配置启用了 IPv6 的路由器”。

由于路由器提供两个或多个网络之间的接口，因此必须为路由器的每个物理网络接口指定唯一名称和 IP 地址。这样，每个路由器都有与其主网络接口关联的主机名和 IP 地址，以及其他每个网络接口的至少一个唯一名称和 IP 地址。

也可以使用以下过程将只有一个物理接口的系统（缺省情况下为主机）配置为路由器。如果单接口系统将要用作 PPP 链路上的一个端点，则可以将它配置为路由器，如《System Administration Guide: Network Services》中的“Planning a Dial-up PPP Link”所述。

注 – 可以在 Oracle Solaris 系统安装过程中配置路由器的所有接口。有关说明，请参见《Oracle Solaris 10 1/13 安装指南：基本安装》。

▼ 如何配置 IPv4 路由器

以下说明假定要在安装后配置路由器的接口。

开始之前 在网络中物理安装路由器后，将路由器配置为以本地文件模式运行，如第 91 页中的“如何以本地文件模式配置主机”中所述。此配置可确保即使网络配置服务器关闭路由器也会引导。

1 在要配置为路由器的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 从 Solaris 10 1/06 发行版开始，使用 `dladm show-link` 命令确定在路由器上物理安装了哪些接口。

```
# dladm show-link
```

`dladm show-link` 的以下示例输出指示，在系统上以物理方式提供了具有四个接口的 `qfe` NIC 和两个 `bge` 接口。

```
qfe0            type: legacy    mtu: 1500       device: qfe0
qfe1            type: legacy    mtu: 1500       device: qfe1
qfe2            type: legacy    mtu: 1500       device: qfe0
qfe3            type: legacy    mtu: 1500       device: qfe1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
```

3 查看在安装过程中配置和检测了路由器上的哪些接口。

```
# ifconfig -a
```

`ifconfig -a` 的以下示例输出显示在安装过程中配置了接口 `qfe0`。此接口位于 `172.16.0.0` 网络中。尚未配置 `qfe` NIC 上的其他接口（即 `qfe1` - `qfe3`）以及 `bge` 接口。

```
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.16.26.232 netmask ffff0000 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:15
```

4 配置并检测另一个接口。

```
# ifconfig interface plumb
```

例如，对于 qfe1，请键入：

```
# ifconfig qfe1 plumb
```

注 – 使用 ifconfig 命令显式配置的接口在重新引导后不会继续存在。

5 将 IPv4 地址和网络掩码指定给接口。

注意 – 可以将 IPv4 路由器配置为通过 DHCP 接收其 IP 地址，但是此建议仅适用于非常有经验的 DHCP 系统管理员。

```
# ifconfig interface IPv4-address netmask netmask
```

例如，要将 IP 地址 192.168.84.3 指定给 qfe1，请执行以下任一操作：

- 如果使用传统的 IPv4 表示法，请键入以下内容：

  ```
  # ifconfig qfe1 192.168.84.3 netmask 255.255.255.0
  ```

- 如果使用 CIDR 表示法，请键入以下内容：

  ```
  # ifconfig qfe1 192.168.84.3/24
  ```

  前缀 /24 自动将 255.255.255.0 网络掩码指定给 qfe1。有关 CIDR 前缀及其点分十进制网络掩码等效项的表，请参阅图 2–2。

6 （可选）要确保在重新引导后接口配置继续存在，请为其他每个物理接口创建 /etc/hostname.*interface*文件。

例如，请创建 /etc/hostname.qfe1 和 /etc/hostname.qfe2 文件。然后在 /etc/hostname.qfe1 文件中键入主机名 timbuktu，在 /etc/hostname.qfe2 中键入主机名 timbuktu-201。有关配置单个接口的更多信息，请参阅第 129 页中的 “如何在安装系统后配置物理接口”。

在创建此文件后，务必进行配置重新引导：

```
# reboot -- -r
```

7 向 /etc/inet/hosts 文件中添加每个接口的主机名和 IP 地址。

例如：

```
172.16.26.232      deadsea        #interface for network 172.16.0.0
192.168.200.20     timbuktu       #interface for network 192.168.200
192.168.201.20     timbuktu-201   #interface for network 192.168.201
192.168.200.9      gobi
192.168.200.10     mojave
192.168.200.110    saltlake
192.168.200.12     chilean
```

接口 timbuktu 和 timbuktu-201 位于同一系统上。请注意，timbuktu-201 的网络地址与 timbuktu 的网络接口不同。之所以不同，是因为网络 192.168.201 的物理网络介质已连接到 timbuktu-201 网络接口，而网络 192.168.200 的介质已连接到 timbuktu 接口。

8 （仅适用于 Solaris 10 11/06 及早期 Solaris 10 发行版）将每个新接口的 IP 地址和主机名添加到 /etc/inet/ipnodes 文件或等效 ipnodes 数据库中。

例如：

```
vi /etc/inet/ipnodes
172.16.26.232      deadsea        #interface for network 172.16.0.0
192.168.200.20     timbuktu       #interface for network 192.168.200
192.168.201.20     timbuktu-201   #interface for network 192.168.201
```

9 如果路由器连接到划分为多个子网的任何网络，请将网络号和网络掩码添加到 /etc/inet/netmasks文件。

- 对于传统的 IPv4 地址表示法（如 192.168.83.0），应键入：

  ```
  192.168.83.0    255.255.255.0
  ```

- 对于 CIDR 地址，在 /etc/inet/netmask 文件的项中使用前缀的点分十进制版本。网络前缀及其点分十进制等效项可以在图 2–2 中找到。例如，可以使用 /etc/netmasks 中的以下项来表示 CIDR 网络前缀 192.168.3.0/22：

  ```
  192.168.3.0 255.255.252.0
  ```

10 在路由器上启用 IPv4 包转发。

使用以下命令之一启用包转发：

- 使用 routeadm 命令，如下所示：

  ```
  # routeadm -e ipv4-forwarding -u
  ```

- 使用以下服务管理工具 (Service Management Facility, SMF) 命令：

  ```
  # svcadm enable ipv4-forwarding
  ```

此时，路由器可以将包转发到本地网络之外。路由器还支持**静态路由**（可以将路由手动添加到路由表的过程）。如果计划在此系统上使用静态路由，则路由器配置已完成。但是，需要在系统路由表中维护路由。有关添加路由的信息，请参见第 111 页中的“配置路由”和 route(1M) 手册页。

11 （可选）启动路由协议。

路由选择守护进程 /usr/sbin/in.routed 自动更新路由表（该过程称为**动态路由**）。使用以下任一方法打开缺省 IPv4 路由协议：

- 使用 routeadm 命令，如下所示：

  ```
  # routeadm -e ipv4-routing -u
  ```

- 使用以下 SMF 命令启动路由协议，如 RIP。

  ```
  # svcadm enable route:default
  ```

与 in.routed 守护进程关联的 SMF FMRI 是 svc:/network/routing/route。

有关 routeadm 命令的信息，请参见 routeadm(1M) 手册页。

示例 5–4 配置网络的缺省路由器

此示例说明如何升级具有多个接口的系统以使其成为缺省路由器。目标是使图 5–3 所示的路由器 2 成为网络 172.20.1.0 的缺省路由器。路由器 2 包含两个有线网络连接，一个是与网络 172.20.1.0 的连接，另一个是与网络 10.0.5.0 的连接。该示例假定路由器在本地文件模式下工作，如第 91 页中的 “如何以本地文件模式配置主机”中所述。

成为超级用户或承担等效角色后，可以确定系统接口的状态。从 Solaris 10 1/06 开始，可以使用 dladm 命令，如下所示：

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1

# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.20.1.10 netmask ffff0000 broadcast 172.20.10.100
        ether 8:0:20:c1:1b:c6
```

dladm show-link 的输出指示有三个链路在系统上是可用的。只有 ce0 接口已使用 IP 地址进行配置。进行缺省路由器配置首先应将 bge0 接口物理连接到 10.0.5.0 网络。然后，检测该接口，并使其在重新引导后继续存在。

```
# ifconfig bge0 plumb
# ifconfig bge0 10.0.5.10/8 up
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.20.1.10 netmask ffff0000 broadcast 172.255.255.255
        ether 8:0:20:c1:1b:c6
bge0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.5.10 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:e5:95:c4
 # vi /etc/hostname.bge0
10.0.5.10
255.0.0.0
```

使用重新配置引导命令，重新引导系统：

```
# reboot -- -r
```

使用有关新检测的接口和它所连接到的网络的信息，继续配置以下网络数据库：

```
# vi /etc/inet/hosts
127.0.0.1       localhost
172.20.1.10        router2        #interface for network 172.20.1
10.0.5.10          router2-out    #interface for network 10.0.5
# vi /etc/inet/netmasks
172.20.1.0    255.255.0.0
10.0.5.0      255.0.0.0
```

最后，使用 SMF 启用包转发，再启用 `in.routed` 路由选择守护进程。

```
# svcadm enable ipv4-forwarding
# svcadm enable route:default
```

现在，在路由器 2 上启用了 IPv4 包转发和通过 RIP 的动态路由。但是，网络 `172.20.1.0` 的缺省路由器配置尚未完成。您需要执行以下操作：

- 修改 `172.10.1.10` 中的每个主机，以便主机从新的缺省路由器获取其路由信息。有关更多信息，请参阅第 116 页中的 “如何在单接口主机上启用静态路由”。
- 在路由器 2 的路由表中定义边界路由器的静态路由。有关更多详细信息，请参阅第 110 页中的 “路由表和路由类型”。

路由表和路由类型

路由器和主机都维护**路由表**。每个系统上的路由选择守护进程都使用所有的已知路由来更新该表。在将包转发到本地网络之前，系统的内核读取路由表。路由表列出了系统知晓的网络的 IP 地址，包括系统本地缺省网络的 IP 地址。该表还列出了每个已知网络的网关系统的 IP 地址。**网关**是一个系统，它可以接收传出包并将它们转发到距本地网络一个跃点的位置。以下是一个仅启用了 IPv4 的网络中某系统的简单路由表：

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.20.1.10          UG        1    532   ce0
224.0.0.0            10.0.5.100           U         1      0   bge0
10.0.0.0             10.0.5.100           U         1      0   bge0
127.0.0.1            127.0.0.1            UH        1     57   lo0
```

可以在 Oracle Solaris 系统上配置以下两种类型的路由：静态路由和动态路由。可以在单个系统上配置其中一种或两种路由类型。实现**动态路由**的系统依赖路由协议（如用于 IPv4 网络的 RIP 和用于 IPv6 网络的 RIPng）来维护其路由表。仅运行**静态路由**的系统不依赖于路由协议来获取路由信息及更新路由表。相反，您必须通过 `route` 命令手动维护系统的已知路由。有关完整的详细信息，请参阅 `route(1M)` 手册页。

为本地网络或自治系统配置路由时，请考虑在特定的路由器和主机上支持哪种路由类型。

下表显示了不同的路由类型，以及各个路由类型分别最适用于哪种网络方案。

路由类型	最适用于
静态	小型网络、从缺省路由器获取其路由的主机，以及仅需要知晓接下来几个跃点上一个或两个路由器的缺省路由器。
动态	较大的互联网络、具有多个主机的本地网络中的路由器以及大型自治系统上的主机。动态路由是大多数网络中系统的最佳选择。
组合的静态和动态路由	将静态路由网络和动态路由网络连接在一起的路由器，以及将内部自治系统与外部网络连接在一起的边界路由器。将系统上的静态路由和动态路由组合在一起是一种常见的做法。

图 5–3 所示的 AS 将静态路由和动态路由组合在一起。

配置路由

要为 IPv4 网络实现动态路由，请使用 `routeadm` 或 `svcadm` 命令启动 `in.routed` 路由选择守护进程。有关说明，请参见第 106 页中的 “如何配置 IPv4 路由器”。动态路由是大多数网络和自治系统的首选策略。但是，您的网络拓扑或您网络中的特定系统可能需要静态路由。在该情况下，必须手动编辑系统路由表，向网关反映已知路由。接下来的过程说明如何添加静态路由。

注 – 到同一目标的两个路由不会自动导致系统进行负载平衡或故障转移。如果需要这些功能，请使用 IPMP，如第 27 章，IPMP 介绍（概述）中所述。

▼ 如何将静态路由添加到路由表

1 查看路由表的当前状态。

使用一般用户帐户运行以下形式的 `netstat` 命令：

```
% netstat -rn
```

输出将与如下所示类似：

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
192.168.5.125        192.168.5.10          U      1   5879   ipge0
224.0.0.0            198.168.5.10          U      1  0       ipge0
default              192.168.5.10          UG     1  91908
127.0.0.1            127.0.0.1             UH     1  811302   lo0
```

2 承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

3 （可选）刷新路由表中的现有项。

```
# route flush
```

4 添加一个在系统重新引导后继续存在的路由。

```
# route -p add -net network-address -gateway gateway-address
```

`-p`	创建一个在系统重新引导后必须继续存在的路由。如果希望路由仅对当前会话有效，则不要使用 `-p` 选项。
`add`	指示将要添加以下路由。
`-net` *network-address*	指定路由将转到具有 *network-address* 中地址的网络。
`-gateway` *gateway-address*	指示指定路由的网关系统具有 IP 地址 *gateway-address*。

示例 5–5 将静态路由添加到路由表

以下示例说明如何将静态路由添加到系统。该系统是路由器 2，即图 5–3 所示的 172.20.1.0 网络的缺省路由器。在示例 5–4 中，为路由器 2 配置了动态路由。为了更好地充当网络 172.20.1.0 中主机的缺省路由器，路由器 2 还需要到 AS 的边界路由器 10.0.5.150 的静态路由。

要查看路由器 2 上的路由表，请执行以下操作：

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.20.1.10          UG        1    249 ce0
224.0.0.0            172.20.1.10          U         1      0 ce0
10.0.5.0             10.0.5.20            U         1     78 bge0
127.0.0.1            127.0.0.1            UH        1     57 lo0
```

路由表指示路由器 2 知晓的两个路由。缺省路由将路由器 2 的 172.20.1.10 接口用作其网关。在路由器 2 上运行的 in.routed 守护进程搜索到第二个路由 10.0.5.0。此路由的网关是 IP 地址为 10.0.5.20 的路由器 1。

要将另一个路由添加到网络 10.0.5.0（它将其网关作为边界路由器），请执行以下操作：

```
# route -p add -net 10.0.5.0/24 -gateway 10.0.5.150
add net 10.0.5.0: gateway 10.0.5.150
```

现在，路由表具有边界路由器（其 IP 地址为 10.0.5.150/24）的路由。

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.20.1.10          UG        1    249 ce0
```

```
224.0.0.0          172.20.1.10        U         1      0 ce0
10.0.5.0           10.0.5.20          U         1     78 bge0
10.0.5.0           10.0.5.150         U         1    375 bge0
127.0.0.1          127.0.0.1          UH        1     57 lo0
```

配置多宿主主机

在 Oracle Solaris中，具有多个接口的系统被视为**多宿主主机**。多宿主主机的接口可以与不同物理网络或同一物理网络中的多个子网连接。

如果一个系统的多个接口连接到同一子网，必须首先将这些接口配置到一个 IPMP 组中。否则，该系统无法成为多宿主主机。有关 IPMP 的更多信息，请参见第 5 部分。

多宿主主机不会转发 IP 包，但可以配置为运行路由协议。通常，可以将以下类型的系统配置为多宿主主机：

通常，可以将以下类型的系统配置为多宿主主机：

- 可以将 NFS 服务器（尤其是用作大型数据中心的那些服务器）连接到多个网络，以便在大量用户之间共享文件。这些服务器无需维护路由表。
- 数据库服务器可以具有多个网络接口，从而可为大量用户提供资源，就像 NFS 服务器那样。
- 防火墙网关是连接公司网络和公共网络（如 Internet）的系统。管理员将设置防火墙作为一项安全措施。当配置为防火墙时，主机不在连接到主机接口的网络之间传递包。但是，主机仍可以为授权用户提供标准 TCP/IP 服务，如 `ssh`。

注 – 当多宿主主机在其任一接口上具有不同类型的防火墙时，请小心谨慎以免无意中中断主机的包。对于有状态防火墙，尤其容易出现此问题。针对此问题的一种解决方案是配置无状态防火墙。有关防火墙的更多信息，请参阅《System Administration Guide: Security Services》中的“Firewall Systems”或第三方防火墙的相应文档。

▼ 如何创建多宿主主机

1 在预期的多宿主主机上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 配置并检测在 Oracle Solaris 安装过程中没有配置的其他每个网络接口。

请参阅第 129 页中的“如何在安装系统后配置物理接口”。

3 **验证并确认未在多宿主主机上启用 IP 转发。**

```
# routeadm
```

不带选项的 `routeadm` 命令可报告路由选择守护进程的状态。`routeadm` 的以下输出说明已启用 IPv4 转发：

```
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled
             IPv4 forwarding  enabled              disabled
             IPv6 forwarding  disabled             disabled

             Routing services   "route:default ripng:default"
```

4 **如果在系统上启用了包转发，请禁用它。**

使用以下命令之一：

- 对于 `routeadm` 命令，请键入以下内容：

  ```
  # routeadm -d ipv4-forwarding -u
  ```

- 要使用 SMF，请键入以下内容：

  ```
  # svcadm disable ipv4-forwarding
  ```

5 **（可选）为多宿主主机打开动态路由。**

使用以下命令之一打开 `in.routed` 守护进程：

- 对于 `routeadm` 命令，请键入以下内容：

  ```
  # routeadm -e ipv4-routing -u
  ```

- 要使用 SMF，请键入以下内容：

  ```
  # svcadm enable route:default
  ```

示例 5–6 配置多宿主主机

以下示例说明如何配置图 5–3 所示的多宿主主机。在该示例中，系统具有主机名 `hostc`。此主机具有两个接口，这两个接口都已连接到网络 `192.168.5.0`。

要开始操作，请显示系统接口的状态。

```
# dladm show-link
hme0            type: legacy    mtu: 1500       device: hme0
qfe0            type: legacy    mtu: 1500       device: qfe0
qfe1            type: legacy    mtu: 1500       device: qfe1
qfe2            type: legacy    mtu: 1500       device: qfe2
qfe3            type: legacy    mtu: 1500       device: qfe3
# ifconfig -a
```

```
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 192.168.5.82 netmask ff000000 broadcast 192.255.255.255
      ether 8:0:20:c1:1b:c6
```

dladm show-link 命令报告，hostc 具有两个接口，共有五个可能的链路。但是，仅检测到 hme0。要将 hostc 配置为多宿主主机，必须在 qfe NIC 上添加 qfe0 或其他链路。首先，请以物理方式将 qfe0 接口连接到 192.168.5.0 网络。然后，检测 qfe0 接口，并使其在重新引导后继续存在。

```
# ifconfig qf0 plumb
# ifconfig qfe0 192.168.5.85/8 up
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.5.82 netmask ff0000 broadcast 192.255.255.255
        ether 8:0:20:c1:1b:c6
qfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.5.85 netmask ff000000 broadcast 192.255.255.255
        ether 8:0:20:e1:3b:c4
 # vi /etc/hostname.qfe0
192.168.5.85
255.0.0.0
```

使用重新配置命令，重新引导系统：

```
# reboot -- -r
```

接下来，将 qfe0 接口添加到 hosts 数据库：

```
# vi /etc/inet/hosts
127.0.0.1           localhost
192.168.5.82        host3    #primary network interface for host3
192.168.5.85        host3-2  #second interface
```

然后，在 host3 上检查包转发和路由的状态：

```
# routeadm
              Configuration   Current             Current
                     Option   Configuration       System State
---------------------------------------------------------------
               IPv4 routing   enabled             enabled
               IPv6 routing   disabled            disabled
             IPv4 forwarding  enabled             enabled
             IPv6 forwarding  disabled            disabled

           Routing services   "route:default ripng:default"
```

routeadm 命令会报告，当前启用了通过 in.routed 守护进程的动态路由和包转发。但是，您将需要关闭包转发：

```
# svcadm disable ipv4-forwarding
```

也可以使用 `routeadm` 命令（如第 113 页中的 “如何创建多宿主主机”所示）关闭包转发。禁用包转发后，`host3` 将成为多宿主主机。

为单接口系统配置路由

单接口主机需要实现某种形式的路由。如果主机要从一个或多个本地缺省路由器获取其路由，则必须将该主机配置为使用静态路由。否则，建议对该主机使用动态路由。以下过程包含启用这两种路由类型的说明。

▼ 如何在单接口主机上启用静态路由

以下过程可在单接口主机上启用静态路由。使用静态路由的主机不运行动态路由协议（如 RIP）。相反，主机必须依赖于缺省路由器的服务来获取路由信息。第 103 页中的 “IPv4 自治系统拓扑”图显示了几个缺省路由器及其客户机主机。如果在安装特定主机时提供了缺省路由器的名称，则该主机已经配置为使用静态路由。

注 – 也可以使用以下过程在多宿主主机上配置静态路由。

有关 `/etc/defaultrouter` 文件的信息，请参见第 207 页中的 “`/etc/defaultrouter` 文件”。有关静态路由和路由表的信息，请参阅第 110 页中的 “路由表和路由类型”。

1. **在单接口主机上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **验证主机上是否存在 `/etc/defaultrouter` 文件。**

   ```
   # cd /etc
   # ls | grep defaultrouter
   ```

3. **打开文本编辑器以创建或修改 `/etc/defaultrouter` 文件。**

4. **添加缺省路由器的项。**

   ```
   # vi  /etc/defaultrouter
   router-IP
   ```

   其中 *router-IP* 指示供主机使用的缺省路由器的 IP 地址。

5 验证路由和包转发没有在主机上运行。

```
# routeadm
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled
             IPv4 forwarding   disabled            disabled
             IPv6 forwarding   disabled             disabled

           Routing services   "route:default ripng:default"
```

6 在本地 /etc/inet/hosts 文件中添加缺省路由器的项。

有关配置 /etc/inet/hosts 的信息，请参阅第 96 页中的 “如何更改 IPv4 地址和其他网络配置参数”。

示例 5–7 为单接口主机配置缺省路由器和静态路由

以下示例说明如何为图 5–3 所示的网络 172.20.1.0 中的单接口主机 hostb配置静态路由。hostb 需要使用路由器 2 作为其缺省路由器。

首先，请以超级用户身份登录到 hostb 或承担等效角色。然后，确定主机上是否存在 /etc/defaultrouter 文件：

```
# cd /etc
# ls | grep defaultrouter
```

如果没有来自 grep 的响应，则表示您需要创建 /etc/defaultrouter 文件。

```
# vi /etc/defaultrouter
172.20.1.10
```

/etc/defaultrouter 文件中的项是连接到 172.20.1.0 网络的路由器 2 上接口的 IP 地址。接下来，验证主机当前启用包转发还是启用路由。

```
# routeadm
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled
             IPv4 forwarding   enabled              enabled
             IPv6 forwarding   disabled             disabled

           Routing services   "route:default ripng:default"
```

已对此特定主机启用包转发。可按如下所示将其禁用：

```
# svcadm disable ipv4-forwarding
```

最后，确保主机的 /etc/inet/hosts 文件包含新缺省路由器的项。

```
# vi /etc/inet/hosts
127.0.0.1           localhost
172.20.1.18         host2    #primary network interface for host2
172.20.1.10         router2  #default router for host2
```

▼ 如何在单接口主机上启用动态路由

动态路由是管理主机上路由的最简单的方法。使用动态路由的主机运行由 IPv4 的 in.routed 守护进程或 IPv6 的 in.ripngd 守护进程提供的路由协议。使用接下来的过程在单接口主机上启用 IPv4 动态路由。有关动态路由的更多信息，请参阅第 100 页中的 “IPv4 网络上的包转发和路由”。

1 在主机上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 验证是否存在 /etc/defaultrouter 文件。

```
# cd /etc
# ls | grep defaultrouter
```

3 如果存在 /etc/defaultrouter，则删除在该文件中找到的所有项。

/etc/defaultrouter 文件为空会强制主机使用动态路由。

4 验证是否在主机上启用了包转发和路由。

```
# routeadm
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled
             IPv4 forwarding  enabled              enabled
             IPv6 forwarding  disabled             disabled

           Routing services   "route:default ripng:default"
```

5 如果启用了包转发，请将其关闭

使用以下命令之一：

- 对于 routeadm 命令，请键入以下内容：

  ```
  # routeadm -d ipv4-forwarding -u
  ```

- 要使用 SMF，请键入以下内容：

  ```
  # svcadm disable ipv4-forwarding
  ```

6 **在主机上启用路由协议。**

使用以下命令之一：

- 对于 `routeadm` 命令，请键入以下内容：

```
# routeadm -e ipv4-routing -u
```

- 要使用 SMF，请键入以下内容：

```
# svcadm enable route:default
```

现在已启用 IPv4 动态路由。主机的路由表是由 `in.routed` 守护进程动态维护的。

示例 5–8 在单接口主机上运行动态路由

以下示例说明如何为图 5–3 所示的网络 `192.168.5.0` 中的单接口主机 `hosta` 配置动态路由。`hosta` 当前使用路由器 1 作为其缺省路由器。但是，`hosta` 现在需要运行动态路由。

首先，请以超级用户身份登录到 `hosta` 或承担等效角色。然后，确定主机上是否存在 `/etc/defaultrouter` 文件：

```
# cd /etc
# ls | grep defaultrouter
defaultrouter
```

`grep` 的响应指示 `hosta` 存在相应的 `/etc/defaultrouter` 文件。

```
# vi /etc/defaultrouter
192.168.5.10
```

该文件具有项 `192.168.5.10`（这是路由器 1 的 IP 地址）。请删除此项以启用静态路由。接下来，需要验证是否已对该主机启用了包转发和路由。

```
# routeadm   Configuration   Current              Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled
            IPv4 forwarding   disabled             disabled
            IPv6 forwarding   disabled             disabled

          Routing services   "route:default ripng:default"
```

对于 `hosta`，路由和包转发均处于关闭状态。启用路由以完成 `hosta` 的动态路由配置，如下所示：

```
# svcadm enable route:default
```

监视和修改传输层服务

传输层协议 TCP、SCTP 和 UDP 是标准 Oracle Solaris软件包的一部分。这些协议通常无需进行干预即可正常运行。但是，站点上的具体情况可能要求您记录或修改通过传输层协议运行的服务。然后，您必须使用服务管理工具 (Service Management Facility, SMF) 来修改这些服务的配置文件，如《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”中所述。

`inetd` 守护进程负责在系统引导时启动标准 Internet 服务。这些服务包括将 TCP、SCTP 或 UDP 用作其传输层协议的应用程序。可以使用 SMF 命令修改现有的 Internet 服务或添加新服务。有关 `inetd` 的更多信息，请参阅第 214 页中的 “inetd Internet 服务守护进程”。

涉及传输层协议的操作包括：

- 记录所有的传入 TCP 连接
- 添加通过传输层协议（例如 SCTP）运行的服务
- 为访问控制配置 TCP 包装工具

有关 `inetd` 守护进程的详细信息，请参阅 `inetd(1M)` 手册页。

▼ 如何记录所有传入 TCP 连接的 IP 地址

1 **在本地系统上，承担网络管理角色或成为超级用户。**

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **对于 `inetd` 管理的所有服务，将 TCP 跟踪设置为“启用”。**

```
# inetadm -M tcp_trace=TRUE
```

▼ 如何添加使用 SCTP 协议的服务

SCTP 传输协议以与 TCP 类似的方式为应用层协议提供服务。但是，SCTP 允许单方或双方为多宿主系统的两个系统进行通信。SCTP 连接称为**关联**。在关联中，应用程序将要传输的数据分为一个或多个消息流，即**多流化**。SCTP 连接可以转到有多个 IP 地址的端点，这对电话应用程序尤其重要。如果站点使用 IP 过滤器或 IPsec，则 SCTP 的多宿主功能是出于安全考虑。`sctp(7P)` 手册页介绍了其中一些安全方面的考虑。

缺省情况下，SCTP 包括在 Oracle Solaris 中，且不需要其他配置。但是，可能需要显式配置某些应用层服务才能使用 SCTP。`echo` 和 `discard` 就是这样的应用程序。下一过程说明如何添加使用 SCTP 一对一样式套接字的回显服务。

注 – 也可以使用以下过程为 TCP 和 UDP 传输层协议添加服务。

以下任务说明如何将 `inetd` 守护进程管理的 SCTP `inet` 服务添加到 SMF 系统信息库。然后，该任务说明如何使用服务管理工具 (Service Management Facility, SMF) 命令添加该服务。

- 有关 SMF 命令的信息，请参阅《Oracle Solaris 管理：基本管理》中的“SMF 命令行管理实用程序”。
- 有关语法信息，请参阅该过程中所引用的 SMF 命令的手册页。
- 有关 SMF 的详细信息，请参阅 `smf(5)` 手册页。

开始之前 执行以下过程之前，请为服务创建清单文件。该过程以 `echo` 服务的清单 `echo.sctp.xml` 为例。

1. **使用拥有系统文件的写入特权的用户帐户，登录到本地系统。**

2. **编辑 `/etc/services` 文件并添加新服务的定义。**

   对于服务定义，使用以下语法。

   service-name |*port/protocol* | *aliases*

3. **添加新服务。**

   转到存储服务清单的目录，然后键入以下内容：

   ```
   # cd dir-name
   # svccfg import service-manifest-name
   ```

   有关 `svccfg` 的完整语法，请参阅 `svccfg(1M)` 手册页。

   假定您希望使用当前位于 `service.dir` 目录中的清单 `echo.sctp.xml` 添加新的 SCTP `echo` 服务，应键入以下内容：

   ```
   # cd service.dir
   # svccfg import echo.sctp.xml
   ```

4. **验证是否已添加服务清单：**

   ```
   # svcs FMRI
   ```

   对于 *FMRI* 参数，使用服务清单的故障管理资源标识符 (Fault Managed Resource Identifier, FMRI)。例如，对于 SCTP `echo` 服务，应使用以下命令：

   ```
   # svcs svc:/network/echo:sctp_stream
   ```

输出应该与如下所示类似：

```
    STATE          STIME    FMRI
disabled       16:17:00 svc:/network/echo:sctp_stream
```

有关 svcs 命令的详细信息，请参阅 svcs(1) 手册页。

该输出指明，新的服务清单当前处于禁用状态。

5 列出服务的属性以确定是否必须进行修改。

```
# inetadm -l FMRI
```

有关 inetadm 命令的详细信息，请参阅 inetadm(1M) 手册页。

例如，对于 SCTP echo 服务，应键入以下内容：

```
# inetadm -l svc:/network/echo:sctp_stream
SCOPE    NAME=VALUE
             name="echo"
             endpoint_type="stream"
             proto="sctp"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
         .
         .
         default  tcp_trace=FALSE
           default  tcp_wrappers=FALSE
```

6 启用新服务：

```
# inetadm -e FMRI
```

7 验证服务是否已启用：

例如，对于新的 echo 服务，应键入以下内容：

```
# inetadm | grep sctp_stream
.
.
    enabled   online         svc:/network/echo:sctp_stream
```

示例 5–9 添加使用 SCTP 传输协议的服务

以下示例给出要使用的命令以及使回显服务使用 SCTP 传输层协议所需的文件项。

```
$ cat /etc/services
.
.
echo            7/tcp
echo            7/udp
echo            7/sctp

# cd service.dir
```

```
    # svccfg import echo.sctp.xml

# svcs network/echo*
    STATE          STIME    FMRI
    disabled       15:46:44 svc:/network/echo:dgram
    disabled       15:46:44 svc:/network/echo:stream
    disabled       16:17:00 svc:/network/echo:sctp_stream

# inetadm -l svc:/network/echo:sctp_stream
    SCOPE    NAME=VALUE
             name="echo"
             endpoint_type="stream"
             proto="sctp"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
             user="root"
    default  bind_addr=""
    default  bind_fail_max=-1
    default  bind_fail_interval=-1
    default  max_con_rate=-1
    default  max_copies=-1
    default  con_rate_offline=-1
    default  failrate_cnt=40
    default  failrate_interval=60
    default  inherit_env=TRUE
    default  tcp_trace=FALSE
    default  tcp_wrappers=FALSE

# inetadm -e svc:/network/echo:sctp_stream

# inetadm | grep echo
    disabled  disabled       svc:/network/echo:stream
    disabled  disabled       svc:/network/echo:dgram
    enabled   online         svc:/network/echo:sctp_stream
```

▼ 如何使用 TCP 包装控制对 TCP 服务的访问

`tcpd` 程序可实现 *TCP* **包装**。TCP 包装介于守护进程和传入的服务请求之间，为诸如 `ftpd` 之类的服务守护进程提供了安全措施。TCP 包装记录成功的和不成功的连接尝试。此外，TCP 包装可以提供访问控制，根据发出请求的位置允许或拒绝连接。可以使用 TCP 包装保护诸如 SSH、Telnet 和 FTP 之类的守护进程。`sendmail` 应用程序也可以使用 TCP 包装，如《System Administration Guide: Network Services》中的“Support for TCP Wrappers From Version 8.12 of sendmail”所述。

1 **在本地系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 **将TCP包装设置为"启用"。**

```
# inetadm -M tcp_wrappers=TRUE
```

3 **配置TCP包装访问控制策略，如`hosts_access(3)`手册页中所述。**

此手册页可以在SFW CD-ROM（它与Oracle Solaris CD-ROM一起打包）上的`/usr/sfw/man`目录中找到。

◆ ◆ ◆ 第 6 章

管理网络接口（任务）

本章包含有关网络接口的任务和信息：

网络接口管理方面的新增功能

本章中的信息介绍从 Solaris 10 1/06 发行版开始的接口配置。有关 Oracle Solaris 新增功能的完整列表以及 Oracle Solaris 发行版的说明，请参阅《Oracle Solaris 10 1/13 新增功能》。

在 Solaris 10 1/06 中，引入了以下新功能：

- 在第 129 页中的“如何在安装系统后配置物理接口”中，介绍了用于查看接口状态的新增 `dladm` 命令。
- VLAN 支持已扩展到 GLDv3 接口，如第 134 页中的“管理虚拟局域网”所述。
- 在第 139 页中的“链路聚合概述”中介绍了链路聚合支持。

在 Solaris 10 7/07 中，`/etc/inet/ipnodes` 已过时。只能对早期 Solaris 10 发行版使用 `/etc/inet/ipnodes`，如以下各个过程中所述。

接口管理（任务列表）

下表列出了配置网络接口的各种任务，包括 VLAN 和链路聚合等特殊配置。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
检查系统上接口的状态。	列出系统上的所有接口，并检查已检测哪些接口。	第 128 页中的 “如何获取接口状态”
在安装系统后添加单个接口。	通过配置其他接口将系统更改为多宿主主机或路由器。	第 129 页中的 “如何在安装系统后配置物理接口”
SPARC：检查接口的 MAC 地址是否唯一。	确保接口是使用其出厂安装的 MAC 地址而不是系统 MAC 地址（仅限 SPARC）配置的。	第 133 页中的 “SPARC: 如何确保接口的 MAC 地址是唯一的”
规划虚拟局域网 (Virtual Local Area Network, VLAN)。	在创建 VLAN 之前执行必要的规划任务。	第 137 页中的 “如何规划 VLAN 配置”
配置 VLAN。	在网络上创建和修改 VLAN。	第 138 页中的 “如何配置 VLAN”
计划聚合。	在配置聚合前先设计聚合并执行必要的规划任务。	第 139 页中的 “链路聚合概述”
配置聚合。	执行各种与链路聚合相关的任务。	第 143 页中的 “如何创建链路聚合”
规划并配置 IPMP 组。	为属于 IPMP 组的接口配置故障转移和故障恢复。	第 625 页中的 “如何规划 IPMP 组” 第 626 页中的 “如何配置具有多个接口的 IPMP 组”

管理物理接口的基础知识

网络接口提供系统和网络之间的连接。基于 Oracle Solaris 的系统可以具有两种类型的接口：物理接口和逻辑接口。**物理接口**由软件驱动程序和连接网络介质（如以太网电缆）的连接器组成。可以对物理接口进行分组，以达到管理性或可用性目的。**逻辑接口**是在现有物理接口上配置的，通常用于在物理接口上添加地址和创建隧道端点。

注 – 逻辑网络接口在使用它们的任务（如 IPv6 任务、IPMP 任务、DHCP 任务和其他任务）中说明。

大多数计算机系统至少有一个由制造商在主系统板上**内置**的物理接口。一些系统也可能具有多个内置接口。

除了内置接口外，您可以向系统添加另行购买的接口。另行购买的接口称为**网络接口卡** (Network Interface Card, NIC)。应按照制造商的说明安装 NIC。

注 – NIC 也称为**网络适配器**。

在系统安装过程中，Oracle Solaris 安装程序检测物理安装的任何接口并显示每个接口的名称。至少必须配置接口列表中的一个接口。在安装过程中配置的第一个接口成为**主网络接口**。主网络接口的 IP 地址与系统的已配置主机名（存储在 `/etc/nodename` 文件中）关联。但是，可以在安装过程中或安装之后配置任何其他接口。

网络接口名称

每个物理接口都由唯一的设备名称标识。设备名称的语法如下：

```
<driver-name><instance-number>
```

Oracle Solaris 系统上的驱动程序名称可能包括 `ce`、`hme`、`bge`、`e1000g` 和许多其他驱动程序名称。变量 *instance-number* 可以具有从零到 *n* 的值，具体取决于在系统上安装了多少个该驱动程序类型的接口。

以 100BASE-TX 快速以太网接口为例，该接口通常用作主机系统和服务器系统上的主网络接口。此接口的一些典型驱动程序名称是 `eri`、`qfe` 和 `hme`。在用作主网络接口时，快速以太网接口具有诸如 `eri0` 或 `qfe0` 之类的设备名称。

诸如 `eri` 和 `hme` 之类的 NIC 只有一个接口。但是，许多品牌的 NIC 有多个接口。例如，Quad 快速以太网 (`qfe`) 卡具有四个接口：从 `qfe0` 到 `qfe3`。

检测接口

必须先对接口进行**检测**，之后它才能在系统和网络之间传递通信流量。检测过程涉及将接口与设备名称进行关联。然后，设置流以使 IP 协议可以使用该接口。物理接口和逻辑接口都必须进行检测。使用 `ifconfig` 命令的适当语法，作为引导序列的一部分或显式地检测接口。

在安装过程中配置接口时，将自动检测该接口。如果在安装过程中决定不在系统上配置其他接口，则不会检测这些接口。

Oracle Solaris 接口类型

从 Solaris 10 1/06 发行版开始，Oracle Solaris 支持以下两种类型的接口：

- **传统接口**－这些接口是 DLPI 接口和 GLDv2 接口。一些传统接口类型是 `eri`、`qfe` 和 `ce`。使用 `dladm show-link` 命令检查接口状态时，这些接口将被报告为“传统”。
- **非 VLAN 接口**－这些接口是 GLDv3 接口。

注 – 目前以下接口类型支持 GLDv3： bge、xge 和 e1000g。

管理单个网络接口

安装 Oracle Solaris 后，您可能会在系统上配置或管理接口以达到以下目的：

- 升级系统使其成为多宿主主机。有关更多信息，请参阅第 113 页中的 “配置多宿主主机”。
- 将主机更改为路由器。有关配置路由器的说明，请参阅第 105 页中的 “配置 IPv4 路由器”。
- 将接口配置为 VLAN 的一部分。有关更多信息，请参阅第 134 页中的 “管理虚拟局域网”。
- 将接口配置为聚合的成员。有关更多信息，请参阅第 139 页中的 “链路聚合概述”。
- 将接口添加到 IPMP 组。有关配置 IPMP 组的说明，请参阅第 624 页中的 “使用 IPMP 组获得高可用性”。

本节包含有关配置单个网络接口的信息（从 Solaris 10 1/06 发行版开始）。有关将接口配置到以下分组之一的信息，请参阅以下各节：

- 有关如何将接口配置到 VLAN 中，请参阅第 134 页中的 “管理虚拟局域网”。
- 有关如何将接口配置到聚合中，请参阅第 139 页中的 “链路聚合概述”。
- 有关如何将接口配置为 IPMP 组的成员，请参阅第 624 页中的 “使用 IPMP 组获得高可用性”。

▼ 如何获取接口状态

从 Solaris 10 1/06 开始，此过程说明如何确定系统上当前可用的接口及其状态。此过程还显示当前已检测哪些接口。如果使用的是早期的 Solaris 10 3/05，请参阅第 178 页中的 “如何获取有关特定接口的信息”。

1 在要配置接口的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 确定当前在系统上安装了哪些接口。

```
# dladm show-link
```

此步骤使用 dladm 命令（该命令在 dladm(1M) 手册页中详述）。此命令报告它找到的所有接口驱动程序，而不管接口当前是否已配置。

3 确定当前在系统上已检测哪些接口。

```
# ifconfig -a
```

ifconfig 命令具有许多附加功能，其中包括检测接口。有关更多信息，请参阅 ifconfig(1M) 手册页。

示例 6–1 使用 dladm 命令获取接口的状态

以下示例说明 dladm 命令显示的状态。

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
```

dladm show-link 的输出指示本地主机可以使用四个接口驱动程序。可以针对 VLAN 配置 ce 和 bge 接口。但是，只有**非** VLAN 类型的 GLDV3 接口才可以用于链路聚合。

以下示例说明 ifconfig -a 命令显示的状态。

```
# ifconfig -a
 lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
         inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>mtu 1500 index 3
         inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
        ether 0:3:ba:7:84:5e
bge0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4>mtu 1500 index 2
         inet 10.8.57.39 netmask ffffff00 broadcast 10.8.57.255
        ether 0:3:ba:29:fc:cc
```

ifconfig -a 命令的输出仅显示两个接口（ce0 和 bge0）的统计信息。此输出说明仅检测了 ce0 和 bge0，而且它们可以由网络通信流量使用。可以在 VLAN 中使用这些接口。由于已检测 bge0，无法在聚合中再使用此接口。

▼ 如何在安装系统后配置物理接口

开始之前

- 确定要用于附加接口的 IPv4 地址。
- 确保要配置的物理接口已安装在系统上。有关安装另行购买的 NIC 硬件的信息，请参阅 NIC 附带的制造商说明。
- 如果刚安装了接口，则在继续下一任务之前应执行重新配置引导。

1 在要配置接口的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 **确定当前在系统上安装了哪些接口。**

```
# dladm show-link
```

3 **配置并检测每个接口。**

```
# ifconfig interface plumb up
```

例如，对于 qfe0，请键入：

```
# ifconfig qfe0 plumb up
```

注 – 使用 ifconfig 命令显式配置的接口在重新引导后不会继续存在。

---

4 **将 IPv4 地址和网络掩码指定给接口。**

```
# ifconfig interface IPv4-address netmask+netmask
```

例如，对于 qfe0，请键入：

```
# ifconfig
qfe0 192.168.84.3 netmask + 255.255.255.0
```

注 – 可以使用传统的 IPv4 表示法或 CIDR 表示法指定 IPv4 地址。

---

5 **验证新配置的接口是否已检测并配置，或者是否带有 "UP" 标志。**

```
# ifconfig
-a
```

检查所显示的每个接口的状态行。确保状态行上的输出包含 UP 标志，例如：

```
qfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 2
```

6 **（可选）要使接口配置在重新引导后继续存在，请执行以下步骤：**

a. **为要配置的每个接口创建 /etc/hostname.*interface* 文件。**

例如，要添加 qfe0 接口，请创建以下文件：

```
# vi /etc/hostname.qfe0
```

注 – 如果为同一接口创建备用主机名文件，则备用文件也必须遵循命名格式 `hostname.[0-9]*`，如 `hostname.qfe0.a123`。诸如 `hostname.qfe0.bak` 或 `hostname.qfe0.old` 之类的名称无效，且会在系统引导期间被脚本忽略。

也请注意，一个给定的接口只能有一个相应的主机名文件。如果用有效的文件名为接口创建一个备用主机名文件，例如 `/etc/hostname.qfe` 和 `/etc/hostname.qfe.a123`，则引导脚本会尝试同时引用这两个主机名文件的内容来进行配置，因而会产生错误。要避免这些错误，请为给定配置中不使用的主机名文件使用无效的文件名。

---

b. 编辑 `/etc/hostname.`*interface* 文件。

至少将接口的 IPv4 地址添加到该文件。可以使用传统的 IPv4 表示法或 CIDR 表示法指定接口的 IP 地址。还可以将网络掩码和其他配置信息添加到该文件。

注 – 有关如何将 IPv6 地址添加到接口，请参阅第 158 页中的 “修改主机和服务器的 IPv6 接口配置”。

---

c. 对于 Solaris 10 11/06 和 Solaris 10 的早期发行版，将新接口的项添加到 `/etc/inet/ipnodes` 文件中。

d. 将新接口的项添加到 `/etc/inet/hosts` 文件中。

e. 执行重新配置引导。

```
# reboot -- -r
```

f. 验证是否已配置在 `/etc/hostname.` *interface* 文件中创建的接口。

```
# ifconfig -a
```

有关示例，请参阅示例 6–2。

示例 6–2 添加持久性接口配置

该示例说明如何将接口 `qfe0` 和 `qfe1` 配置为主机。这些接口在重新引导后仍继续存在。

```
# dladm show-link
eri0    type: legacy    mtu: 1500       device: eri0
qfe0    type: legacy    mtu: 1500       device: qfe0
qfe1    type: legacy    mtu: 1500       device: qfe1
qfe2    type: legacy    mtu: 1500       device: qfe2
qfe3    type: legacy    mtu: 1500       device: qfe3
bge0    type: non-vlan  mtu: 1500       device: bge0
# vi /etc/hostname.qfe0
192.168.84.3 netmask 255.255.255.0
# vi /etc/hostname.qfe1
192.168.84.72 netmask 255.255.255.0
```

```
# vi /etc/inet/hosts
# Internet host table
#
127.0.0.1       localhost
10.0.0.14       myhost
192.168.84.3       interface-2
192.168.84.72       interface-3
For Solaris 10 11/06 and earlier releases:# vi /etc/inet/ipnodes
10.0.0.14 myhost
192.168.84.3       interface-2
192.168.84.72       interface-3
```

此时，可以重新引导系统。

```
# reboot -- -r
```

引导系统后，验证接口配置。

```
ifconfig -a
# ifconfig -a lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu
8232 index 1
         inet 127.0.0.1 netmask ff000000
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
         inet 10.0.0.14netmask ff000000 broadcast 10.255.255.255
         ether 8:0:20:c1:8b:c3
qfe0:flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
      inet 192.168.84.3 netmask ffffff00 broadcast 192.255.255.255
      ether 8:0:20:c8:f4:1d
qfe1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>mtu 1500 index 4
         inet 192.168.84.72 netmask ffffff00 broadcast 10.255.255.255
        ether 8:0:20:c8:f4:1e
```

另请参见

- 有关如何在接口上配置 IPv6 地址，请参阅第 150 页中的 “如何启用当前会话的 IPv6 接口”。
- 有关如何使用 IP 网络多路径 (IP Network Multipathing, IPMP) 为接口设置故障转移检测和故障恢复，请参阅第 28 章，管理 IPMP（任务）。

▼ 如何删除物理接口

1 在要删除接口的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 删除物理接口。

```
# ifconfig interface down unplumb
```

例如，要删除接口 `qfe1`，可键入：

```
# ifconfig qfe1 down unplumb
```

▼ SPARC: 如何确保接口的 MAC 地址是唯一的

使用此过程配置 MAC 地址。

一些应用程序要求主机上的每个接口都具有唯一的 MAC 地址。但是，每个基于 SPARC 的系统都具有系统范围的 MAC 地址，缺省情况下所有接口都使用该地址。以下是在 SPARC 系统上可能希望为接口配置出厂安装的 MAC 地址的两种情况。

- 对于链路聚合，应该在聚合配置中使用接口的出厂设置 MAC 地址。
- 对于 IPMP 组，组中的每个接口都必须具有唯一的 MAC 地址。这些接口必须使用其出厂安装的 MAC 地址。

EEPROM 参数 `local-mac-address?` 确定 SPARC 系统上的所有接口使用系统范围的 MAC 地址还是其唯一 MAC 地址。以下过程说明如何使用 `eeprom` 命令检查 `local-mac-address?` 的当前值以及更改它的值（如有必要）。

1 在要配置接口的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 确定系统中的所有接口当前是否都使用系统范围的 MAC 地址。

```
# eeprom local-mac-address?
local-mac-address?=false
```

在此示例中，对 `eeprom` 命令的响应 `local-mac-address?=false` 表示所有接口确实使用了系统范围的 MAC 地址。只有 `local-mac-address?=false` 的值先更改为 `local-mac-address?=true` 后，接口才能成为 IPMP 组的成员。对于聚合，也应该将 `local-mac-address?=false` 更改为 `local-mac-address?=true`。

3 （如有必要）按如下所示更改 `local-mac-address?` 的值：

```
# eeprom local-mac-address?=true
```

重新引导系统时，具有出厂安装的 MAC 地址的接口现在使用这些出厂设置，而不是系统范围的 MAC 地址。没有出厂设置的 MAC 地址的接口继续使用系统范围的 MAC 地址。

4 检查系统中所有接口的 MAC 地址。

查找多个接口具有相同 MAC 地址的情况。在此示例中，所有接口都使用系统范围的 MAC 地址 `8:0:20:0:0:1`。

```
ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
      inet 127.0.0.1 netmask ff000000
hme0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 10.0.0.112 netmask ffffff80 broadcast 10.0.0.127
      ether 8:0:20:0:0:1
ce0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
```

```
      inet 10.0.0.114 netmask ffffff80 broadcast 10.0.0.127
      ether 8:0:20:0:0:1
ce1: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 10.0.0.118 netmask ffffff80 broadcast 10.0.0.127
      ether 8:0:20:0:0:1
```

注 – 仅当多个网络接口仍具有相同 MAC 地址时，才继续执行下一步。否则，转到最后一步。

---

5 **手动配置其余的接口，以便所有接口都具有唯一的 MAC 地址（如有必要）。**

在 `/etc/hostname.`*interface* 文件中为特定接口指定唯一的 MAC 地址。

对于上一步的示例，需要用本地管理的 MAC 地址配置 `ce0` 和 `ce1`。例如，要用本地管理的 MAC 地址 `06:05:04:03:02` 重新配置 `ce1`，可将以下行添加到 `/etc/hostname.ce1` 中：

```
ether 06:05:04:03:02
```

注 – 要防止手动配置的 MAC 地址与网络中的其他 MAC 地址冲突所带来的任何风险，必须始终配置**本地管理的** MAC 地址，如 IEEE 802.3 标准定义的那样。

也可以使用 `ifconfig ether` 命令为当前会话配置接口的 MAC 地址。但是，使用 `ifconfig` 直接进行的任何更改在重新引导后都不会保留。有关详细信息，请参阅 `ifconfig(1M)` 手册页。

6 **重新引导系统。**

管理虚拟局域网

虚拟局域网 *(Virtual Local Area Network, VLAN)* 是在 TCP/IP 协议栈的数据链路层上对局域网的细分。可以为采用交换机技术的局域网创建 VLAN。通过将用户组指定给 VLAN，可以加强整个本地网络的网络管理和安全性。还可以将同一系统上的接口指定给不同的 VLAN。

如果需要实现以下目的，请考虑将本地网络划分为多个 VLAN：

- 创建工作组的逻辑分区。

  例如，假定某楼层的所有主机都连接到一个基于交换的本地网络。可以为该楼层的每个工作组创建单独的 VLAN。

- 对各个工作组强制实施不同的安全策略。

  例如，财务部和信息技术部的安全需求大不相同。如果这两个部门的系统共享同一本地网络，则可以为每个部门创建单独的 VLAN。然后，基于每个 VLAN 强制实施适当的安全策略。

- 将工作组拆分为易管理的广播域。

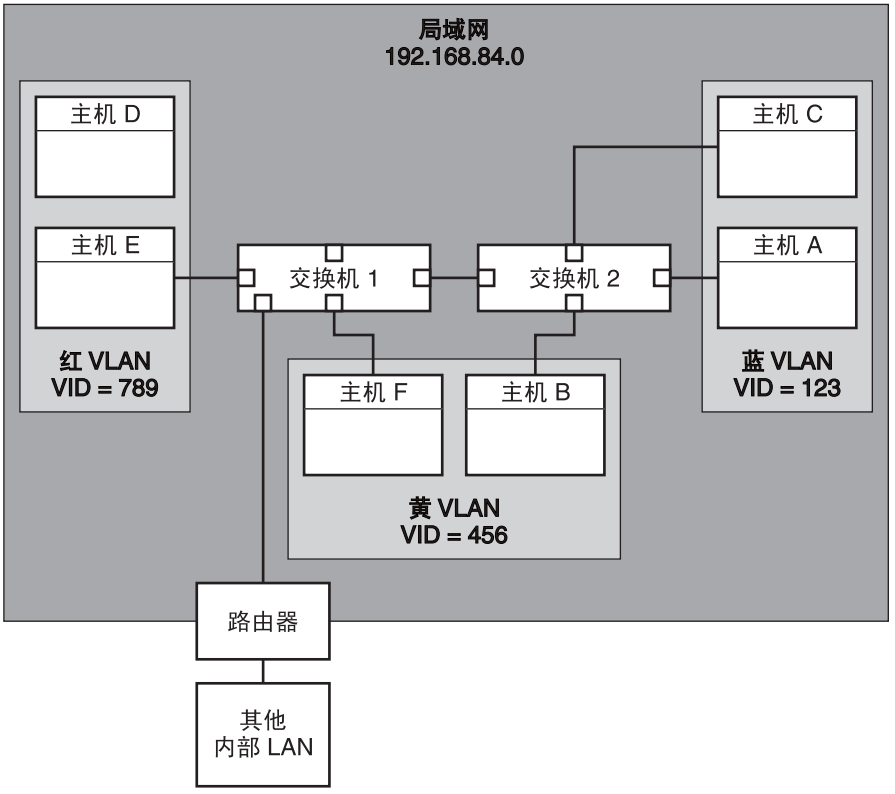
使用 VLAN 可减小广播域的大小并提高网络效率。

VLAN 拓扑概述

使用交换 LAN 技术，可以将本地网络中的系统组织到 VLAN 中。将本地网络划分为 VLAN 之前，必须先获取支持 VLAN 技术的交换机。可以对交换机上的所有端口进行配置，使其为单个 VLAN 或多个 VLAN 提供服务，具体取决于 VLAN 拓扑设计。配置交换机端口的过程因交换机制造商而异。

下图显示了子网地址为 192.168.84.0 的局域网。此 LAN 已细分为三个 VLAN：红 VLAN、黄 VLAN 和蓝 VLAN。

图 6–1 具有三个 VLAN 的局域网



LAN 192.168.84.0 上的连通性由交换机 1 和交换机 2 处理。红 VLAN 包含会计工作组中的系统。人力资源工作组的系统位于黄 VLAN 中。信息技术工作组的系统被指定给蓝 VLAN。

VLAN 标记和物理连接点

局域网中的每个 VLAN 都由 VLAN 标记或 *VLAN ID (VID)* 标识。VID 是在 VLAN 配置过程中指定的。VID 是一个介于 1 和 4094 之间的 12 位标识符，为每个 VLAN 提供唯一标识。在图 6–1 中，红 VLAN 的 VID 是 789，黄 VLAN 的 VID 是 456，蓝 VLAN 的 VID 是 123。

配置交换机使其支持 VLAN 时，需要为每个端口指定 VID。端口上的 VID 必须与指定给该端口所连接的接口的 VID 相同，如下图所示。

图 6–2 具有 VLAN 的网络的交换机配置

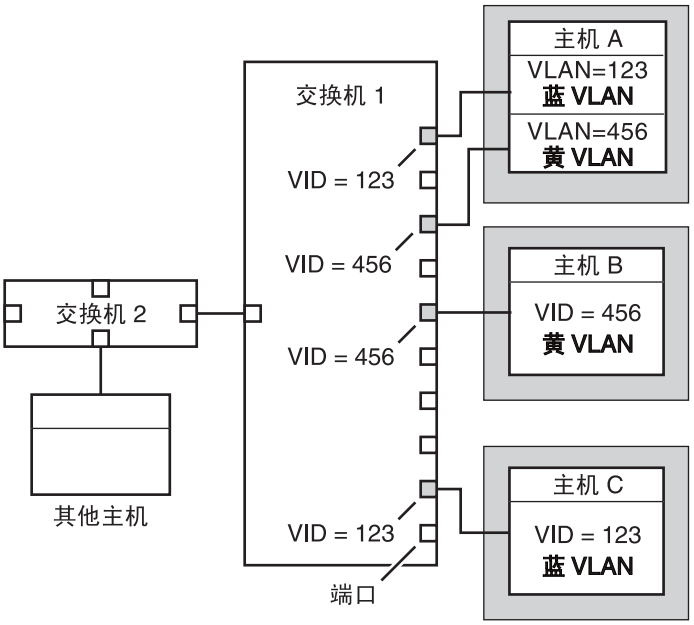


图 6–2 显示了连接到不同 VLAN 的多台主机。两台主机属于同一个 VLAN。在此图中，三台主机的主网络接口连接到交换机 1。主机 A 是蓝 VLAN 的成员。因此，主机 A 的接口配置为 VID 123。此接口连接到交换机 1 上的端口 1，该端口随后被配置为 VID 123。主机 B 是 VID 为 456 的黄 VLAN 的成员。主机 B 的接口连接到交换机 1 上的端口 5，该端口随后被配置为 VID 456。最后，主机 C 的接口连接到交换机 1 上的端口 9。蓝 VLAN 被配置为 VID 123。

该图还显示了一台主机也可以属于多个 VLAN。例如，主机 A 通过主机的接口配置了两个 VLAN。第二个 VLAN 被配置为 VID 456，并连接到端口 3，而端口 3 也被配置为 VID 456。因此，主机 A 同是蓝 VLAN 和黄 VLAN 的成员。

在 VLAN 配置过程中，必须指定 VLAN 的**物理连接点**（即 *PPA*）。使用以下公式可以获得 PPA 值：

driver-name + VID * 1000 + *device-instance*

请注意，*device-instance* 编号必须小于 1000。

例如，若要将 ce1 接口配置为 VLAN 456 的一部分，则创建以下 PPA：

```
ce + 456 * 1000 + 1= ce456001
```

规划网络中的 VLAN

使用以下过程可规划网络上的 VLAN。

▼ 如何规划 VLAN 配置

1 **检查本地网络拓扑，并确定在何处划分 VLAN 比较合适。**
有关此类拓扑的基本示例，请参阅图 6–1。

2 **创建 VID 的编号方案，并将 VID 指定给每个 VLAN。**

注 – VLAN 编号方案可能已存在于网络中。如果是这样，则必须在现有的 VLAN 编号方案中创建 VID。

---

3 **在每个系统上，确定哪些接口将是特定 VLAN 的成员。**

a. **确定系统上配置了哪些接口。**

```
# dladm show-link
```

b. **确定哪个 VID 将与系统上的相应数据链路相关联。**

c. **为要配置为 VLAN 一部分的每个接口创建 PPA。**

并非系统上的所有接口都一定要在同一 VLAN 上进行配置。

4 **检查接口与网络交换机的连接。**
记下每个接口的 VID 以及每个接口所连接到的交换机端口。

5 **用与所连接到的接口相同的 VID 配置交换机的每个端口。**
有关配置说明，请参阅交换机制造商的文档。

配置 VLAN

目前，Oracle Solaris在以下接口类型上支持 VLAN：

- ce
- bge
- xge
- e1000g

对于传统接口类型，只有 ce 接口可以成为 VLAN 的成员。可以在同一 VLAN 中配置不同类型的接口。

注 – 可以配置多个 VLAN 到 IPMP 组中。有关 IPMP 组的更多信息，请参阅第 615 页中的“IPMP 接口配置”。

▼ 如何配置 VLAN

1 承担主管理员角色，或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 确定正在系统上使用的接口的类型。

```
# dladm show-link
```

以下输出显示可用的接口类型：

```
ce0             type: legacy    mtu: 1500       device: ce0
 ce1             type: legacy    mtu: 1500       device: ce1
 bge0            type: non-vlan  mtu: 1500       device: bge0
 bge1            type: non-vlan  mtu: 1500       device: bge1
 bge2            type: non-vlan  mtu: 1500       device: bge2
```

3 将接口配置为 VLAN 的一部分。

```
# ifconfig interface-PPA plumb IP-address up
```

例如，可以使用以下命令，将新 IP 地址为 10.0.0.2 的接口 ce1 配置到 VID 为 123 的 VLAN 中：

```
# ifconfig ce123001 plumb 10.0.0.2
up
```

注 – 您可以将 IPv4 和 IPv6 地址指定给 VLAN，就像您对其他接口所执行的那样。

---

4 （可选）要使 VLAN 设置在重新引导后继续存在，请为配置为 VLAN 的一部分的每个接口创建 **hostname.***interface-PPA* 文件。

```
# cat hostname.interface-PPA
IPv4-address
```

5 在交换机上，设置 VLAN 标记和 VLAN 端口，使之与在系统上设置的 VLAN 相对应。

示例 6–3 配置 VLAN

此示例说明如何将设备 bge1 和 bge2 配置到 VID 为 123 的 VLAN 中。

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
# ifconfig bge123001 plumb 10.0.0.1 up
# ifconfig bge123002 plumb 10.0.0.2 up
# cat hostname.bge123001   10.0.0.1
# cat hostname.bge123002   10.0.0.2
# ifconfig -a
 lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
         inet 127.0.0.1 netmask ff000000
 bge123001: flags=201000803<UP,BROADCAST,MULTICAST,IPv4,CoS> mtu 1500 index 2
         inet 10.0.0.1 netmask ff000000 broadcast 10.255.255.255
         ether 0:3:ba:7:84:5e
bge123002:flags=201000803 <UP,BROADCAST,MULTICAST,IPv4,CoS> mtu 1500 index 3
         inet 10.0.0.2 netmask ff000000 broadcast 10.255.255.255
         ether 0:3:ba:7:84:5e
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>mtu 1500 index 4
         inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
         ether 0:3:ba:7:84:5e
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
bge123001       type: vlan 123  mtu: 1500       device: bge1
bge123002       type: vlan 123  mtu: 1500       device: bge2
```

链路聚合概述

注 – 最初的 Solaris 10 发行版和更早的 Solaris OS 版本不支持链路聚合。要为这些早期的 Solaris 发行版创建链路聚合，请使用 Sun Trunking，如《Sun Trunking 1.3 Installation and Users Guide》所述。

Oracle Solaris 支持将网络接口组织到链路聚合中。**链路聚合**由系统上配置在一起作为单个逻辑单元的若干接口组成。链路聚合也称为**中继**，它在 IEEE 802.3ad Link Aggregation Standard (`http://www.ieee802.org/3/index.html`)（IEEE 802.3ad 链路聚合标准）中进行了定义。

IEEE 802.3ad 链路聚合标准提供了一种方法，可将多个全双工以太网链路的容量组合到单个逻辑链路中。然后此链路聚合组被视为单个链路，实际上也是这样。

以下是链路聚合的功能：

- **增加了带宽**－将多个链路的容量组合到一个逻辑链路中。
- **自动故障转移/故障恢复**－将来自故障链路的通信转移到聚合中的工作链路。
- **负载平衡**－传入和外发通信都是根据用户选择的负载平衡策略（如源和目标 MAC 或 IP 地址）进行分配的。
- **支持冗余**－可以使用并行聚合来配置两个系统。
- **改进了管理**－所有接口作为一个单元进行管理。
- **减少了网络地址池消耗**－可以将一个 IP 地址指定给整个聚合。

链路聚合基础

基本链路聚合拓扑涉及由一组物理接口组成的单个聚合。在以下情况下，可能使用基本链路聚合：

- 对于运行具有分布式大通信流量的应用程序的系统，可以将聚合专用于该应用程序的通信流量。
- 对于具有有限的 IP 地址空间但仍需要很大带宽的站点，大的接口聚合仅需要一个 IP 地址。
- 对于需要隐藏内部接口的存在的站点，聚合的 IP 地址对外部应用程序隐藏其接口。

图 6–3 显示了承载常见 Web 站点的服务器的聚合。该站点需要增加带宽以满足 Internet 用户和站点数据库服务器之间的查询通信流量。出于安全目的，必须对外部应用程序隐藏服务器上各个接口的存在。解决方案是使用 IP 地址为 `192.168.50.32` 的聚合 `aggr1`。此聚合由三个接口（`bge0` 至 `bge2`）组成。这些接口专用于发出通信流量以响应用户查询。来自所有接口的包流量上的传出地址是`aggr1` 的 IP 地址，即 `192.168.50.32`。

图 6–3 基本链路聚合的拓扑

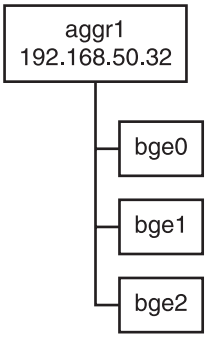
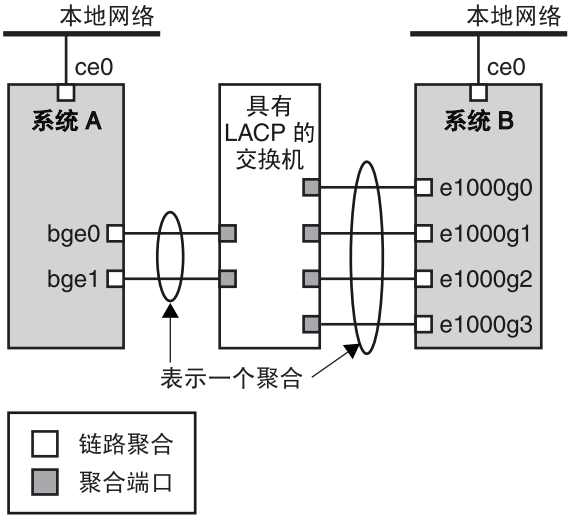


图 6–4 描述具有两个系统的本地网络，其中每个系统都配置了一个聚合。这两个系统由交换机连接在一起。如果需要通过交换机运行聚合，则该交换机必须支持聚合技术。对于高可用性和冗余系统，此类型的配置尤其有用。

在该图中，系统 A 具有的聚合由两个接口（即 `bge0` 和 `bge1`）组成。这些接口通过聚合端口连接到交换机。系统 B 具有的聚合由四个接口（即 `e1000g0` 至 `e1000g3`）组成。这些接口也连接到交换机上的聚合端口。

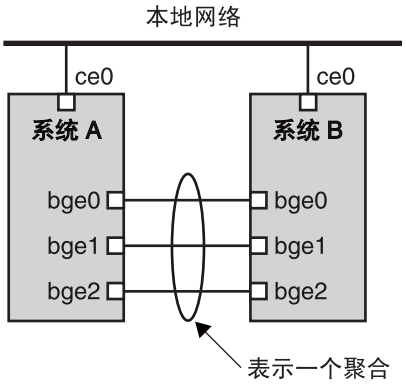
图 6–4 具有交换机的链路聚合拓扑



背对背链路聚合

背对背链路聚合拓扑涉及两个单独的系统，这两个系统通过电缆直接互连，如下图所示。这两个系统运行并行聚合。

图 6–5 基本的背对背聚合拓扑



在此图中，系统 A 上的设备 `bge0` 直接链接到系统 B 上的 `bge0`，依此类推。这样，系统 A 和 B 可以支持冗余和高可用性，以及这两个系统之间的高速通信。每个系统还将接口 `ce0` 配置为用于本地网络内的通信流。

背对背链路聚合最常见的应用是镜像数据库服务器。这两个服务器需要一起更新，因此需要很大的带宽、高速通信流和可靠性。最常使用背对背链路聚合的是数据中心。

策略和负载平衡

如果计划使用链路聚合，请考虑定义传出通信的策略。此策略可以指定希望如何在聚合的可用链路之间分配包，从而建立负载平衡。以下是可能用于聚合策略的层说明符及其意义：

- **L2** – 通过散列每个包的 MAC (L2) 头来确定传出链路
- **L3** – 通过散列每个包的 IP (L3) 头来确定传出链路
- **L4** – 通过散列每个包的 TCP、UDP 或其他 ULP (L4) 头来确定传出链路

这些策略的任意组合也是有效的。缺省策略是 L4。有关更多信息，请参阅 `dladm`(1M) 手册页。

聚合模式和交换机

如果聚合拓扑涉及通过交换机的连接，则必须注意此交换机是否支持**链路聚合控制协议** *(link aggregation control protocol, LACP)*。如果交换机支持 LACP，则必须为交换机和聚合配置 LACP。但是，可以定义运行 LACP 的以下**模式**之一：

- **关闭模式**－聚合的缺省模式。不生成 LACP 包（称为 *LACPDU*）。
- **主动模式**－系统按固定的时间间隔（您可以指定该时间间隔）生成 LACPDU。
- **被动模式**－系统仅在从交换机收到 LACPDU 时才生成 LACPDU。如果聚合和交换机均在被动模式下进行配置，则它们无法交换 LACPDU。

有关语法信息，请参见 `dladm`(1M) 手册页和交换机制造商文档。

链路聚合的要求

链路聚合配置必须符合以下要求：

- 必须使用 `dladm` 命令配置聚合。
- 已检测的接口不能成为聚合的成员。
- 接口必须为 GLDv3 类型：`xge`、`e1000g` 和 `bge`。
- 聚合中的所有接口必须以相同的速度和全双工模式运行。
- 必须将 EEPROM 参数 `local-mac-address?` 中的 MAC 地址值设置为 "true"。有关说明，请参阅第 133 页中的 “SPARC: 如何确保接口的 MAC 地址是唯一的”。

▼ 如何创建链路聚合

开始之前

注 – 链路聚合仅对以相同速度运行的全双工点对点链路起作用。确保聚合中的接口符合此要求。

如果要在聚合拓扑中使用交换机，请确保在该交换机上执行了以下操作：

- 配置了要用作聚合的端口
- 以主动模式或被动模式配置 LACP（如果交换机支持 LACP）

1 承担主管理员角色，或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 确定当前在系统上安装了哪些接口。

```
# dladm show-link
```

3 确定已检测哪些接口。

```
# ifconfig -a
```

4 创建聚合。

```
# dladm create-aggr -d interface -d interface [...]key
```

interface 表示要成为聚合一部分的接口的设备名称。

key 是标识聚合的编号。最小的密钥编号是 1。不允许将零用作密钥。

例如：

```
# dladm create-aggr -d bge0 -d bge1 1
```

5 配置并检测新创建的聚合。

```
# ifconfig aggrkey plumb IP-address up
```

例如：

```
# ifconfig aggr1  plumb 192.168.84.14 up
```

6 检查刚创建的聚合的状态。

```
# dladm show-aggr
```

将看到以下输出：

```
key: 1 (0x0001) policy: L4      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby
```

该输出显示已创建一个密钥为 1、策略为 L4 的聚合。

7 （可选）使链路聚合的 IP 配置在重新引导后继续存在。

a. 对于具有 IPv4 地址的链路聚合，创建一个 **/etc/hostname.aggr***key* 文件。对于基于 IPv6 的链路聚合，创建一个 **/etc/hostname6.aggr***key* 文件。

b. 将链路聚合的 IPv4 或 IPv6 地址输入到该文件中。

例如，您可以为在本过程中创建的聚合创建以下文件：

```
# vi /etc/hostname.aggr1
192.168.84.14
```

c. 执行重新配置引导。

```
# reboot -- -r
```

d. 验证您在 /etc/hostname.aggr*key* 文件中输入的链路聚合配置是否已配置。

```
# ifconfig -a
.
.
aggr1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.84.14 netmask ff000000 broadcast 192.255.255.
```

示例 6–4 创建链路聚合

此示例说明用于创建具有两个设备（即 bge0 和 bge1）的链路聚合的命令，以及命令的输出。

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
# ifconfig -a
lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
        ether 0:3:ba:7:84:5e
# dladm create-aggr -d bge0 -d bge1 1
# ifconfig aggr1 plumb 192.168.84.14 up
# dladm show-aggr
key: 1 (0x0001) policy: L4      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby

# ifconfig -a
lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
        ether 0:3:ba:7:84:5e
aggr1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.84.14 netmask ff000000 broadcast 192.255.255.255
        ether 0:3:ba:7:84:5e
```

请注意，用于聚合的两个接口以前没有经过 ifconfig 检测。

▼ 如何修改聚合

此过程说明如何对聚合定义进行以下更改：

- 修改聚合的策略
- 更改聚合的模式

1. **承担主管理员角色，或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **修改聚合以更改策略。**

   ```
   # dladm modify-aggr -Ppolicy key
   ```

   policy 表示策略 L2、L3 和 L4 中的一个或多个，如第 142 页中的 “策略和负载平衡”中所述。

   key 是标识聚合的编号。最小的密钥编号是 1。不允许将零用作密钥。

3. **如果 LACP 正运行在聚合中的设备所连接到的交换机上，则修改该聚合使其支持 LACP。**

   如果交换机在被动模式下运行 LACP，请务必为聚合配置主动模式。

   ```
   # dladm modify-aggr -l LACP mode -t timer-value key
   ```

   `-l` *LACP mode* 指示运行聚合的 LACP 模式。值包括 `active`、`passive` 和 `off`。

   `-t` *timer-value* 指示 LACP 计时器值（`short` 或 `long`）。

   key 是标识聚合的编号。最小的密钥编号是 1。不允许将零用作密钥。

示例 6–5 **修改链路聚合**

此示例说明如何将聚合 `aggr1` 的策略修改为 L2，并在随后打开活动 LACP 模式。

```
# dladm modify-aggr -P L2 1
# dladm modify-aggr -l active -t short 1
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby
```

▼ 如何删除聚合中的接口

1. **承担主管理员角色，或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **删除聚合中的接口。**

   ```
   # dladm remove-aggr -d interface
   ```

示例 6–6 删除聚合中的接口

此示例说明如何删除聚合 aggr1 的接口。

```
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby
# dladm remove-aggr -d bge1 1
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
```

▼ 如何删除聚合

1 **承担主管理员角色，或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **删除聚合。**

```
# dladm delete-aggr key
```

key 是标识聚合的编号。最小的密钥编号是 1。不允许将零用作密钥。

示例 6–7 如何删除聚合

此示例说明如何删除聚合 aggr1。

```
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
     device   address           speed     duplex  link    state
# dladm delete-aggr -d 1
```

▼ 如何通过链路聚合配置 VLAN

与通过接口配置 VLAN 的方式一样，也可以基于链路聚合创建 VLAN。第 134 页中的 “管理虚拟局域网”中对 VLAN 进行了介绍。本节综合介绍了如何配置 VLAN 和链路聚合。

开始之前 创建链路聚合。请记下聚合的 key 的值，在通过聚合创建 VLAN 时将需要该值。要创建链路聚合，请参阅第 143 页中的 “如何创建链路聚合”。

1. **如果先前已经创建链路聚合，请获取该聚合的密钥。**

   ```
   # dladm show-aggr
   ```

2. **通过链路聚合创建 VLAN。**

   ```
   # ifconfig aggrVIDkey plumb
   ```

   其中

   VID VLAN 的 ID

   key 要基于其创建 VLAN 的链路聚合的密钥。密钥必须采用 3 位格式。例如，如果聚合的密钥是 1，那么 VLAN 名称中所包含的密钥号为 001。

3. **重复步骤 2，通过聚合创建其他 VLAN。**

4. **使用有效的 IP 地址配置 VLAN。**

5. **要创建持久的 VLAN 配置，请将 IP 地址信息添加到相应的 /etc/hostname.*VLAN* 配置文件。**

示例 6–8 通过链路聚合配置多个 VLAN

在此示例中，基于链路聚合配置了两个 VLAN。dladm show-aggr 命令的输出指示链路聚合的密钥是 1。为 VLAN 指定的 VID 分别为 193 和 194。

```
# dladm show-aggr
key: 1 (0x0001) policy: L4      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby

# ifconfig aggr193001 plumb
# ifconfig aggr193001 192.168.10.0/24 up

# ifconfig aggr194001 plumb
# ifconfig aggr194001 192.168.20.0/24 up

# vi /etc/hostname.aggr193001
192.168.10.0/24

# vi /etc/hostname.aggr194001
192.168.20.0/24
```

◆ ◆ ◆ 第 7 章

7 配置 IPv6 网络（任务）

本章包含在网络上配置 IPv6 需要执行的任务。本章主要包含以下主题：

有关 IPv6 的各种类型的信息，请参阅以下源：

- 有关 IPv6 概念的概述：第 3 章，IPv6 介绍（概述）。
- 有关 IPv6 规划任务：第 4 章，规划 IPv6 网络（任务）
- 有关为使用 IP 隧道做准备：第 81 页中的 “在网络拓扑中规划隧道”
- 有关参考信息：Chapter 11, IPv6 详解（参考）

配置 IPv6 接口

作为在网络上使用 IPv6 的初始步骤，请在系统的 IP 接口上配置 IPv6。

在 Oracle Solaris 安装过程中，可以在一个或多个系统接口上启用 IPv6。如果在安装期间启用 IPv6 支持，则在安装完成后，将存在以下 IPv6 相关的文件和表：

- 针对 IPv6 启用的每个接口现在都有一个与之关联的 `/etc/hostname6.`*interface* 文件，如 `hostname6.dmfe0`。
- 对于 Solaris 10 11/06 及更早的版本，创建了 `/etc/inet/ipnodes` 文件。安装之后，此文件通常仅包含 IPv6 和 IPv4 回送地址。
- 修改了 `/etc/nsswitch.conf` 文件，以便使用 IPv6 地址进行查找。
- `name-service/switch` SMF 服务已修改为包含使用 IPv6 地址的查找。

- 创建了 IPv6 地址选择策略表。该表确定通过启用了 IPv6 的接口进行传输时所用 IP 地址格式的优先级。

本节介绍如何在安装 Oracle Solaris 完成后在接口上启用 IPv6。

在接口上启用 IPv6（任务列表）

下表列出了各种配置 IPv6 接口的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
在已经装有 Oracle Solaris 的系统的接口上启用 IPv6。	使用此任务可以在安装 Oracle Solaris 之后在接口上启用 IPv6。	第 150 页中的 “如何启用当前会话的 IPv6 接口”
使启用了 IPv6 的接口的配置在重新引导后仍然持保持不变。	使用此任务可以使接口的 IPv6 地址成为永久地址。	第 152 页中的 “如何启用持久性 IPv6 接口”
关闭 IPv6 地址自动配置。	如果需要手动配置 IPv6 地址的接口 ID 部分，请使用此任务。	第 153 页中的 “如何关闭 IPv6 地址自动配置”

▼ 如何启用当前会话的 IPv6 接口

开始配置 IPv6 时，请首先在将成为 IPv6 节点的所有系统的接口上启用 IPv6。最初，接口通过自动配置过程获取其 IPv6 地址，如第 73 页中的 “IPv6 地址自动配置”中所述。然后，可以根据节点在 IPv6 网络中的作用（作为主机、服务器或路由器）来调整节点的配置。

注 – 如果接口与当前正在通告某个 IPv6 前缀的路由器在同一链路上，则接口会获取该站点前缀，并将其作为自动配置的地址的一部分。有关更多信息，请参阅第 155 页中的 “如何配置启用了 IPv6 的路由器”。

以下过程说明如何为安装 Oracle Solaris 之后添加的接口启用 IPv6。

开始之前 完成 IPv6 网络的规划任务，如升级硬件和软件以及准备寻址计划。有关更多信息，请参见第 75 页中的 “IPv6 规划（任务列表）”。

1 以主管理员身份或超级用户身份登录预期的 IPv6 节点。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **在接口上启用 IPv6。**

```
# ifconfig interface inet6 plumb up
```

3 **启动 IPv6 守护进程 `in.ndpd`。**

```
# /usr/lib/inet/in.ndpd
```

注 – 可以使用 `ifconfig -a6` 命令显示节点上启用了 IPv6 的接口的状态。

示例 7–1 在安装之后启用 IPv6 接口

此示例说明如何在 `qfe0` 接口上启用 IPv6。在开始之前，请检查系统上已配置的所有接口的状态。

```
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1000863 <UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500
           index 2
         inet 172.16.27.74 netmask ffffff00 broadcast 172.16.27.255
         ether 0:3:ba:13:14:e1
```

目前仅为该系统配置了 `qfe0` 接口。请按如下所示在该接口上启用 IPv6：

```
# ifconfig qfe0 inet6 plumb up
# /usr/lib/inet/in.ndpd
# ifconfig -a6
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 0:3:ba:13:14:e1
        inet6 fe80::203:baff:fe13:14e1/10
```

此示例显示了系统接口在 `qfe0` 启用 IPv6 前后的状态。`ifconfig` 的 `-a6` 选项仅显示 `qfe0` 的 IPv6 信息和回送接口。请注意，输出结果表明仅为 `qfe0` 配置了一个链路本地地址 `fe80::203:baff:fe13:14e1/10`。此地址指明到目前为止，该节点的本地链路上没有任何路由器通告站点前缀。

启用 IPv6 之后，可以使用 `ifconfig -a` 命令显示系统上所有接口的 IPv4 和 IPv6 地址。

接下来的步骤

- 要将 IPv6 节点配置为路由器，请转至第 154 页中的 “配置 IPv6 路由器”。
- 要使 IPv6 接口配置在重新引导后仍然保持不变，请参见第 152 页中的 “如何启用持久性 IPv6 接口”。
- 要在节点上禁用地址自动配置，请参见第 153 页中的 “如何关闭 IPv6 地址自动配置”。
- 要将节点调整为服务器，请参见第 163 页中的 “在服务器上管理启用了 IPv6 的接口”中的建议。

▼ 如何启用持久性 IPv6 接口

此过程说明如何启用具有自动配置的 IPv6 地址的 IPv6 接口，这些地址在随后进行重新引导时将一直保持不变。

注 – 如果接口与当前正在通告某个 IPv6 前缀的路由器在同一链路上，则接口会获取该站点前缀，并将其作为自动配置的地址的一部分。有关更多信息，请参阅第 155 页中的“如何配置启用了 IPv6 的路由器”。

---

1 以主管理员身份或超级用户身份登录 IPv6 节点。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 为安装之后添加的接口创建 IPv6 地址。

a. 创建配置文件。

```
# touch /etc/hostname6.interface
```

b. 向配置文件添加地址。

```
ipv6-address up
...
```

3 创建静态 IPv6 缺省路由。

```
# /usr/sbin/route -p add -inet6 default ipv6-address
```

4 （可选）创建一个 /etc/inet/ndpd.conf 文件，该文件定义了节点上接口变量的参数。

如果需要为主机的接口创建临时地址，请参阅第 158 页中的“将临时地址用于接口”。有关 /etc/inet/ndpd.conf 的详细信息，请参阅 ndpd.conf(4) 手册页和第 231 页中的“ndpd.conf 配置文件”。

5 重新引导该节点。

```
# reboot -- -r
```

在重新引导过程中将发送路由器搜索包。如果路由器以站点前缀进行响应，则节点可以使用对应的包含全局 IPv6 地址的 /etc/hostname6.*interface* 文件配置任何接口。否则，仅能使用链路本地地址配置启用了 IPv6 的接口。重新引导还将以 IPv6 模式重新启动 in.ndpd 和其他网络守护进程。

示例 7–2 使 IPv6 接口的配置在重新引导过程中持续保留

此示例说明如何使 qfe0 接口的 IPv6 配置在重新引导过程中持续保留。在此示例中，本地链路上的路由器会通告站点前缀和子网 ID 2001:db8:3c4d:15/64。

首先，检查系统接口的状态。

```
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1000863 <UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500
           index 2
        inet 172.16.27.74 netmask ffffff00 broadcast 172.16.27.255
        ether 0:3:ba:13:14:e1

# touch /etc/hostname6.qfe0
# vi /etc/hostname6.qfe0
inet6 fe80::203:baff:fe13:1431/10 up
addif 2001:db8:3c4d:15:203:baff:fe13:14e1/64 up

# route -p add -inet6 default fe80::203:baff:fe13:1431
# reboot -- -r
```

验证已配置的 IPv6 地址是否仍适用于 `qfe0` 接口。

```
# ifconfig -a6
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
       ether 0:3:ba:13:14:e1
       inet6 fe80::203:baff:fe13:14e1/10
 qfe0:1: flags=2180841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500
          index 2
        inet6 2001:db8:3c4d:15:203:baff:fe13:14e1/64
```

`ifconfig -a6` 的输出显示了 `qfe0` 的两项信息。标准 `qfe0` 项包括 MAC 地址和链路本地地址。第二项 `qfe0:1` 表示为 `qfe0` 接口上的其他 IPv6 地址创建了伪接口。新的全局 IPv6 地址 `2001:db8:3c4d:15:203:baff:fe13:14e1/64` 包括由本地路由器通告的站点前缀和子网 ID。

接下来的步骤

- 要将新的 IPv6 节点配置为路由器，请转至第 154 页中的 “配置 IPv6 路由器”。
- 要在节点上禁用地址自动配置，请参见第 153 页中的 “如何关闭 IPv6 地址自动配置”。
- 要将新节点调整为服务器，请参见第 163 页中的 “在服务器上管理启用了 IPv6 的接口”中的建议。

▼ 如何关闭 IPv6 地址自动配置

通常应当使用地址自动配置来为主机和服务器的接口生成 IPv6 地址。但是，有时可能希望关闭地址自动配置，尤其是在希望手动配置标记时，如第 161 页中的 “配置 IPv6 标记”中所述。

1 以主管理员身份或超级用户身份登录 IPv6 节点。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **为节点创建 /etc/inet/ndpd.conf 文件。**

`/etc/inet/ndpd.conf` 文件定义了特定节点的接口变量。必须在该文件中包含以下内容，才能关闭所有服务器接口的地址自动配置：

if-variable-name `StatelessAddrConf false`

有关 `/etc/inet/ndpd.conf` 的详细信息，请参阅 `ndpd.conf(4)` 手册页和第 231 页中的 “ndpd.conf 配置文件”。

3 **使用所做更改更新 IPv6 守护进程。**

```
# pkill -HUP in.ndpd
```

配置 IPv6 路由器

在网络上配置 IPv6 的第一步是在路由器上配置 IPv6。路由器配置涉及到许多独立的任务，本节将介绍这些任务。您可以根据站点要求执行部分或全部任务。

IPv6 路由器配置（任务列表）

按下表中的顺序执行接下来的任务，以配置 IPv6 网络。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
1. 在开始配置 IPv6 之前，确保已完成了必需的先决条件。	必须先在启用了 IPv6 的接口上完成规划任务和 Oracle Solaris 安装，然后再配置启用了 IPv6 的路由器。	第 4 章，规划 IPv6 网络（任务）和第 149 页中的 “配置 IPv6 接口”。
2. 配置路由器。	定义网络的站点前缀。	第 155 页中的 “如何配置启用了 IPv6 的路由器”
3. 在路由器上配置隧道接口。	在路由器上设置手动隧道或 6to4 隧道接口。本地 IPv6 网络需要使用隧道来与其他隔离的 IPv6 网络通信。	■ 第 167 页中的 “如何配置 6to4 隧道” ■ 第 165 页中的 “如何手动配置 IPv6 over IPv4 隧道” ■ 第 166 页中的 “如何手动配置 IPv6 over IPv6 隧道” ■ 第 167 页中的 “如何配置 IPv4 over IPv6 隧道”
4. 在网络上配置交换机。	如果网络配置中包括交换机，此时请在配置过程中针对 IPv6 配置交换机。	请参阅交换机制造商文档。
5. 在网络上配置任何集线器。	如果网络配置中包括集线器，此时请在配置过程中针对 IPv6 配置集线器。	请参阅集线器制造商文档。

任务	说明	参考
6. 针对 IPv6 配置网络名称服务。	针对 IPv6 配置路由器之后，将主名称服务（DNS、NIS 或 LDAP）配置为识别 IPv6 地址。	第 172 页中的 “如何向 DNS 中添加 IPv6 地址”
7.（可选）在主机和服务器上修改启用了 IPv6 的接口的地址。	配置 IPv6 路由器之后，对启用了 IPv6 的主机和服务器进一步进行修改。	第 158 页中的 “修改主机和服务器的 IPv6 接口配置”
8. 将应用程序配置为支持 IPv6。	为了支持 IPv6，不同的应用程序可能需要不同的操作。	请参阅应用程序文档。

▼ 如何配置启用了 IPv6 的路由器

此过程假定在 Oracle Solaris 安装期间针对 IPv6 配置了路由器上的所有接口。

1 **在即将成为 IPv6 路由器的系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **检查在安装过程中针对 IPv6 配置了路由器上的哪些接口。**

```
# ifconfig -a
```

检查输出，确保现在已经使用链路本地地址检测了要针对 IPv6 配置的接口。以下 `ifconfig -a` 的样例命令输出显示了已经为路由器的接口配置的 IPv4 和 IPv6 地址。

```
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.16.26.232 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:15
dmfe1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4 mtu 1500 index 3
        inet 172.16.26.220 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:16
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
dmfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 0:3:ba:11:b1:15
        inet6 fe80::203:baff:fe11:b115/10
dmfe1: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 3
        ether 0:3:ba:11:b1:16
        inet6 fe80::203:baff:fe11:b116/10
```

该输出还显示在安装过程中已经使用 IPv6 链路本地地址 `fe80::203:baff:fe11:b115/10` 和 `fe80::203:baff:fe11:b116/10` 配置了主网络接口 `dmfe0` 和附加接口 `dmfe1`。

3 **在路由器的所有接口上配置 IPv6 包转发。**

对于 Solaris 10 11/03 及更早的版本，请使用以下命令：

```
# routeadm -e ipv6-forwarding -u
```

使用以下任一命令启用包转发：

- 使用 `routeadm` 命令，如下所示：

  ```
  # routeadm -e ipv6-forwarding -u
  ```

- 使用以下服务管理工具 (Service Management Facility, SMF) 命令，如下所示：

  ```
  # svcadm enable ipv6-forwarding
  ```

4 **启动路由选择守护进程。**

`in.ripngd` 守护进程可处理 IPv6 路由。通过以下任一方式启用 IPv6 路由：

对于 Solaris 10 11/06 及更早的版本，请键入以下命令启动 `in.ripngd`：

```
# routeadm -e ipv6-routing
# routeadm -u
```

- 使用 `routeadm` 命令：

  ```
  # routeadm -e ipv6-routing -u
  ```

- 使用合适的 SMF 命令：

  ```
  # svcadm enable ripng:default
  ```

有关 `routeadm` 命令的语法信息，请参见 `routeadm(1M)` 手册页。

5 **创建 `/etc/inet/ndpd.conf` 文件。**

在 `/etc/inet/ndpd.conf` 中指定要由路由器通告的站点前缀以及其他配置信息。此文件由 `in.ndpd` 守护进程读取，该守护进程实现了 IPv6 相邻节点搜索协议。

有关变量和允许值的列表，请参阅第 231 页中的 “ndpd.conf 配置文件”和 `ndpd.conf(4)` 手册页。

6 **在 `/etc/inet/ndpd.conf` 文件中键入以下文本：**

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on
```

此文本通告 `in.ndpd` 守护进程通过路由器上针对 IPv6 配置的所有接口发出路由器通告。

7 **向 `/etc/inet/ndpd.conf` 文件中添加其他文本，以便在路由器的各接口上配置站点前缀。**

该文本应采用以下格式：

`prefix` *global-routing-prefix*:*subnet ID*`/64` *interface*

以下样例 /etc/inet/ndpd.conf 文件将路由器配置为通过接口 dmfe0 和 dmfe1 通告站点前缀 2001:0db8:3c4d::/48。

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on

if dmfe0 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:15::0/64 dmfe0

if dmfe1 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:16::0/64 dmfe1
```

8 **重新引导系统。**

IPv6 路由器随即开始在本地链路上通告 ndpd.conf 文件中的任何站点前缀。

示例 7–3 显示 IPv6 接口的 ifconfig 输出

以下示例显示了 ifconfig -a 命令的输出，在完成第 154 页中的 “配置 IPv6 路由器”过程之后将看到这类输出。

```
lo0: flags=1000849 <UP LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.16.15.232 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:15
dmfe1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4 mtu 1500 index 3
        inet 172.16.16.220 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:16
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
dmfe0: flags=2100841 <UP,RUNNING,MULTICAST,ROUTER,IPv6> mtu 1500 index 2
        ether 0:3:ba:11:b1:15
        inet6 fe80::203:baff:fe11:b115/10
dmfe0:1: flags=2180841 <UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6> mtu 1500
          index 2
        inet6 2001:db8:3c4d:15:203:baff:fe11:b115/64
dmfe1: flags=2100841 <UP,RUNNING,MULTICAST,ROUTER,IPv6> mtu 1500 index 3
        ether 0:3:ba:11:b1:16
        inet6 fe80::203:baff:fe11:b116/10
dmfe1:1: flags=2180841 <UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6> mtu 1500
           index 3
        inet6 2001:db8:3c4d:16:203:baff:fe11:b116/64
```

在此示例中，已经针对 IPv6 配置的每个接口现在都有两个地址。包含接口名称的项（如 dmfe0）显示该接口的链路本地地址。形式为 *interface:n* 的项（如 dmfe0:1）显示全局 IPv6 地址。此地址除包括接口 ID 外，还包括已在 /etc/ndpd.conf 文件中配置的站点前缀。

另请参见

- 有关如何配置 IPv6 网络拓扑中标识的路由器的任何隧道的信息，请参阅第 165 页中的 “针对 IPv6 支持配置隧道”。
- 有关在网络上配置交换机和集线器的信息，请参阅制造商文档。

- 要配置 IPv6 主机，请参阅第 158 页中的 “修改主机和服务器的 IPv6 接口配置”。
- 要改进服务器对 IPv6 的支持，请参阅第 163 页中的 “在服务器上管理启用了 IPv6 的接口”。
- 有关 IPv6 命令、文件和守护进程的详细信息，请参阅第 231 页中的 “Oracle Solaris IPv6 实现”。

修改主机和服务器的 IPv6 接口配置

本节介绍如何修改作为主机或服务器的节点上启用了 IPv6 的接口的配置。大多数情况下，应当针对启用了 IPv6 的接口使用地址自动配置，如第 73 页中的 “无状态自动配置概述”中所述。但是，可以按照本节中的任务说明，根据需要修改接口的 IPv6 地址。

修改 IPv6 接口配置（任务列表）

下表列出了各种可修改现有 IPv6 网络的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
关闭 IPv6 地址自动配置。	如果需要手动配置 IPv6 地址的接口 ID 部分，请使用此任务。	第 153 页中的 “如何关闭 IPv6 地址自动配置”
为主机创建临时地址。	通过对随机创建的临时地址（用作地址中较低的 64 位）进行配置，隐藏主机的接口 ID。	第 159 页中的 “如何配置临时地址”
为系统的接口 ID 配置标记。	在 IPv6 地址中创建要用作接口 ID 的 64 位标记。	第 161 页中的 “如何配置用户指定的 IPv6 标记”

将临时地址用于接口

IPv6 **临时地址**包括一个随机生成的用作接口 ID 的 64 位数字，而不是包括接口的 MAC 地址。对于要保持匿名的 IPv6 节点上的任何接口都可以使用临时地址。例如，您可能希望对于需要访问公共 Web 服务器的主机的接口使用临时地址。临时地址可实现 IPv6 保密性增强功能。RFC 3041 中介绍了这些增强功能，可从《Privacy Extensions for Stateless Address Autoconfiguration in IPv6》（《IPv6 中用于无状态地址自动配置的专用扩展》）(`http://www.ietf.org/rfc/rfc3041.txt?number=3041`)中获取。

如果需要的话，可以在 `/etc/inet/ndpd.conf` 文件中为一个或多个接口启用临时地址。但是，与自动配置的标准 IPv6 地址不同，临时地址由 64 位子网前缀和一个随机生成的 64 位数字组成。这个随机数将成为 IPv6 地址的接口 ID 部分。临时地址作为接口 ID 时，不会生成链路本地地址。

请注意，临时地址的缺省**首选生命周期**为一天。启用临时地址生成功能时，还可以在 `/etc/inet/ndpd.conf` 文件中配置下列变量：

有效生命周期 TmpValidLifetime	临时地址存在的时间跨度，在此之后临时地址将从主机中删除。
首选生命周期 TmpPreferredLifetime	临时地址过时之前已经过的时间。此时间跨度应短于有效生命周期。
地址重新生成时间	在首选生命周期到期之前的持续时间，在这段时间内，主机应生成新的临时地址。

可以按如下所示表示临时地址的持续时间：

n	*n* 秒数（缺省值）
n h	*n* 小时数 (h)
n d	*n* 天数 (d)

▼ 如何配置临时地址

1 以主管理员或超级用户身份登录到 IPv6 主机。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 如有必要，请在主机的接口上启用 IPv6。

请参阅第 150 页中的 “如何启用当前会话的 IPv6 接口”。

3 编辑 `/etc/inet/ndpd.conf` 文件以打开临时地址生成功能。

- 要在主机的所有接口上配置临时地址，请将以下行添加到 `/etc/inet/ndpd.conf` 中：

  ```
  ifdefault TmpAddrsEnabled true
  ```

- 要配置特定接口的临时地址，请将以下一行添加到 `/etc/inet/ndpd.conf` 中：

  ```
  if interface TmpAddrsEnabled true
  ```

4 （可选）指定临时地址的有效生命周期。

`ifdefault TmpValidLifetime` *duration*

此语法为主机上的所有接口指定有效生命周期。*duration* 的值应当以秒、小时或天为单位。缺省的有效生命周期为 7 天。另外，还可以使用带有 `if` *interface* 关键字的 `TmpValidLifetime` 来为特定接口的临时地址指定有效生命周期。

5 （可选）为临时地址指定首选生命周期，在此之后临时地址将过时。

if *interface* **TmpPreferredLifetime** *duration*

此语法为特定接口的临时地址指定首选生命周期。缺省的首选生命周期为一天。另外，还可以使用带有 `ifdefault` 关键字的 `TmpPreferredLifetime` 来为主机所有接口上的临时地址指定首选生命周期。

注 – 缺省地址选择可为已经过时的 IPv6 地址指定较低的优先级。如果某个 IPv6 临时地址已过时，则缺省地址选择会将未过时的地址选作包的源地址。未过时的地址可能是自动生成的 IPv6 地址，也可能是接口的 IPv4 地址。有关缺省地址选择的更多信息，请参见第 197 页中的 “管理缺省地址选择”。

---

6 （可选）指定地址过时之前的前导时间，在这段时间内，主机应生成新的临时地址。

ifdefault TmpRegenAdvance duration

此语法可为主机上所有接口的临时地址指定地址过时之前的前导时间。缺省值是 5 秒。

7 更改 **in.ndpd** 守护进程的配置。

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

8 运行 **ifconfig -a6** 命令来验证临时地址是否已创建，如示例 7–5 中所示。

在 `ifconfig` 的输出中，接口定义所在的行中应包含 `TEMPORARY` 一词。

示例 7–4 /etc/inet/ndpd.conf 文件中的临时地址变量

以下示例显示了针对主网络接口启用了临时地址的 `/etc/inet/ndpd.conf` 文件片段。

```
ifdefault TmpAddrsEnabled true

ifdefault TmpValidLifetime 14d

ifdefault TmpPreferredLifetime 7d

ifdefault TmpRegenAdvance 6s
```

示例 7–5 启用临时地址后的 ifconfig -a6 命令输出

此示例显示了创建临时地址之后 `ifconfig` 命令的输出。

```
# ifconfig -a6
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
     inet6 ::1/128
hme0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
     ether 8:0:20:b9:4c:54
```

```
     inet6 fe80::a00:20ff:feb9:4c54/10
hme0:1: flags=2080841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
     inet6 2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
hme0:2: flags=802080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6,TEMPORARY> mtu 1500 index 2
      inet6 2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

请注意，`hme0:2` 接口后面的行中包括单词 `TEMPORARY`。此名称表示地址 `2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64` 具有临时接口 ID。

另请参见

- 要为 IPv6 地址设置名称服务支持，请参见第 172 页中的 “针对 IPv6 配置名称服务支持”。
- 要为服务器配置 IPv6 地址，请参见第 161 页中的 “如何配置用户指定的 IPv6 标记”。
- 要监视 IPv6 节点上的活动，请参见第 8 章，管理 TCP/IP 网络（任务）。

配置 IPv6 标记

如第 67 页中的 “IPv6 寻址概述”中所述，IPv6 地址的 64 位接口 ID 又称作**标记**。在地址自动配置过程中，该标记与接口的 MAC 地址相关联。大多数情况下，非路由节点（即 IPv6 主机和服务器）应当使用为其自动配置的标记。

但是，对于在系统维护过程中经常需要交换接口的服务器，使用自动配置的标记可能会产生问题。如果更换接口卡，则 MAC 地址也会随之更改。因此，依赖稳定 IP 地址的服务器将会遇到问题。网络基础结构的各个部分（如 DNS 或 NIS）可能已经存储了服务器接口的特定 IPv6 地址。

为了避免出现地址更改问题，可以手动配置要用作 IPv6 地址中接口 ID 的标记。要创建此标记，需要指定一个 64 位或更少的十六进制数字，使其占用 IPv6 地址的接口 ID 部分。在后续的地址自动配置过程中，相邻节点搜索协议不会基于接口的 MAC 地址创建接口 ID。相反，手动创建的标记将成为接口 ID。此标记始终被指定给该接口，即使更换了卡也是如此。

注 – 用户指定的标记和临时地址之间的区别在于，临时地址是随机生成的，而不是由用户显式创建的。

▼ 如何配置用户指定的 IPv6 标记

接下来的说明对于经常更换接口的服务器尤其有用。它们也可用于在任何 IPv6 节点上配置用户指定的标记。

1 验证要配置标记的接口是否已经过检测。

必须先检测接口，然后才能为其 IPv6 地址配置标记。

```
# ifconfig a6
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
```

```
        ether 0:3:ba:13:14:e1
        inet6 fe80::203:baff:fe13:14e1/10
```

此输出显示网络接口 `qfe0` 已经过检测并且具有链路本地地址 `fe80::203:baff:fe13:14e1/10`。此地址是在安装过程中自动配置的。

2 **创建一个或多个要用作节点接口标记的 64 位十六进制数字。有关标记的示例，请参阅 第 71 页中的“链路本地单播地址”。**

3 **配置每个接口的标记。**

对于每个要具有用户指定接口 ID（标记）的接口，请使用以下形式的 `ifconfig` 命令：

`ifconfig` *interface* `inet6 token` *address/64*

例如，可使用以下命令配置 `qfe0` 的标记：

```
# ifconfig qfe0 inet6 token ::1a:2b:3c:4d/64
```

对于要具有用户指定标记的每个接口，重复该步骤。

4 **（可选）使新的 IPv6 地址在重新引导过程中持续保留。**

a. **对于每个配置了标记的接口，编辑或创建 `/etc/hostname6.`*interface* 文件。**

b. **在每个 `/etc/hostname6.`*interface* 文件的末尾添加以下文本：**

`token ::`*token-name*`/64`**

例如，可以在 `/etc/hostname6.`*interface* 文件的末尾添加以下文本：

```
token ::1a:2b:3c:4d/64
```

在系统重新引导之后，在 `/etc/hostname6.` *interface* 文件中配置的标记将应用于接口的 IPv6 地址。此 IPv6 地址在后续的重新引导过程中仍会持续保留。

5 **使用所做更改更新 IPv6 守护进程。**

```
# pkill -HUP in.ndpd
```

示例 7–6 在 IPv6 接口上配置用户指定的标记

在以下示例中，`bge0:1` 接口具有自动配置的 IPv6 地址。子网前缀 `2001:db8:3c4d:152:/64` 由节点本地链路上的路由器通告。接口 ID `2c0:9fff:fe56:8255` 是用 `bge0:1` 的 MAC 地址生成的。

```
# ifconfig -a6
lo0: flags=2002000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
bge0: flags=2100801 <UP,MULTICAST,IPv6> mtu 1500 index 5
        inet6 fe80::2c0:9fff:fe56:8255/10
        ether 0:c0:9f:56:82:55
```

```
bge0:1: flags=2180801 <UP, MULTICAST,ADDRCONF,IPv6>mtu 1500 index 5
        inet6 2001:db8:3c4d:152:c0:9fff:fe56:8255/64
# ifconfig bge0 inet6 token ::1a:2b:3c:4d/64
# vi /etc/hostname6.bge0
token ::1a:2b:3c:4d/64
# pkill -HUP in.ndpd
# ifconfig -a6
lo0: flags=2002000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
bge0: flags=2100801 <UP,MULTICAST,IPv6> mtu 1500 index 5
        inet6 fe80::2c0:9fff:fe56:8255/10
        ether 0:c0:9f:56:82:55
bge0:1: flags=2180801 <UP, MULTICAST,ADDRCONF,IPv6>mtu 1500 index 5
        inet6 2001:db8:3c4d:152:1a:2b:3c:4d/64
```

配置标记之后，`bge0:1` 的第二个状态行上的全局地址现在包含为其配置的接口 ID `1a:2b:3c:4d`。

另请参见

- 要使用服务器的 IPv6 地址更新名称服务，请参见第 172 页中的 “针对 IPv6 配置名称服务支持”。
- 要监视服务器性能，请参见第 8 章，管理 TCP/IP 网络（任务）。

在服务器上管理启用了 IPv6 的接口

如果计划在服务器上配置 IPv6，则在服务器的接口上启用 IPv6 时，必须做出几个决定。所做的决定会影响用于配置接口 IPv6 地址的接口 ID（又称作**标记**）的策略。

▼ 如何在服务器接口上启用 IPv6

开始之前 下一过程假定以下情况成立：

- 已在服务器上安装了 Oracle Solaris。
- 已经使用第 149 页中的 “配置 IPv6 接口”中的过程，在 Oracle Solaris 安装过程中或安装之后在服务器接口上启用了 IPv6。

如果适用，请升级应用程序软件以支持 IPv6。请注意，许多在 IPv4 协议栈上运行的应用程序也能在 IPv6 上成功运行。有关更多信息，请参阅第 79 页中的 “如何准备网络服务以支持 IPv6”。

1. **在服务器上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **确保与服务器在同一链路上的路由器上配置了 IPv6 子网前缀。**

   有关更多信息，请参阅第 154 页中的 “配置 IPv6 路由器”。

3 **对服务器上启用了 IPv6 的接口，使用适当的接口 ID 策略。**

缺省情况下，在创建 IPv6 地址的接口 ID 部分时，IPv6 地址自动配置会使用接口的 MAC 地址。如果接口的 IPv6 地址是已知的，则使用一个接口交换另一个接口会导致问题。新接口的 MAC 地址将会不同。在地址自动配置过程中，会生成新的接口 ID。

- 对于不打算替换的启用了 IPv6 的接口，请使用自动配置的 IPv6 地址，如第 73 页中的 “IPv6 地址自动配置”中所述。
- 对于必须匿名显示在本地网络外部的启用了 IPv6 的接口，请考虑对接口 ID 使用随机生成的标记。有关说明和示例，请参阅第 159 页中的 “如何配置临时地址”。
- 对于计划定期交换的启用了 IPv6 的接口，请为接口 ID 创建标记。有关说明和示例，请参阅第 161 页中的 “如何配置用户指定的 IPv6 标记”。

针对 IPv6 支持配置隧道所需的任务（任务列表）

下表列出了各种配置不同类型的 IPv6 隧道的任务。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
手动配置 IPv6 over IPv4 隧道。	手动创建经由 IPv4 网络的 IPv6 隧道，此解决方案可用于在大型企业网络（主要是 IPv4 企业网络）中访问远程 IPv6 网络。	第 165 页中的 “如何手动配置 IPv6 over IPv4 隧道”
手动配置 IPv6 over IPv6 隧道。	手动配置经由 IPv6 网络的 IPv6 隧道，此解决方案通常用在大型企业网络中。	第 166 页中的 “如何手动配置 IPv6 over IPv6 隧道”
手动配置 IPv4 over IPv6 隧道。	手动配置经由 IPv6 网络的 IPv4 隧道，此解决方案通常用在同时包含 IPv4 网络和 IPv6 网络的大型网络中。	第 167 页中的 “如何配置 IPv4 over IPv6 隧道”
自动配置 IPv6 over IPv4 隧道（6to4 隧道）。	创建 6to4 自动隧道，此解决方案用于通过 Internet 访问外部的 IPv6 站点。	第 167 页中的 “如何配置 6to4 隧道”
在 6to4 路由器和 6to4 中继路由器之间配置隧道。	使用 `6to4relay` 命令启用连接到 6to4 中继路由器的隧道。	第 170 页中的 “如何配置通往 6to4 中继路由器的 6to4 隧道”

针对 IPv6 支持配置隧道

在大型 IPv4 网络中，IPv6 网络通常是隔离的实体。IPv6 网络上的节点可能需要与隔离的 IPv6 网络上的节点通信（在企业内或以远程方式进行）。通常，尽管 IPv6 主机也可以充当隧道端点，但是仍需要在 IPv6 路由器之间配置一条隧道。有关隧道规划的信息，请参阅第 81 页中的 “在网络拓扑中规划隧道”。

可以为 IPv6 网络设置自动或手动配置的隧道。Oracle Solaris IPv6 实现支持下列类型的隧道封装：

- IPv6 over IPv4 隧道
- IPv6 over IPv6 隧道
- IPv4 over IPv6 隧道
- 6to4 隧道

有关隧道的概念性说明，请参见第 250 页中的 “IPv6 隧道”。

▼ 如何手动配置 IPv6 over IPv4 隧道

本过程介绍如何设置从 IPv6 节点经由 IPv4 网络到达远程 IPv6 节点的隧道。

1 **以主管理员或超级用户身份登录到本地隧道的端点。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **创建 `/etc/hostname6.ip.tun` *n* 文件。**

其中 *n* 表示隧道编号，第一个隧道的编号从零开始。然后，按照下列子步骤来添加项：

a. **添加隧道的源地址和目标地址。**

```
tsrc IPv4-source-address tdst IPv4-destination-address up
```

b. **（可选）为源 IPv6 地址和目标 IPv6 地址各添加一个逻辑接口。**

```
addif IPv6-source-address  IPv6-destination-address
```

如果希望为该接口自动配置地址，请忽略该子步骤，无需为隧道配置链路本地地址。

3 **重新引导系统。**

4 **对隧道的另一端重复该任务。**

示例 7–7 IPv6 over IPv4 手动隧道的 /etc/hostname6.ip.tun 文件中的项

此样例 /etc/hostname6.ip.tun 文件显示了为其手动配置全局源地址和全局目标地址的隧道。

```
tsrc 192.168.8.20 tdst 192.168.7.19 up
addif 2001:db8:3c4d:8::fe12:528 2001:db8:3c4d:7:a00:20ff:fe12:1234 up
```

▼ 如何手动配置 IPv6 over IPv6 隧道

本过程介绍如何设置从 IPv6 节点经由 IPv6 网络到达远程 IPv6 节点的隧道。

1. **以主管理员或超级用户身份登录到本地隧道的端点。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **创建 /etc/hostname6.ip6.tun *n* 文件。**

   n 可使用值 0、1、2 等。然后，按照下列子步骤来添加项：

   a. **添加隧道的源地址和目标地址。**

      `tsrc` *IPv6-source-address* `tdst` *IPv6-destination-address*
      IPv6-packet-source-address *IPv6-packet-destination-address* `up`

   b. **（可选）为源 IPv6 地址和目标 IPv6 地址各添加一个逻辑接口。**

      `addif` *IPv6-source-address* *IPv6-destination-address* `up`

      如果希望为该接口自动配置地址，请忽略该步骤，无需为隧道配置链路本地地址。

3. **重新引导系统。**

4. **对于隧道的另一端重复此过程。**

示例 7–8 IPv6 over IPv6 隧道的 /etc/hostname6.ip6.tun 文件中的项

此示例显示了 IPv6 over IPv6 隧道的项。

```
tsrc 2001:db8:3c4d:22:20ff:0:fe72:668c tdst 2001:db8:3c4d:103:a00:20ff:fe9b:a1c3
fe80::4 fe80::61 up
```

▼ 如何配置 IPv4 over IPv6 隧道

本过程介绍如何配置从一台 IPv4 主机经由 IPv6 网络到达另一台 IPv4 主机的隧道。如果您的公司网络是异构网络，并且 IPv4 子网由 IPv6 子网分隔，则可以使用此过程。

1 **以主管理员或超级用户身份登录到本地 IPv4 隧道的端点。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **创建 `/etc/hostname.ip6.tun`*n* 文件。**

n 可使用值 0、1、2 等。然后，按照下列步骤来添加项：

a. **添加隧道的源地址和目标地址。**

```
tsrc IPv6-source-address tdst IPv6-destination-address
```

b. **（可选）为源 IPv6 地址和目标 IPv6 地址各添加一个逻辑接口。**

```
addif IPv6-source-address  IPv6-destination-address up
```

3 **重新引导本地主机。**

4 **对于隧道的另一端重复此过程。**

示例 7–9 IPv4 over IPv6 隧道的 `/etc/hostname6.ip6.tun` 中的项

此示例显示了 IPv4 over IPv6 隧道的项。

```
tsrc 2001:db8:3c4d:114:a00:20ff:fe72:668c tdst 2001:db8:3c4d:103:a00:20ff:fe9b:a1c3
10.0.0.4 10.0.0.61 up
```

▼ 如何配置 6to4 隧道

如果 IPv6 网络需要与远程 IPv6 网络通信，请考虑使用 6to4 自动隧道。6to4 隧道的配置过程包括将边界路由器配置为 *6to4* 路由器。6to4 路由器可充当您的网络与远程 IPv6 网络上的端点路由器之间的 6to4 隧道的端点。

开始之前 在 IPv6 网络上配置 6to4 路由之前，必须已经完成以下操作：

- 已经按照第 158 页中的 “修改主机和服务器的 IPv6 接口配置”中所述在将来的 6to4 站点的所有相应节点上配置了 IPv6。
- 至少已经选择了一个连接到 IPv4 网络的路由器作为 6to4 路由器。
- 已经为 IPv4 网络配置了在将来的 6to4 路由器接口上全局唯一的 IPv4 地址。该 IPv4 地址必须是静态的。

注 – 请勿使用动态分配的 IPv4 地址，如第 12 章，关于 DHCP（概述）中所述。全局动态分配的地址可能会随着时间而更改，这会对 IPv6 寻址计划造成不良影响。

1 **以主管理员或超级用户身份登录到将来的 6to4 路由器。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **通过创建 `/etc/hostname6.ip.6to4tun0` 文件在路由器上配置 6to4 伪接口。**

- 如果计划使用建议的约定（子网 ID=0，主机 ID=1），请针对 `/etc/hostname6.ip.6to4tun0` 使用短格式：

  `tsrc` *IPv4-address* `up`

- 如果计划对子网 ID 和主机 ID 使用其他约定，请针对 `/etc/hostname6.ip.6to4tun0` 使用长格式：

  `tsrc` *IPv4-address* `2002:`*IPv4-address*`:`*subnet-ID*`:`*interface-ID*`:/64 up`

以下是 `/etc/hostname6.ip.6to4tun0` 的必需参数：

`tsrc`	表示此接口用作隧道源。
IPv4-address	以点分十进制格式指定在作为 6to4 伪接口的物理接口上配置的 IPv4 地址。

其余参数是可选的。但是，如果指定了一个可选参数，则必须指定所有的可选参数。

`2002`	指定 6to4 前缀。
IPv4-address	以十六进制表示法指定伪接口的 IPv4 地址。
subnet-ID	以十六进制表示法指定除 0 以外的子网 ID。
interface-ID	指定除 1 以外的接口 ID。
`/64`	表示 6to4 前缀的长度为 64 位。
`up`	将 6to4 接口配置为 "up"。

注 – 网络上的两个 IPv6 隧道不能具有相同的源地址和目标地址。否则，包会被丢弃。如果 6to4 路由器还通过 `atun` 命令执行隧道连接，则可能会发生这种类型的事件。有关 `atun` 的信息，请参阅 `tun(7M)` 手册页。

3 **（可选）在路由器上创建其他 6to4 伪接口。**

每个将来的 6to4 伪接口都必须具有一个已配置的全局唯一的 IPv4 地址。

4 **重新引导 6to4 路由器。**

5 **验证接口状态。**

```
# ifconfig ip.6to4tun0 inet6
```

如果接口已正确配置，则将接收到以下类似输出：

```
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6> mtu 1480 index 11
        inet tunnel src 111.222.33.44
        tunnel hop limit 60
        inet6 2002:6fde:212c:10:/64
```

6 **编辑 `/etc/inet/ndpd.conf` 文件以通告 6to4 路由。**

有关详细信息，请参阅 `ndpd.conf(4)` 手册页。

a. **在第一行中指定要接收通告的子网。**

创建具有以下格式的 `if` 项：

`if` *subnet-interface* `AdvSendAdvertisements 1`

例如，要向连接到 `hme0` 接口的子网通告 6to4 路由，请将 *subnet-interface* 替换为 `hme0`。

```
if hme0 AdvSendAdvertisements 1
```

b. **在通告的第二行中添加 6to4 前缀。**

创建具有以下格式的 `prefix` 项：

`prefix 2002:`*IPv4-address*`:`*subnet-ID*`::/64` *subnet-interface*

7 **重新引导路由器。**

或者，可以向 `/etc/inet/in.ndpd` 守护进程发出 `sighup`，以便开始发送路由器通告。要接收 6to4 前缀的每个子网上的 IPv6 节点现在可以使用 6to4 派生地址自动进行配置。

8 **将节点的新 6to4 派生地址添加到在 6to4 站点上使用的名称服务中。**

有关说明，请转至第 172 页中的 “针对 IPv6 配置名称服务支持”。

示例 7–10 6to4 路由器配置（短形式）

下面举例说明了 `/etc/hostname6.ip.6to4tun0` 的短形式：

```
# cat /etc/hostname6.ip.6to4tun0
tsrc 111.222.33.44 up
```

示例 7–11 6to4 路由器配置（长形式）

下面举例说明了 `/etc/hostname6.ip.6to4tun0` 的长形式：

```
# cat /etc/hostname6.ip.6to4tun0
tsrc 111.222.33.44 2002:6fde:212c:20:1/64 up
```

示例 7–12 显示 6to4 伪接口的 ifconfig 输出

以下样例说明了针对 6to4 伪接口的 ifconfig 命令的输出：

```
# ifconfig ip.6to4tun0 inet6
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6> mtu 1480 index 11
        inet tunnel src 192.168.87.188
        tunnel hop limit 60
        inet6 2002:c0a8:57bc::1/64
```

示例 7–13 /etc/inet/ndpd.conf 中的 6to4 通告

以下样例 /etc/inet/ndpd.conf 文件将在两个子网上通告 6to4 路由：

```
if qfe0 AdvSendAdvertisements 1
prefix  2002:c0a8:57bc:10::/64 qfe0

if qfe1 AdvSendAdvertisements 1
prefix  2002:c0a8:57bc:2::/64 qfe1
```

更多信息 在 6to4 站点上配置多个路由器

对于多路由器站点，可能需要进一步配置 6to4 路由器后面的路由器以支持 6to4。如果站点使用 RIP，则必须在每个非 6to4 路由器上配置通往 6to4 路由器的静态路由。如果使用商业路由协议，则无需创建通往 6to4 路由器的静态路由。

▼ 如何配置通往 6to4 中继路由器的 6to4 隧道

注意 – 由于 6to4 中继路由器存在重要的安全问题，因此，在缺省情况下，Oracle Solaris 中会禁用 6to4 中继路由器支持。请参见建立通往 6to4 中继路由器的隧道时的安全问题。

开始之前 在启用通往 6to4 中继路由器的隧道之前，必须先完成下列任务：

- 按照第 167 页中的 “如何配置 6to4 隧道”中的说明在站点上配置了 6to4 路由器
- 检查建立通往 6to4 中继路由器的隧道连接时涉及到的安全问题

1 **以主管理员或超级用户身份登录到 6to4 路由器。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **使用以下格式之一启用通往 6to4 中继路由器的隧道：**

- 启用通往任播 6to4 中继路由器的隧道。

```
# /usr/sbin/6to4relay -e
```

-e 选项可用于在 6to4 路由器和任播 6to4 中继路由器之间设置隧道。任播 6to4 中继路由器具有已知的 IPv4 地址 192.88.99.1。物理位置距离您的站点最近的任播中继路由器将成为 6to4 隧道的端点。该中继路由器随后将在 6to4 站点和本机 IPv6 站点之间转发包。

有关任播 6to4 中继路由器的详细信息，请参阅 RFC 3068, "An Anycast Prefix for 6to4 Relay Routers" (ftp://ftp.rfc-editor.org/in-notes/rfc3068.txt)。

- 启用通往特定 6to4 中继路由器的隧道。

```
# /usr/sbin/6to4relay -e -a relay-router-address
```

-a 选项表示后面将跟有一个特定路由器地址。请将 *relay-router-address* 替换为用以启用隧道的特定 6to4 中继路由器的 IPv4 地址。

除非删除 6to4 隧道的伪接口，否则通往 6to4 中继路由器的隧道将一直保持活动状态。

3 **如果不再需要隧道，请删除通往 6to4 中继路由器的隧道：**

```
# /usr/sbin/6to4relay -d
```

4 **（可选）使通往 6to4 中继路由器的隧道在重新引导过程中持续保留。**

您的站点可能迫切要求通往 6to4 中继路由器的隧道在 6to4 路由器每次重新引导时都进行恢复。要支持此方案，必须执行下列操作：

a. **编辑 /etc/default/inetinit 文件。**

需要修改的行位于该文件的末尾。

b. **将 ACCEPT6TO4RELAY=NO 行中的 "NO" 值更改为 "YES"。**

c. **（可选）创建通往特定 6to4 中继路由器的隧道，该隧道在重新引导过程中持续保留。**

对于 RELAY6TO4ADDR 参数，请将 192.88.99.1 地址更改为要使用的 6to4 中继路由器的 IPv4 地址。

示例 7–14 获取有关 6to4 中继路由器支持的状态信息

可以使用 `/usr/bin/6to4relay` 命令来确定对 6to4 中继路由器是否启用了的支持。以下示例显示了禁用 6to4 中继路由器支持（此为 Oracle Solaris 中的缺省设置）时的输出：

```
# /usr/sbin/6to4relay
6to4relay: 6to4 Relay Router communication support is disabled.
```

启用对 6to4 中继路由器的支持时，将接收到以下输出：

```
# /usr/sbin/6to4relay
6to4relay: 6to4 Relay Router communication support is enabled.
IPv4 remote address of Relay Router=192.88.99.1
```

针对 IPv6 配置名称服务支持

本节介绍如何将 DNS 和 NIS 名称服务配置为支持 IPv6 服务。

注 – LDAP 无需执行特定于 IPv6 的配置任务即可支持 IPv6。

有关管理 DNS、NIS 和 LDAP 的全部详细信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

▼ 如何向 DNS 中添加 IPv6 地址

1. **以主管理员或超级用户身份登录到主 DNS 服务器或辅助 DNS 服务器。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **通过为每个启用了 IPv6 的节点添加 AAAA 记录，来编辑相应的 DNS 区域文件：**

   hostname `IN AAAA` *host-address*

3. **编辑 DNS 反向区域文件并添加 PTR（指针）记录：**

   hostaddress `IN PTR` *hostname*

   有关 DNS 管理的详细信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

示例 7–15 DNS 反向区域文件

此示例显示了反向区域文件中的 IPv6 地址。

```
$ORIGIN    ip6.int.
8.2.5.0.2.1.e.f.f.f.9.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0 \
    IN        PTR        vallejo.Eng.apex.COM.
```

向 NIS 中添加 IPv6 地址

在 Solaris 10 11/06 及更早的版本中，已为 NIS 添加了两个映射： `ipnodes.byname` 和 `ipnodes.byaddr`。这些映射中既包含 IPv4 主机名和地址之间的关联，又包含 IPv6 主机名和地址之间的关联。可识别 IPv6 的工具使用的是 `ipnodes` NIS 映射。`hosts.byname` 和 `hosts.byaddr` 映射中仅包含 IPv4 主机名和地址之间的关联。为了便于现有的应用程序使用，这些映射仍保持原样。对 `ipnodes` 映射的管理与对 `hosts.byname` 和 `hosts.byaddr` 映射的管理类似。对于 Solaris 10 11/06 而言，有一点很重要，在用 IPv4 地址更新**主机**映射时，*ipnode* 映射也会使用相同信息进行更新。

注 – Oracle Solaris 10 的后续版本不再使用 `ipnodes` 映射。现在，`ipnodes` 映射的 IPv6 功能保留在 `hosts` 映射中。

有关管理 NIS 映射的说明，请参阅《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》中的第 5 章 “Setting Up and Configuring NIS Service”。

▼ 如何显示 IPv6 名称服务信息

可以使用 `nslookup` 命令显示 IPv6 名称服务信息。

1. **使用您的用户帐户运行 `nslookup` 命令。**

   ```
   % /usr/sbin/nslookup
   ```

   此时会出现缺省的服务器名称和地址，后跟 `nslookup` 命令的尖括号提示符。

2. **在尖括号提示符下键入以下命令，查看有关特定主机的信息：**

   ```
   >set q=any
   >hostname
   ```

3. **键入以下命令，以便仅查看 AAAA 记录：**

   ```
   >set q=AAAA
   hostname
   ```

4. **键入 `exit`，退出 `nslookup` 命令。**

示例 7–16 使用 nslookup 显示 IPv6 信息

此示例显示了 nslookup 在 IPv6 网络环境中的输出结果。

```
%  /usr/sbin/nslookup
Default Server:  dnsserve.local.com
Address:  10.10.50.85
> set q=AAAA
> host85
Server:  dnsserve.local.com
Address:  10.10.50.85

host85.local.com      IPv6 address = 2::9256:a00:fe12:528
> exit
```

▼ 如何验证 DNS IPv6 PTR 记录是否已正确更新

在此过程中，可使用 nslookup 命令显示 DNS IPv6 的 PTR 记录。

1. **使用您的用户帐户运行 nslookup 命令。**

   ```
   % /usr/sbin/nslookup
   ```

   此时会出现缺省的服务器名称和地址，后跟 nslookup 命令的尖括号提示符。

2. **在尖括号提示符下键入以下命令，查看 PTR 记录：**

   ```
   >set q=PTR
   ```

3. **键入 exit，退出该命令。**

示例 7–17 使用 nslookup 显示 PTR 记录

以下示例显示了使用 nslookup 命令时所显示的 PTR 记录。

```
%  /usr/sbin/nslookup
Default Server:  space1999.Eng.apex.COM
Address:  192.168.15.78
> set q=PTR
> 8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int

8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int name =
vallejo.ipv6.Eng.apex.COM
ip6.int nameserver = space1999.Eng.apex.COM
> exit
```

▼ 如何通过 NIS 显示 IPv6 信息

在此过程中，可使用 ypmatch 命令，通过 NIS 显示 IPv6 信息：

- **使用您的用户帐户键入以下命令，显示 NIS 中的 IPv6 地址：**

  % **ypmatch** *hostname* **hosts ipnodes.**byname

  此时会显示有关指定 *hostname* 的信息。

注 – Solaris 10 11/06 之后的 Oracle Solaris 发行版中不再包含 ipnodes 映射。现在，ipnodes 的 IPv6 功能保留在 hosts 映射中。

示例 7–18 ypmatch 命令输出的 IPv6 地址

对于 Solaris 10 11/06 及更早的版本，以下样例说明了针对 ipnodes.byname 数据库执行 ypmatch 操作的结果。

```
% ypmatch farhost hosts ipnodes.byname
2001:0db8:3c4d:15:a00:20ff:fe12:5286       farhost
```

▼ 如何显示与名称服务无关的 IPv6 信息

此过程仅适用于 Solaris 10 11/06 及更早的版本。对于后续版本，可以针对 hosts 数据库执行相同的操作。

- **使用您的用户帐户键入以下命令：**

  % **getent ipnodes** *hostname*

  此时会显示有关指定 *host-name* 的信息。

示例 7–19 显示 ipnodes 数据库中的 IPv6 信息

以下样例说明了 getent 命令的输出：

```
% getent ipnodes vallejo

2001:0db8:8512:2:56:a00:fe87:9aba    myhost myhost
fe80::56:a00:fe87:9aba     myhost myhost
```

◆ ◆ ◆ 第 8 章

管理 TCP/IP 网络（任务）

本章介绍管理 TCP/IP 网络的任务。本章包含以下主题：

注 – 要监视网络接口，请参见第 178 页中的 “使用 `ifconfig` 命令监视接口配置”。

这些任务假设您的站点拥有正常运行的 TCP/IP 网络，该网络仅启用了 IPv4 或启用了双栈 IPv4/IPv6。如果希望在站点实施 IPv6 但尚未实现，请参阅以下各章了解更多信息

- 有关如何规划 IPv6 实现，请参阅第 4 章，规划 IPv6 网络（任务）。
- 要配置 IPv6 和创建双栈网络环境，请参阅第 7 章，配置 IPv6 网络（任务）。

主要的 TCP/IP 管理任务（任务列表）

下表列出了进行初始配置后的其他网络管理任务，例如显示网络信息。此表中包含对各项任务要完成的工作的说明，以及当前文档中详细介绍用于执行任务的特定步骤的章节。

任务	说明	参考
显示有关接口的配置信息。	确定系统上每个接口的当前配置。	第 178 页中的 “如何获取有关特定接口的信息”

任务	说明	参考
显示指定的接口地址。	确定为本地系统上所有接口指定的地址。	第 180 页中的 “如何显示指定的接口地址”
按协议显示统计信息。	监视特定系统上网络协议的性能。	第 182 页中的 “如何按协议显示统计信息”
显示网络状态。	通过显示所有套接字和路由表项来监视系统。输出包括 IPv4 的 inet 地址族和 IPv6 的 inet6 地址族。	第 185 页中的 “如何显示套接字的状态”
显示网络接口的状态。	监视网络接口的性能，这对于解决传输问题非常有用。	第 184 页中的 “如何显示网络接口状态”
显示包传输状态。	监视包在网络上传送时的状态。	第 187 页中的 “如何显示特定地址类型的包的传输状态”
控制与 IPv6 相关的命令的显示输出。	控制 ping、netstat、 ifconfig 和 traceroute 命令的输出。创建名为 inet_type 的文件。在此文件中设置 DEFAULT_IP 变量。	第 190 页中的 “如何控制与 IP 相关的命令的显示输出”
监视网络通信。	使用 snoop 命令显示所有 IP 包。	第 196 页中的 “如何监视 IPv6 网络通信”
跟踪网络路由器已知的所有路由。	使用 traceroute 命令显示所有路由。	第 193 页中的 “如何跟踪所有路由”

使用 ifconfig 命令监视接口配置

可以使用 ifconfig 命令为接口手动指定 IP 地址并手动配置接口参数。此外，Oracle Solaris 启动脚本还运行 ifconfig 来配置伪接口，例如 6to4 隧道端点。

本书介绍多个使用通用 ifconfig 命令的各种选项的任务。有关此命令及其选项和变量的完整说明，请参阅 ifconfig(1M) 手册页。ifconfig 的基本语法如下所示：

ifconfig *interface [protocol-family]*

▼ 如何获取有关特定接口的信息

使用 ifconfig 命令可确定有关特定系统的接口的基本信息。例如，通过一个简单的 ifconfig 查询可获得以下信息：

- 系统上所有接口的设备名称
- 为接口指定的所有 IPv4 地址以及所有 IPv6 地址（如果适用）
- 当前是否已配置这些接口

以下过程说明了如何使用 ifconfig 命令来获取有关系统接口的基本配置信息。

1 **在本地主机上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **获取有关特定接口的信息。**

```
# ifconfig interface
```

ifconfig 命令的输出格式如下：

- 状态行

  ifconfig 命令输出中的第一行包括接口名称以及当前与接口关联的状态标志。此外，状态行还包括为特定接口配置的最大传输单元 (Maximum Transmission Unit, MTU) 以及索引号。使用状态行可确定接口的当前状态。

- IP 地址信息行

  ifconfig 输出的第二行包括为接口配置的 IPv4 地址或 IPv6 地址。对于 IPv4 地址，还显示已配置的网络掩码和广播地址。

- MAC 地址行

  以超级用户或类似角色的身份运行 ifconfig 命令时，ifconfig 输出包含第三行。对于 IPv4 地址，第三行显示了为接口指定的 MAC 地址（以太网层地址）。对于 IPv6 地址，输出中的第三行显示了 IPv6 in.ndpd 守护进程根据 MAC 地址生成的链路本地地址。

示例 8–1 使用 ifconfig 命令生成的基本接口信息

以下示例说明了如何使用 ifconfig 命令来获取有关特定主机上的 eri 接口的信息。

```
# ifconfig eri
eri0: flags=863<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 1
      inet 10.0.0.112 netmask ffffff80 broadcast 10.8.48.127
      ether 8:0:20:b9:4c:54
```

下表描述 ifconfig 查询中的变量信息，同时说明如何在屏幕上显示变量以及提供的信息类型。使用上述输出作为示例。

变量	屏幕输出	说明
接口名称	eri0	指示已在 ifconfig 命令中请求其状态的接口的设备名称。
接口状态	flags=863<UP	显示接口的状态，包括当前与接口关联的所有标志。可以据此确定接口当前已初始化 (UP) 还是未初始化 (DOWN)。

变量	屏幕输出	说明
广播状态	BROADCAST	指示接口支持 IPv4 广播。
传输状态	RUNNING	指示系统正在通过接口传输包。
多播状态	MULTICAST, IPv4	显示接口支持多播传输。示例中的接口支持 IPv4 多播传输。
最大传输单元	mtu 1500	显示此接口的最大传输大小为 1500 个八位字节。
IP 地址	inet 10.0.0.112	显示为接口指定的 IPv4 或 IPv6 地址。示例接口 eri0 的 IPv4 地址为 10.0.0.112。
网络掩码	netmask ffffff80	显示特定接口的 IPv4 网络掩码。请注意，IPv6 地址不使用网络掩码。
MAC 地址	ether 8:0:20:b9:4c:54	显示接口的以太网层地址。

▼ 如何显示指定的接口地址

路由器和多宿主主机具有多个接口，并且通常为每个接口指定多个 IP 地址。可以使用 ifconfig 命令来显示为系统接口指定的所有地址，还可以使用 ifconfig 命令仅显示指定的 IPv4 或 IPv6 地址。要另外显示接口的 MAC 地址，您必须首先以超级用户或相应角色的身份登录。

有关 ifconfig 命令的更多信息，请参见 ifconfig(1M) 手册页。

1. **在本地系统上，承担网络管理角色或成为超级用户。**

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **获取有关所有接口的信息。**

   您可以使用 ifconfig -a 命令的变体执行以下操作：

   - 查看系统上所有接口的所有地址。

     ```
     # ifconfig -a
     ```

   - 查看为系统接口指定的所有 IPv4 地址。

     ```
     # ifconfig -a4
     ```

   - 如果本地系统启用了 IPv6，则显示为系统接口指定的所有 IPv6 地址。

     ```
     ifconfig -a6
     ```

示例 8–2 显示所有接口的地址信息

此示例显示只具有一个主网络接口 (qfe0) 的主机的地址项。但是，ifconfig 输出显示当前为 qfe0 指定的三种形式的地址： 回送 (lo0)、IPv4 (inet) 和 IPv6 (inet6)。请注意，在输出的 IPv6 部分中，接口 qfe0 的行显示本地链路 IPv6 地址。qfe0 的第二个地址显示在 qfe0:1 行中。

```
% ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.112 netmask ffffff80 broadcast 10.0.0.127
        ether 8:0:20:b9:4c:54
lo0: flags=2000849 <UP,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 8:0:20:b9:4c:54
        inet6 fe80::a00:20ff:feb9:4c54/10
qfe0:1: flags=2080841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
        inet6 2001:db8:3c4d:48:a00:20ff:feb9:4c54/64
```

示例 8–3 显示所有 IPv4 接口的地址信息

此示例显示了为多宿主主机配置的 IPv4 地址。不需要以超级用户的身份登录便可运行此形式的 ifconfig 命令。

```
% ifconfig -a4
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.112 netmask ffffff80 broadcast 10.0.0.127
        ether 8:0:20:b9:4c:54
qfe1: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.118 netmask ffffff80 broadcast 10.0.0.127
        ether 8:0:20:6f:5e:17
```

示例 8–4 显示所有 IPv6 接口的地址信息

此示例仅显示了为特定主机配置的 IPv6 地址。不需要以超级用户的身份登录，便可运行此形式的 ifconfig 命令。

```
% ifconfig -a6
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 8:0:20:b9:4c:54
        inet6 fe80::a00:20ff:feb9:4c54/10
qfe0:1: flags=2080841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
        inet6 2001:db8:3c4d:48:a00:20ff:feb9:4c54/64
```

此 ifconfig 输出显示了为主机的单个接口指定的以下三种形式的 IPv6 地址：

lo0
: IPv6 回送地址。

`inet6 fe80::a00:20ff:feb9:4c54/10`
为主网络接口指定的链路本地地址。

`inet6 2001:db8:3c4d:48:a00:20ff:feb9:4c54/64`
IPv6 地址，包括子网前缀。输出中的 ADDRCONF 一词指示此地址是由主机自动配置的。

使用 netstat 命令监视网络状态

netstat 命令生成包含网络状态和协议统计信息的显示内容。可以通过表格形式显示 TCP、SCTP（流控制传输协议）和 UDP（用户数据报协议）端点的状态，还可以显示路由表信息和接口信息。

netstat 可显示各种类型的网络数据，具体取决于所选择的命令行选项。这些显示信息对于系统管理非常有价值。netstat 的基本语法如下所示：

`netstat [-m] [-n] [-s] [-i | -r] [-f` *address-family*`]`

本节介绍最常用的 netstat 命令选项。有关所有 netstat 选项的详细说明，请参阅 netstat(1M) 手册页。

▼ 如何按协议显示统计信息

netstat -s 选项显示 UDP、TCP、SCTP、ICMP 和 IP 协议的统计信息。

注 – 可以使用 Oracle Solaris 用户帐户获取 netstat 命令的输出。

- **显示协议状态。**

```
$ netstat -s
```

示例 8–5 网络协议统计信息

以下示例显示了 netstat -s 命令的输出。某些输出信息已被截断。输出可以指明存在协议问题的区域。例如，ICMPv4 和 ICMPv6 的统计信息可以指明 ICMP 协议发现错误的位置。

```
RAWIP
        rawipInDatagrams    =  4701     rawipInErrors       =     0
        rawipInCksumErrs    =     0     rawipOutDatagrams   =     4
        rawipOutErrors      =     0

UDP
        udpInDatagrams      = 10091     udpInErrors         =     0
        udpOutDatagrams     = 15772     udpOutErrors        =     0
```

```
TCP     tcpRtoAlgorithm     =     4     tcpRtoMin           =   400
        tcpRtoMax           = 60000     tcpMaxConn          =    -1
        .
        .
        tcpListenDrop       =     0     tcpListenDropQ0     =     0
        tcpHalfOpenDrop     =     0     tcpOutSackRetrans   =     0

IPv4    ipForwarding        =     2     ipDefaultTTL        =   255
        ipInReceives        =300182     ipInHdrErrors       =     0
        ipInAddrErrors      =     0     ipInCksumErrs       =     0
        .
        .
        ipsecInFailed       =     0     ipInIPv6            =     0
        ipOutIPv6           =     3     ipOutSwitchIPv6     =     0

IPv6    ipv6Forwarding      =     2     ipv6DefaultHopLimit =   255
        ipv6InReceives      = 13986     ipv6InHdrErrors     =     0
        ipv6InTooBigErrors  =     0     ipv6InNoRoutes      =     0
        .
        .
        rawipInOverflows    =     0     ipv6InIPv4          =     0

       ipv6OutIPv4         =     0     ipv6OutSwitchIPv4   =     0

ICMPv4  icmpInMsgs          = 43593     icmpInErrors        =     0
        icmpInCksumErrs     =     0     icmpInUnknowns      =     0
        .
        .
        icmpInOverflows     =     0

ICMPv6  icmp6InMsgs         = 13612     icmp6InErrors       =     0
        icmp6InDestUnreachs =     0     icmp6InAdminProhibs =     0
        .
        .
        icmp6OutGroupQueries=     0     icmp6OutGroupResps  =     2
        icmp6OutGroupReds   =     0

IGMP:
      12287 messages received
          0 messages received with too few bytes
          0 messages received with bad checksum
      12287 membership queries received
SCTP  sctpRtoAlgorithm     =  vanj
      sctpRtoMin           =  1000
      sctpRtoMax           = 60000
      sctpRtoInitial       =  3000
      sctpTimHearBeatProbe =     2
      sctpTimHearBeatDrop  =     0
      sctpListenDrop       =     0
      sctpInClosed         =     0
```

▼ 如何显示传输协议的状态

可以通过 netstat 命令显示传输协议的状态。有关详细信息，请参阅 netstat(1M) 手册页。

1. **显示系统上 TCP 和 SCTP 传输协议的状态。**

   ```
   $ netstat
   ```

2. **显示系统上特定传输协议的状态。**

   ```
   $ netstat -P transport-protocol
   ```

   transport-protocol 变量的值为 tcp、sctp 或 udp。

示例 8–6 显示 TCP 和 SCTP 传输协议的状态

此示例显示基本 netstat 命令的输出。请注意，仅显示与 IPv4 有关的信息。

```
$ netstat

TCP: IPv4
   Local Address      Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost-1.login      ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost-1.1014     mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT
SCTP:
Local Address    Remote Address  Swind  Send-Q  Rwind  Recv-Q StrsI/O  State
---------------- --------------  -----  ------ ------ ------  ------   -------
 *.echo            0.0.0.0            0       0 102400      0   128/1   LISTEN
 *.discard         0.0.0.0            0       0 102400      0   128/1   LISTEN
 *.9001            0.0.0.0            0       0 102400      0   128/1   LISTEN
```

示例 8–7 显示特定传输协议的状态

此示例显示指定了 netstat 的 -P 选项时的结果。

```
$ netstat -P tcp

TCP: IPv4
   Local Address      Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost.login        ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost.1014       mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT

TCP: IPv6
 Local Address    Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------- ---------------------- ------ ----- ------ ----------- -----
localhost.38983   localhost.32777       49152      0 49152      0 ESTABLISHED
localhost.32777   localhost.38983       49152      0 49152      0 ESTABLISHED
localhost.38986   localhost.38980       49152      0 49152      0 ESTABLISHED
```

▼ 如何显示网络接口状态

netstat 命令的 i 选项显示本地系统上配置的网络接口的状态。可以使用此选项确定系统在每个网络中传输和接收的包数。

- **显示网络中接口的状态。**

```
$ netstat -i
```

示例 8–8 网络接口状态显示

下面的示例显示通过主机接口的 IPv4 和 IPv6 包流的状态。

例如，每次客户机尝试引导时，显示的服务器输入包计数 (Ipkts) 都会增加，而输出包计数 (Opkts) 保持不变。这种情况表示服务器正在查看来自客户机的引导请求包。但是，服务器却不知道对它们做出响应。这种混乱可能是由 hosts、ipnodes 或 ethers 数据库中的错误地址引起的。

但是，如果输入包计数在一段时间内保持不变，则说明计算机根本未查看包。这种情况说明出现了其他类型的故障，如硬件问题。

```
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      142    0     142    0     0      0
hme0  1500 host58        host58        1106302 0     52419  0     0      0

Name  Mtu  Net/Dest      Address                    Ipkts  Ierrs Opkts  Oerrs Collis
lo0   8252 localhost     localhost                   142    0     142    0     0
hme0  1500 fe80::a00:20ff:feb9:4c54/10 fe80::a00:20ff:feb9:4c54 1106305 0 52422 0  0
```

▼ 如何显示套接字的状态

使用 netstat 命令的 -a 选项，可以查看本地主机上套接字的状态。

- **键入以下内容显示套接字和路由表项的状态：**

使用用户帐户便可运行 netstat 的 -a 选项。

```
% netstat -a
```

示例 8–9 显示所有套接字和路由表项

netstat -a 命令的输出显示详细的统计信息。以下示例显示 netstat -a 典型输出的各部分信息。

```
UDP: IPv4
   Local Address         Remote Address     State
-------------------- -------------------- -------
      *.bootpc                            Idle
host85.bootpc                             Idle
      *.*                                 Unbound
      *.*                                 Unbound
      *.sunrpc                            Idle
      *.*                                 Unbound
      *.32771                             Idle
      *.sunrpc                            Idle
      *.*                                 Unbound
```

```
      *.32775                            Idle
      *.time                             Idle
       .
       .
      *.daytime                          Idle
      *.echo                             Idle
      *.discard                          Idle

UDP: IPv6
   Local Address                 Remote Address                  State      If
-------------------------------- -------------------------------- ---------- -----
      *.*                                                        Unbound
      *.*                                                        Unbound
      *.sunrpc                                                   Idle
      *.*                                                        Unbound
      *.32771                                                    Idle
      *.32778                                                    Idle
      *.syslog                                                   Idle
      .
      .
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*                0      0 49152      0 IDLE
localhost.4999             *.*                0      0 49152      0 LISTEN
      *.sunrpc             *.*                0      0 49152      0 LISTEN
      *.*                  *.*                0      0 49152      0 IDLE
      *.sunrpc             *.*                0      0 49152      0 LISTEN
      .
      .
      *.printer            *.*                0      0 49152      0 LISTEN
      *.time               *.*                0      0 49152      0 LISTEN
      *.daytime            *.*                0      0 49152      0 LISTEN
      *.echo               *.*                0      0 49152      0 LISTEN
      *.discard            *.*                0      0 49152      0 LISTEN
      *.chargen            *.*                0      0 49152      0 LISTEN
      *.shell              *.*                0      0 49152      0 LISTEN
      *.shell              *.*                0      0 49152      0 LISTEN
      *.kshell             *.*                0      0 49152      0 LISTEN
      *.login
       .
       .
           *.*                0      0 49152      0 LISTEN
   *TCP: IPv6
 Local Address            Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------------- ---------------------- ----- ------ ----- ------    ----
   *.*                      *.*                  0      0 49152      0      IDLE
   *.sunrpc                 *.*                  0      0 49152      0      LISTEN
   *.*                      *.*                  0      0 49152      0      IDLE
   *.32774                  *.*                  0      0 49152
```

▼ 如何显示特定地址类型的包的传输状态

使用 netstat 命令的 -f 选项可查看与特定地址族的包传输相关的统计信息。

- **查看 IPv4 或 IPv6 包传输的统计信息。**

  ```
  $ netstat -f inet | inet6
  ```

  要查看 IPv4 传输信息，请键入 inet 作为 netstat -f 的参数。使用 inet6 作为 netstat -f 的参数可查看 IPv6 信息。

示例 8–10 IPv4 包传输的状态

以下示例显示了 netstat -f inet 命令的输出。

```
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
host58.734         host19.nfsd       49640      0 49640      0 ESTABLISHED
host58.38063       host19.32782      49640      0 49640      0 CLOSE_WAIT
host58.38146       host41.43601      49640      0 49640      0 ESTABLISHED
host58.996         remote-host.login 49640      0 49206      0 ESTABLISHED
```

示例 8–11 IPv6 包传输的状态

以下示例显示了 netstat - f inet6 命令的输出。

```
TCP: IPv6
 Local Address          Remote Address        Swind Send-Q Rwind Recv-Q   State    If
------------------ ------------------------- ----- ------ ----- ------ --------- -----
localhost.38065         localhost.32792       49152   0 49152      0    ESTABLISHED
localhost.32792         localhost.38065       49152   0 49152      0    ESTABLISHED
localhost.38089         localhost.38057       49152   0 49152      0    ESTABLISHED
```

▼ 如何显示已知路由的状态

netstat 命令的 -r 选项显示本地主机的路由表。该表显示主机知晓的所有路由的状态。使用用户帐户便可运行 netstat 的 -r 选项。

- **显示 IP 路由表。**

  ```
  $ netstat -r
  ```

示例 8–12 netstat 命令生成的路由表输出

以下示例显示了 netstat -r 命令的输出。

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
```

```
host15              myhost              U         1  31059  hme0
10.0.0.14           myhost              U         1      0  hme0
default             distantrouter       UG        1      2  hme0
localhost           localhost           UH        42019361  lo0

Routing Table: IPv6
  Destination/Mask            Gateway                   Flags Ref   Use   If
--------------------------- --------------------------- ----- --- ------ -----
2002:0a00:3010:2::/64    2002:0a00:3010:2:1b2b:3c4c:5e6e:abcd U  1      0 hme0:1
fe80::/10                fe80::1a2b:3c4d:5e6f:12a2    U       1     23 hme0
ff00::/8                 fe80::1a2b:3c4d:5e6f:12a2    U       1      0 hme0
default                  fe80::1a2b:3c4d:5e6f:12a2    UG      1      0 hme0
localhost                localhost                   UH      9  21832 lo0
```

下表解释了 `netstat -r` 命令的屏幕输出的各种参数。

参数	说明
`Destination` `Destination/Mask`	指定作为路由目标端点的主机。请注意，IPv6 路由表将 6to4 隧道端点 (`2002:0a00:3010:2::/64`) 的前缀显示为路由目标端点。
`Gateway`	指定用于转发包的网关。
`Flags`	指示路由的当前状态。`U` 标志指示路由处于运行状态。`G` 标志指示路由指向网关。
`Use`	显示已发送的包数。
`Interface`	指示作为传输源端点的本地主机上的特定接口。

使用 ping 命令探测远程主机

可以使用 `ping` 命令确定远程主机的状态。运行 `ping` 时，ICMP 协议会将数据报发送到指定的主机，并请求响应。ICMP 是负责 TCP/IP 网络中错误处理的协议。使用 `ping`，可查明是否存在与指定的远程主机的 IP 连接。

以下是 `ping` 的基本语法：

`/usr/sbin/ping` *host [timeout]*

在此语法中，*host* 是远程主机的名称。*timeout* 参数（可选）指示 `ping` 命令继续尝试到达远程主机所用的时间（以秒为单位）。缺省值为 20 秒。有关其他语法和选项，请参阅 ping(1M) 手册页。

▼ 如何确定远程主机是否正在运行

- **键入以下形式的 `ping` 命令：**

  `$` **`ping`** *hostname*

如果主机 *hostname* 正在接受 ICMP 传输，则会显示以下消息：

hostname is alive

此消息指示 *hostname* 对 ICMP 请求做出了响应。但是，如果 *hostname* 出现故障或者无法接收 ICMP 包，则会从 ping 命令接收到以下响应：

no answer from *hostname*

▼ 如何确定主机是否正在丢弃包

使用 -ping 命令的 s 选项可确定远程主机是否虽在运行但丢失了包。

- **键入以下形式的 ping 命令：**

  $ **ping -s** *hostname*

示例 8–13 用于检测包丢弃的 ping 输出

ping -s *hostname* 命令连续不断地将包发送到指定的主机，直到您发送中断字符或出现超时为止。屏幕上显示的响应信息与以下内容类似：

```
& ping -s host1.domain8
PING host1.domain8 : 56 data bytes
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=0. time=1.67 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=1. time=1.02 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=2. time=0.986 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=3. time=0.921 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=4. time=1.16 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.00 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.980 ms

^C

----host1.domain8  PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 0.921/1.11/1.67/0.26
```

包丢失统计信息指示主机是否已丢弃包。如果 ping 失败，请检查由 ifconfig 和 netstat 命令报告的网络状态。请参阅第 178 页中的 “使用 ifconfig 命令监视接口配置”和第 182 页中的 “使用 netstat 命令监视网络状态”。

管理和记录网络状态显示

以下任务说明如何使用已知的网络命令来检查网络状态。

▼ 如何控制与 IP 相关的命令的显示输出

可以将 `netstat` 和 `ifconfig` 命令的输出控制为仅显示 IPv4 信息或同时显示 IPv4 和 IPv6 信息。

1 **创建 `/etc/default/inet_type` 文件。**

2 **根据您的网络需要，将以下某一项添加到 `/etc/default/inet_type`：**

- 仅显示 IPv4 信息：

  ```
  DEFAULT_IP=IP_VERSION4
  ```

- 同时显示 IPv4 和 IPv6 信息：

  ```
  DEFAULT_IP=BOTH
  ```

  或

  ```
  DEFAULT_IP=IP_VERSION6
  ```

  有关 `inet_type` 文件的更多信息，请参见 `inet_type(4)` 手册页。

注 – `ifconfig` 命令中的 `-4` 和 `-6` 标志将覆盖 `inet_type` 文件中设置的值。`netstat` 命令中的 `-f` 标志也将覆盖 `inet_type` 文件中设置的值。

示例 8–14 将输出控制为有选择地显示 IPv4 和 IPv6 信息

- 在 `inet_type` 文件中指定 `DEFAULT_IP=BOTH` 或 `DEFAULT_IP=IP_VERSION6` 变量时，应该显示以下输出：

```
% ifconfig -a
lo0: flags=1000849 mtu 8232 index 1
        inet 10.10.0.1 netmask ff000000
qfe0: flags=1000843 mtu 1500 index 2
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.86.255
        ether 8:0:20:56:a8
lo0: flags=2000849 mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 mtu 1500 index 2
        ether 8:0:20:56:a8
        inet6 fe80::a00:fe73:56a8/10
qfe0:1: flags=2080841 mtu 1500 index 2
        inet6 2001:db8:3c4d:5:a00:fe73:56a8/64
```

- 当您在 `inet_type` 文件中指定 `DEFAULT_IP=IP_VERSION4` 变量时，应该显示以下输出：

```
% ifconfig -a
lo0: flags=849 mtu 8232
        inet 10.10.0.1 netmask ff000000
qfe0: flags=843 mtu 1500
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.86.255
        ether 8:0:20:56:a8
```

▼ 如何记录 IPv4 路由选择守护进程的操作

如果怀疑 `routed`（IPv4 路由选择守护进程）不能正常运行，则可以启动跟踪此守护进程活动的日志。此日志包括启动 `routed` 守护进程时的所有包传送。

1 **在本地主机上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **创建路由选择守护进程操作的日志文件：**

```
# /usr/sbin/in.routed /var/log-file-name
```

注意 – 在繁忙的网络中，此命令生成的输出几乎是连续的。

示例 8–15 `in.routed` 守护进程的网络日志

以下示例显示由第 191 页中的 “如何记录 IPv4 路由选择守护进程的操作”过程创建的日志的开始部分。

```
-- 2003/11/18 16:47:00.000000 --
Tracing actions started
RCVBUF=61440
Add interface lo0  #1   127.0.0.1      -->127.0.0.1/32
   <UP|LOOPBACK|RUNNING|MULTICAST|IPv4> <PASSIVE>
Add interface hme0 #2   10.10.48.112    -->10.10.48.0/25
    <UP|BROADCAST|RUNNING|MULTICAST|IPv4>
turn on RIP
Add    10.0.0.0        -->10.10.48.112      metric=0  hme0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  hme0  <IF|NOPROP>
```

▼ 如何跟踪 IPv6 相邻节点搜索守护进程的活动

如果您怀疑 IPv6 `in.ndpd` 守护进程不能正常运行，则可以启动跟踪此守护进程的活动的日志。此跟踪显示在标准输出中，直到终止。此跟踪包括启动 `in.ndpd` 守护进程时的所有包传送。

1. **在本地 IPv6 节点上承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **启动对 in.ndpd 守护进程的跟踪。**

   ```
   # /usr/lib/inet/in.ndpd -t
   ```

3. **根据需要按 Ctrl-C 组合键终止跟踪。**

示例 8–16 对 in.ndpd 守护进程的跟踪

以下输出显示了对 in.ndpd 的跟踪的开始部分。

```
# /usr/lib/inet/in.ndpd -t
Nov 18 17:27:28 Sending solicitation to  ff02::2 (16 bytes) on hme0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:b9:4c:54>
Nov 18 17:27:28 Received valid advert from fe80::a00:20ff:fee9:2d27 (88 bytes) on hme0
Nov 18 17:27:28         Max hop limit: 0
Nov 18 17:27:28         Managed address configuration: Not set
Nov 18 17:27:28         Other configuration flag: Not set
Nov 18 17:27:28         Router lifetime: 1800
Nov 18 17:27:28         Reachable timer: 0
Nov 18 17:27:28         Reachable retrans timer: 0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:e9:2d:27>
Nov 18 17:27:28         Prefix: 2001:08db:3c4d:1::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
Nov 18 17:27:28         Prefix: 2002:0a00:3010:2::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
```

使用 traceroute 命令显示路由信息

traceroute 命令将跟踪发往远程系统的 IP 包所经过的路由。有关 traceroute 的详细技术信息，请参见 traceroute(1M) 手册页。

可以使用 traceroute 命令查找所有的路由配置错误以及路由路径错误。如果无法到达特定的主机，则可以使用 traceroute 来查看发往远程主机的包所经由的路径以及可能出现故障的位置。

traceroute 命令还显示在通向目标主机的路径上每个网关的往返时间。此信息对于分析两个主机之间何处出现通信缓慢非常有用。

▼ 如何查找通向远程主机的路由

- **键入以下命令查找通向远程系统的路由：**

  % **traceroute** *destination-hostname*

  使用用户帐户便可运行此 traceroute 命令形式。

示例 8–17 使用 traceroute 命令显示通向远程主机的路由

以下 traceroute 命令输出显示了包从本地系统 nearhost 到达远程系统 farhost 所经由的具有七个跃点的路径。此输出还显示包遍历每个跃点所用的时间。

```
istanbul% traceroute farhost.faraway.com
    traceroute to farhost.faraway.com (172.16.64.39), 30 hops max, 40 byte packets
     1  frbldg7c-86 (172.16.86.1)  1.516 ms  1.283 ms  1.362 ms
     2  bldg1a-001 (172.16.1.211)  2.277 ms  1.773 ms  2.186 ms
     3  bldg4-bldg1 (172.16.4.42)  1.978 ms  1.986 ms  13.996 ms
     4  bldg6-bldg4 (172.16.4.49)  2.655 ms  3.042 ms  2.344 ms
     5  ferbldg11a-001 (172.16.1.236)  2.636 ms  3.432 ms  3.830 ms
     6  frbldg12b-153 (172.16.153.72)  3.452 ms  3.146 ms  2.962 ms
     7  sanfrancisco (172.16.64.39)  3.430 ms  3.312 ms  3.451 ms
```

▼ 如何跟踪所有路由

此过程使用 traceroute 命令的 -a 选项来跟踪所有路由。

- **在本地系统上键入以下命令：**

  % **traceroute -a***host-name*

  使用用户帐户便可运行此 traceroute 命令形式。

示例 8–18 跟踪所有通向双栈主机的路由

此示例显示通向双栈主机的所有可能路由。

```
% traceroute -a v6host.remote.com
traceroute: Warning: Multiple interfaces found; using 2::56:a0:a8 @ eri0:2
traceroute to v6host (2001:db8:4a3b::102:a00:fe79:19b0),30 hops max, 60 byte packets
 1  v6-rout86 (2001:db8:4a3b:56:a00:fe1f:59a1)  35.534 ms  56.998 ms *
 2  2001:db8::255:0:c0a8:717  32.659 ms  39.444 ms *
 3  farhost.faraway.COM (2001:db8:4a3b::103:a00:fe9a:ce7b)  401.518 ms  7.143 ms *
 4  distant.remote.com (2001:db8:4a3b::100:a00:fe7c:cf35)  113.034 ms  7.949 ms *
 5  v6host (2001:db8:4a3b::102:a00:fe79:19b0)  66.111 ms *  36.965 ms

traceroute to v6host.remote.com  (192.168.10.75),30 hops max,40 byte packets
 1  v6-rout86 (172.16.86.1)  4.360 ms  3.452 ms  3.479 ms
 2  flrmpj17u.here.COM (172.16.17.131)  4.062 ms  3.848 ms  3.505 ms
 3  farhost.farway.com (10.0.0.23)  4.773 ms *  4.294 ms
 4  distant.remote.com (192.168.10.104)  5.128 ms  5.362 ms *
 5  v6host  (192.168.15.85)  7.298 ms  5.444 ms *
```

使用 snoop 命令监视包传送

可以使用 snoop 命令监视数据传送的状态。snoop 捕获网络包并以指定的格式显示其内容。系统收到包或将其保存到文件之后，便会立即显示这些包。当 snoop 向中间文件执行写入操作时，在密切跟踪的情况下不可能丢失包。然后，可以使用 snoop 本身来解释此文件。

要以混杂模式捕获进出缺省接口的包，您必须承担网络管理员角色或成为超级用户。在汇总表单中，snoop 仅显示与最高级协议有关的数据。例如，NFS 包仅显示 NFS 信息，而不会显示底层 RPC、UDP、IP 和以太网帧信息，但是如果选择了两个详细选项之一，则会显示这些信息。

坚持不懈地使用 snoop 可以使您熟悉常规系统行为。有关对包进行分析的帮助，请查找最新的白皮书和 RFC，并搜寻专家针对特定领域（如 NFS 或 NIS）提供的建议。有关使用 snoop 及其选项的详细信息，请参阅 snoop(1M) 手册页。

▼ 如何检查来自所有接口的包

1. **在本地主机上，承担网络管理角色或成为超级用户。**

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **列显有关连接到系统的接口的信息。**

   ```
   # ifconfig -a
   ```

   snoop 命令通常使用第一个非回送设备，通常为主网络接口。

3. **键入不带参数的 snoop 开始捕获包，如示例 8–19 所示。**

4. **使用 Ctrl-C 组合键停止此进程。**

示例 8–19 snoop 命令的输出

基本 snoop 命令针对双栈主机返回如下所示的输出。

```
% snoop
Using device /dev/hme (promiscuous mode)
router5.local.com -> router5.local.com ARP R 10.0.0.13, router5.local.com is
    0:10:7b:31:37:80
router5.local.com -> BROADCAST     TFTP Read "network-confg" (octet)
farhost.remote.com -> myhost       RLOGIN C port=993
myhost ->   nisserve2          NIS C MATCH 10.0.0.64 in ipnodes.byaddr
nisserve2 ->    myhost             NIS R MATCH No such key
blue-112 -> slave-253-2        NIS C MATCH 10.0.0.112 in ipnodes.byaddr
myhost -> DNSserver.local.com      DNS C 192.168.10.10.in-addr.arpa. Internet PTR ?
DNSserver.local.com  myhost        DNS R 192.168.10.10.in-addr.arpa. Internet PTR
```

```
   niserve2.
.
.
.
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (5 destinations)
```

在此输出中捕获的包显示了远程登录部分，包括查找 NIS 和 DNS 服务器以便进行地址解析。同时还包括来自本地路由器的定期 ARP 包以及向 `in.ripngd` 发出的 IPv6 链路本地地址的通告。

▼ 如何将 snoop 输出捕获到文件

1 **在本地主机上，承担网络管理角色或成为超级用户。**

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **将 snoop 会话捕获到文件。**

```
# snoop -o filename
```

例如：

```
# snoop -o /tmp/cap
Using device /dev/eri (promiscuous mode)
30 snoop: 30 packets captured
```

此示例中，在名为 `/tmp/cap` 的文件中捕获到了 30 个包。可以将此文件放在任何具有足够磁盘空间的目录中。捕获的包数显示在命令行中，您可以随时按 Ctrl-C 组合键中止捕获。

`snoop` 将在主机上生成大量网络负载，这会使结果失真。要查看实际结果，请从第三方系统运行 `snoop`。

3 **检查 snoop 输出捕获文件。**

```
# snoop -i filename
```

示例 8–20 snoop 输出捕获文件的内容

以下内容显示了可能会作为 `snoop -i` 命令输出接收到的各种捕获。

```
# snoop -i /tmp/cap
1   0.00000 fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375
    ICMPv6 Neighbor advertisement
...
10  0.91493    10.0.0.40 -> (broadcast)  ARP C Who is 10.0.0.40, 10.0.0.40 ?
34  0.43690 nearserver.here.com  -> 224.0.1.1  IP  D=224.0.1.1 S=10.0.0.40 LEN=28,
      ID=47453, TO =0x0, TTL=1
35  0.00034  10.0.0.40 -> 224.0.1.1    IP  D=224.0.1.1 S=10.0.0.40 LEN=28, ID=57376,
     TOS=0x0, TTL=47
```

▼ 如何检查 IPv4 服务器和客户机之间的包

1. **在远离与客户机或服务器相连的集线器的位置建立 snoop 系统。**
   第三方系统（snoop 系统）将检查所有干预通信，因此 snoop 跟踪会反映网络上实际出现的情况。

2. **在 snoop 系统上，承担网络管理角色或成为超级用户。**
   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

3. **键入带有选项的 snoop 并将输出保存到文件。**

4. **检查并解释输出。**
   有关 snoop 捕获文件的详细信息，请参阅《RFC 1761, Snoop Version 2 Packet Capture File Format》（《RFC 1761，Snoop 版本 2 包捕获文件格式》）(`http://www.ietf.org/rfc/rfc1761.txt?number=1761`)。

▼ 如何监视 IPv6 网络通信

您可以使用 snoop 命令来仅显示 IPv6 包。

1. **在本地节点上，承担网络管理角色或成为超级用户。**
   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

2. **捕获 IPv6 包。**

   ```
   # snoop ip6
   ```

   有关 snoop 命令的更多信息，请参见 snoop(1M) 手册页。

示例 8–21 仅显示 IPv6 网络通信

以下示例显示了在节点上运行 snoop ip6 命令时可能显示的典型输出。

```
# snoop ip6
fe80::a00:20ff:fecd:4374 -> ff02::1:ffe9:2d27 ICMPv6 Neighbor solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
      solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
      solicitation
fe80::a00:20ff:febb:e09 -> ff02::9      RIPng R (11 destinations)
fe80::a00:20ff:fee9:2d27 -> ff02::1:ffcd:4375 ICMPv6 Neighbor solicitation
```

管理缺省地址选择

Oracle Solaris 可以让单个接口拥有多个 IP 地址。例如，使用网络多路径 (network multipathing, IPMP) 之类的技术，可以将多个网络接口卡 (Network Interface Card, NIC) 连接到同一 IP 链路层。此链路可以具有一个或多个 IP 地址。此外，启用了 IPv6 的系统上的接口具有一个链路本地 IPv6 地址，至少具有一个 IPv6 路由地址，并且至少一个接口具有 IPv4 地址。

当系统启动事务时，应用程序便会对 `getaddrinfo` 套接字发出调用。`getaddrinfo` 将搜索可能在目标系统上使用的地址。然后，内核将设置此列表的优先级，以便找到包的最佳目标。此过程称为**目标地址排序**。然后，如果确定了包的最佳目标地址，Oracle Solaris 内核将选择相应的源地址格式。此过程称为**地址选择**。有关目标地址排序的更多信息，请参见 `getaddrinfo(3SOCKET)` 手册页。

仅启用了 IPv4 的系统和启用了双栈 IPv4/IPv6 的系统必须执行缺省地址选择。大多数情况下，不需要更改缺省地址选择机制。但是，您可能需要更改地址格式的优先级，以便支持 IPMP 或首选使用 6to4 地址格式等。

▼ 如何管理 IPv6 地址选择策略表

以下过程介绍如何修改地址选择策略表。有关 IPv6 缺省地址选择的概念性信息，请参阅第 236 页中的 “ipaddrsel 命令”。

注意 – 如果不是出于下一个任务中提到的某些原因，请不要更改 IPv6 地址选择策略表。策略表构造不正确可能会导致网络出现问题。请确保保存了策略表的副本（如下一过程所示）。

1. **承担主管理员角色，或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **查看当前的 IPv6 地址选择策略表。**

```
# ipaddrsel
# Prefix                  Precedence Label
::1/128                           50 Loopback
::/0                              40 Default
2002::/16                         30 6to4
::/96                             20 IPv4_Compatible
::ffff:0.0.0.0/96                 10 IPv4
```

3. **备份缺省地址策略表的副本。**

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

4 **使用文本编辑器在 /etc/inet/ipaddrsel.conf 中添加定制信息。**

针对 /etc/inet/ipaddrsel 中的各项使用以下语法：

prefix/prefix-length precedence label [# comment]

下面是一些常见的对策略表的修改：

- 为 6to4 地址指定最高优先级。

```
2002::/16                       50 6to4
::1/128                         45 Loopback
```

6to4 地址格式现在具有最高优先级 50，而先前优先级为 50 的回送现在的优先级变为 45。其他地址格式保持不变。

- 指定与特定目标地址进行通信的特定源地址。

```
::1/128                         50 Loopback
2001:1111:1111::1/128           40 ClientNet
2001:2222:2222::/48             40 ClientNet
::/0                            40 Default
```

对于仅有一个物理接口的主机，此特定项非常有用。此处，2001:1111:1111::1/128 是发往网络 2001:2222:2222::/48 中目标的所有包的首选源地址。优先级 40 使得源地址 2001:1111:1111::1/128 的优先级高于为接口配置的其他地址格式。

- IPv4 地址优先于 IPv6 地址。

```
::ffff:0.0.0.0/96               60 IPv4
::1/128                         50 Loopback
.
.
```

IPv4 格式 ::ffff:0.0.0.0/96 的优先级已从缺省的 10 更改为 60，这是表中的最高优先级。

5 **将已修改的策略表加载到内核。**

```
ipaddrsel -f /etc/inet/ipaddrsel.conf
```

6 **如果已修改的策略表存在问题，请恢复缺省 IPv6 地址选择策略表。**

```
# ipaddrsel -d
```

▼ 如何仅修改当前会话的 IPv6 地址选择表

编辑 /etc/inet/ipaddrsel.conf 文件时，所做的任何修改即使在重新引导系统之后也都会保留下来。如果希望已修改的策略表仅存在于当前会话中，请执行以下过程。

1 **承担主管理员角色，或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 将 **/etc/inet/ipaddrsel** 的内容复制到 *filename*，其中 *filename* 是您选择的文件名称。

```
# cp /etc/inet/ipaddrsel filename
```

3 根据需要在 *filename* 中编辑策略表。

4 将已修改的策略表加载到内核。

```
# ipaddrsel -f filename
```

内核将使用新的策略表，直到重新引导系统。

◆ ◆ ◆ 第 9 章

9 对网络问题进行故障排除（任务）

本章包含网络上可能出现的常见问题的解决方案。本章包含以下主题：

对网络问题进行故障排除方面的新增功能

在 Solaris 10 7/07 中，`/etc/inet/ipnodes` 文件已过时。只能对早期 Solaris 10 发行版使用 `/etc/inet/ipnodes`，如以下各个过程中所述。

一般性网络问题解决技巧

网络出现问题的首要症状之一就是一个或多个主机失去通信。如果在首次将某个主机添加到网络中时，该主机根本就未出现，则问题可能出在某个配置文件上，或者网络接口卡出现故障。如果单个主机突然遇到问题，则问题可能出在网络接口上。如果网络上的主机能够互相通信，但是不能与其他网络通信，则问题可能出在路由器上，也可能是另一个网络出现问题。

可以使用 `ifconfig` 命令获取有关网络接口的信息。可以使用 `netstat` 命令显示路由表和协议统计信息。第三方网络诊断程序提供了许多故障排除工具。有关信息，请参阅第三方文档。

导致网络性能下降的原因可能不是很显而易见。例如，可以使用诸如 `ping` 之类的工具来量化主机上包丢失等问题。

运行基本的诊断检查

如果网络存在问题，可以运行一系列软件检查来诊断和修复与软件相关的基本问题。

▼ 如何执行基本的网络软件检查

1. **在本地系统上，承担网络管理角色或成为超级用户。**

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **使用 `netstat` 命令显示网络信息。**

   有关 `netstat` 命令的语法和信息，请参阅第 182 页中的 “使用 netstat 命令监视网络状态”和 `netstat(1M)` 手册页。

3. **检查 `hosts` 数据库（在 Solaris 10 11/06 及早期发行版中，如果您使用的是 IPv6，则需要检查 `ipnodes` 数据库）确保各项正确且为最新。**

   有关 `/etc/inet/hosts` 数据库的信息，请参阅第 207 页中的 “hosts 数据库”和 `hosts(4)` 手册页。有关 `/etc/inet/ipnodes` 数据库的信息，请参阅第 210 页中的 “ipnodes 数据库”和 `ipnodes(4)` 手册页。

4. **如果正在运行反向地址解析协议 (Reverse Address Resolution Protocol, RARP)，请检查 `ethers` 数据库中的以太网地址，确保各项正确且为最新。**

5. **尝试使用 `telnet` 命令连接到本地主机。**

   有关 `telnet` 的语法和信息，请参阅 `telnet(1)` 手册页。

6. **确保网络守护进程 `inetd` 正在运行。**

   ```
   # ps -ef | grep inetd
   ```

   下面的输出可证实 `inetd` 守护进程正在运行：

   ```
   root 57 1 0 Apr 04 ? 3:19 /usr/sbin/inetd -s
   ```

7. **如果在网络上启用了 IPv6，请验证 IPv6 守护进程 `in.ndpd` 是否正在运行：**

   ```
   # ps -ef | grep in.ndpd
   ```

   下面的输出可证实 `in.ndpd` 守护进程正在运行：

   ```
   root 123  1 0  Oct 27 ?  0:03 /usr/lib/inet/in.ndpd
   ```

部署 IPv6 时的常见问题

本节介绍在站点上规划和部署 IPv6 时可能遇到的疑问和问题。有关实际的规划任务，请参阅第 4 章，规划 IPv6 网络（任务）。

IPv4 路由器无法升级到 IPv6

如果现有设备无法升级，则可能必须购买支持 IPv6 的设备。有关为支持 IPv6 而必须执行的特定于设备的任何过程，请查阅制造商的文档。

某些 IPv4 路由器无法进行升级以支持 IPv6。如果您的拓扑属于这种情况，请紧邻该 IPv4 路由器物理连接一个 IPv6 路由器。然后，可以从 IPv6 路由器建立经由 IPv4 路由器的隧道。有关配置隧道的任务，请参阅第 164 页中的 “针对 IPv6 支持配置隧道所需的任务（任务列表）”。

将服务升级到 IPv6 之后遇到的问题

在准备服务使其支持 IPv6 时，可能会遇到下列情况：

- 某些应用程序，即使在移植到 IPv6 之后，缺省情况下也不会启用 IPv6 支持。可能必须配置这些应用程序以启用 IPv6。
- 运行多个服务（其中的一些服务仅使用 IPv4，而其他服务既使用 IPv4 又使用 IPv6）的服务器可能会遇到问题。某些客户机可能需要同时使用这两种类型的服务，这会在服务器端导致混乱。

当前的 ISP 不支持 IPv6

如果想要部署 IPv6，但是当前的 ISP 不支持 IPv6 寻址，则可考虑通过以下备选方法改用其他 ISP：

- 租用一个 ISP，为您站点的 IPv6 通信提供第二条线路。此解决方案成本较高。
- 获取一个**虚拟** *ISP*。虚拟 ISP 可为您的站点提供 IPv6 连通性，但不提供链路。您需要从您的站点建立一个经由 IPv4 ISP 到达虚拟 ISP 的隧道。
- 使用经由 ISP 到达其他 IPv6 站点的 6to4 隧道。对于地址，请使用 6to4 路由器的已注册 IPv4 地址作为 IPv6 地址的公共拓扑部分。

建立通往 6to4 中继路由器的隧道时的安全问题

本质上，6to4 路由器与 6to4 中继路由器之间的隧道是不安全的。此类隧道存在以下固有安全问题：

- 尽管 6to4 中继路由器确实会对包进行封装和取消封装，但是这些路由器并不检查这些包中所包含的数据。
- 地址欺骗是通往 6to4 中继路由器的隧道中的主要问题。对于传入通信，6to4 路由器无法将中继路由器的 IPv4 地址与源 IPv6 地址匹配。因此，IPv6 主机的地址很容易被欺骗，6to4 中继路由器的地址也可能会被欺骗。

- 缺省情况下，6to4 路由器与 6to4 中继路由器之间不存在信任机制。因此，6to4 路由器无法识别 6to4 中继路由器是否受信任，或者甚至无法识别它是否是合法的 6to4 中继路由器。6to4 站点与 IPv6 目标之间必须存在信任关系，否则这两个站点会很容易受到攻击。

Internet 草案《Security Considerations for 6to4》中对这些问题和 6to4 中继路由器固有的其他安全问题进行了说明。通常，仅出于以下几种原因才考虑启用 6to4 中继路由器支持：

- 6to4 站点尝试与受信任的专用 IPv6 网络通信。例如，可以在由隔离的 6to4 站点和本地 IPv6 站点组成的校园网络上启用 6to4 中继路由器支持。
- 出于迫切的商业需求，6to4 站点需要与某些本地 IPv6 主机通信。
- 已实现了 Internet 草案《Security Considerations for 6to4》中建议的检查和信任模型。

◆ ◆ ◆ 第 10 章

10 TCP/IP 和 IPv4 详解（参考）

本章提供有关网络配置文件的 TCP/IP 网络参考信息，包括文件项的类型、用途和格式，同时还详细介绍了现有的网络数据库。本章还介绍如何基于已定义的网络分类和子网号衍生得到 IPv4 地址结构。

本章包含以下信息：

TCP/IP 和 IPv4 中的新增功能详解

在 Solaris 10 7/07 中，/etc/inet/ipnodes 文件已过时。只能对早期 Solaris 10 发行版使用 /etc/inet/ipnodes，如以下各个过程中所述。

TCP/IP 配置文件

网络中的每个系统都可以从以下 TCP/IP 配置文件和网络数据库中获取其 TCP/IP 配置信息：

- /etc/hostname.*interface* 文件
- /etc/nodename 文件
- /etc/defaultdomain 文件
- /etc/defaultrouter 文件（可选）
- hosts 数据库
- ipnodes 数据库（在 Solaris 10 11/06 及早期发行版中）
- netmasks 数据库（可选）

Oracle Solaris 安装程序在安装过程中创建上述文件。也可以按照本节中的说明手动编辑这些文件。hosts 和 netmasks 数据库是两个网络数据库，可供 Oracle Solaris 网络上可用

的名称服务读取。第 214 页中的 “网络数据库和 nsswitch.conf 文件”详细介绍了网络数据库的概念。对于 Solaris 10 11/06 及早期发行版，有关 ipnodes 文件的信息，请参见第 210 页中的 “ipnodes 数据库”。

/etc/hostname.*interface* 文件

此文件定义了本地主机上的物理网络接口。本地系统上至少应该有一个 /etc/hostname.*interface* 文件。Oracle Solaris 安装程序会为安装过程中找到的第一个接口创建 /etc/hostname.*interface* 文件。此接口通常具有最低的设备编号（例如 eri0），并称为**主网络接口**。如果安装程序找到其他接口，您也可以在安装过程中对这些接口进行配置（可选）。

注 – 如果为同一接口创建备用主机名文件，则备用文件也必须遵循命名格式 hostname.[0-9]*，如 hostname.qfe0.a123。诸如 hostname.qfe0.bak 或 hostname.qfe0.old 之类的名称无效，且会在系统引导期间被脚本忽略。

也请注意，一个给定的接口只能有一个相应的主机名文件。如果用有效的文件名为接口创建一个备用主机名文件，例如 /etc/hostname.qfe 和 /etc/hostname.qfe.a123，则引导脚本会尝试同时引用这两个主机名文件的内容来进行配置，因而会产生错误。要避免这些错误，请为给定配置中不使用的主机名文件使用无效的文件名。

如果在安装之后为系统添加新的网络接口，必须为此接口创建 /etc/hostname.*interface* 文件，如第 129 页中的 “如何在安装系统后配置物理接口”中所述。另外，为使 Oracle Solaris 软件识别并使用新的网络接口，需要将此接口的设备驱动程序加载到相应的目录中。有关适当的 *interface* 名称和设备驱动程序的说明，请参阅新网络接口附带的文档。

基本的 /etc/hostname.*interface* 文件包含一项内容：与此网络接口关联的主机名或 IPv4 地址。IPv4 地址可用传统的点分十进制格式或 CIDR 表示法表示。如果在 /etc/hostname.*interface* 文件中使用主机名，/etc/inet/hosts 文件也必须包含此主机名。

例如，假定 smc0 是称为 tenere 的系统的主网络接口。/etc/hostname.smc0 文件所包含的项可以是以点分十进制或 CIDR 表示法表示的 IPv4 地址，也可以是主机名 tenere。

注 – IPv6 使用 /etc/hostname6.*interface* 文件定义网络接口。有关更多信息，请参阅第 235 页中的 “IPv6 接口配置文件”。

/etc/nodename 文件

此文件应该包含一项内容，即本地系统的主机名。例如，在 timbuktu 系统上，/etc/nodename 文件将会包含 timbuktu 项。

/etc/defaultdomain 文件

此文件应该包含一项内容，即本地主机网络所属的管理域的全限定域名。可以将此名称提供给 Oracle Solaris 安装程序或在以后编辑此文件。有关网络域的更多信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

/etc/defaultrouter 文件

对于每个直接连接到网络的路由器，此文件可以包含一个对应项。该项应该是作为网络间路由器的网络接口的名称。/etc/defaultrouter 文件的存在表明系统是支持静态路由的。

hosts 数据库

hosts 数据库包含网络中各系统的 IPv4 地址和主机名。如果使用 NIS 或 DNS 名称服务，或者使用 LDAP 目录服务，则 hosts 数据库在专门存储主机信息的数据库中进行维护。例如，在运行 NIS 的网络中，hosts 数据库在 hostsbyname 文件中进行维护。

如果使用本地文件提供名称服务，则 hosts 数据库将在 /etc/inet/hosts 文件中进行维护。此文件包含主网络接口的主机名和 IPv4 地址、连接到系统的其他网络接口的主机名和 IPv4 地址以及系统必须检查的其他网络地址。

注 – 为了与基于 BSD 的操作系统兼容，/etc/hosts 文件是指向 /etc/inet/hosts 的符号链接。

/etc/inet/hosts 文件格式

/etc/inet/hosts 文件使用以下基本语法。有关完整的语法信息，请参阅 hosts(4) 手册页。

IPv4-address hostname [nicknames] [#comment]

IPv4-address	包含本地主机必须识别的每个接口的 IPv4 地址。
hostname	包含设置期间指定给系统的主机名，以及指定给本地主机必须识别的其他网络接口的主机名。
[nickname]	包含主机别名的可选字段。
[#comment]	可选的注释字段。

初始 /etc/inet/hosts 文件

在系统上运行 Oracle Solaris 安装程序时，该程序将配置初始 /etc/inet/hosts 文件。此文件包含本地主机所需的最少数目的项。其中包括回送地址、主机 IPv4 地址以及主机名。

例如，Oracle Solaris 安装程序可能会为图 5–1 中所示的 tenere 系统创建以下 /etc/inet/hosts 文件：

示例 10–1 系统 tenere 的 /etc/inet/hosts 文件

```
127.0.0.1     localhost         loghost    #loopback address
192.168.200.3   tenere                        #host name
```

回送地址

在示例 10–1中，IPv4 地址 127.0.0.1是**回送地址**。回送地址是本地系统用来允许进程间通信的保留网络接口。主机可使用此地址将数据包发送给自己。ifconfig 命令使用回送地址进行配置和测试，如第 178 页中的 “使用 ifconfig 命令监视接口配置”中所述。TCP/IP 网络中的每个系统都必须使用 IP 地址 127.0.0.1 作为本地主机的 IPv4 回送地址。

主机名

IPv4 地址 192.168.200.1 和名称 tenere 是本地系统的地址和主机名。它们指定给系统的主网络接口。

多个网络接口

一些系统具有多个网络接口，因为它们是路由器或者多宿主主机。每个连接到系统的网络接口都需要有自己的 IP 地址以及与其关联的名称。在安装过程中，必须配置主网络接口。安装时，如果特定系统具有多个接口，Oracle Solaris 安装程序会针对其他接口对您进行提示。此时，您可以选择性地配置一个或多个其他接口，或者以后手动进行配置。

安装 Oracle Solaris 后，可以通过将接口信息添加到系统的 /etc/inet/hosts 文件，来为路由器或多宿主主机配置其他接口。有关配置路由器和多宿主主机的更多信息，请参阅第 105 页中的 “配置 IPv4 路由器”和第 113 页中的 “配置多宿主主机”。

示例 10–2 显示了图 5–1 中所示的系统 timbuktu 的 /etc/inet/hosts 文件。

示例 10–2 系统 timbuktu 的 /etc/inet/hosts 文件

```
127.0.0.1        localhost     loghost
192.168.200.70   timbuktu      #This is the local host name
192.168.201.10   timbuktu-201  #Interface to network 192.9.201
```

通过这两个接口，timbuktu 作为路由器连接网络 192.168.200 和 192.168.201。

名称服务如何影响 hosts 数据库

NIS 和 DNS 名称服务以及 LDAP 目录服务在一台或多台服务器上维护主机名和地址。这些服务器维护 hosts 数据库，该数据库包含服务器所在的网络中每台主机和路由器（如果适用）的信息。有关这些服务的更多信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

本地文件何时提供名称服务

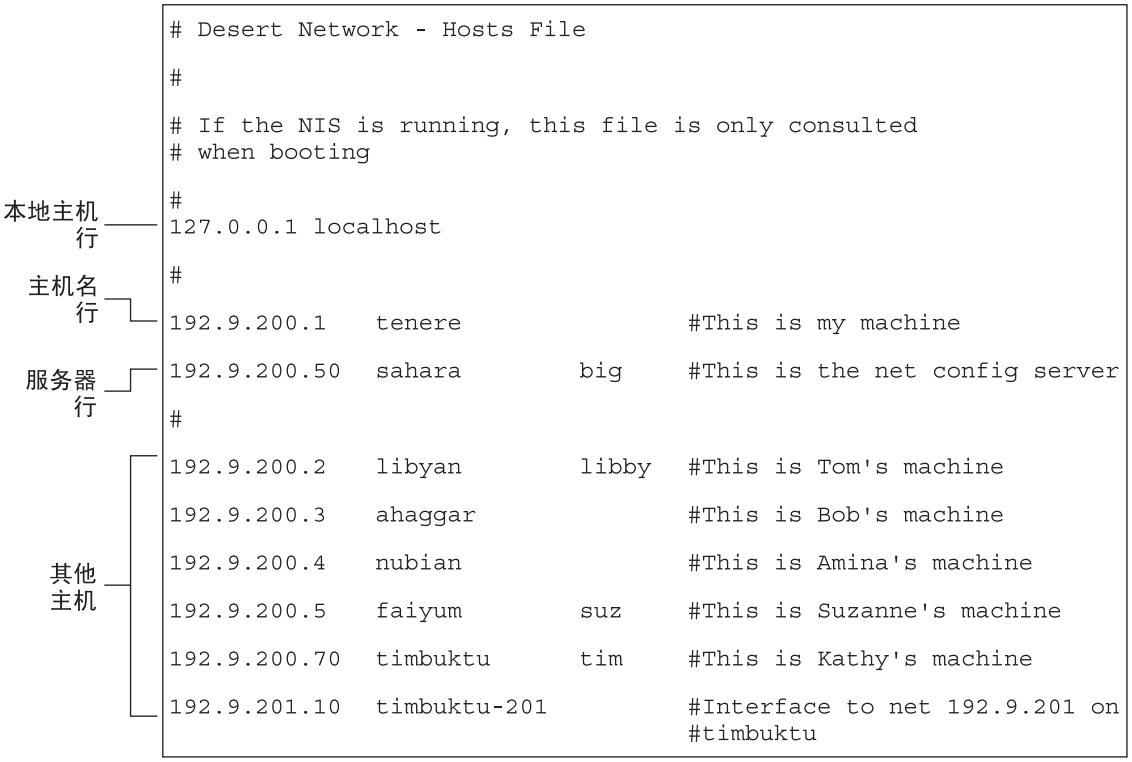
在使用本地文件提供名称服务的网络中，以本地文件模式运行的系统将在各自的 /etc/inet/hosts 文件中查找网络中其他系统的 IPv4 地址和主机名。因此，这些系统的 /etc/inet/hosts 文件必须包含以下内容：

- 回送地址
- 本地系统（主网络接口）的 IPv4 地址和主机名
- 连接到此系统的其他网络接口的 IPv4 地址和主机名（如果适用）
- 本地网络中所有主机的 IPv4 地址和主机名
- 系统必须知晓的任何路由器的 IPv4 地址和主机名（如果适用）
- 您的系统想通过其主机名引用的任何系统的 IPv4 地址。

图 10–1 显示了系统 tenere的 /etc/inet/hosts文件。此系统以本地文件模式运行。请注意，此文件包含 192.9.200 网络中每个系统的 IPv4 地址和主机名。此文件还包含接口名称 timbuktu-201 及其对应的 IPv4 地址。此接口将 192.9.200 网络连接到 192.9.201 网络。

配置为网络客户机的系统对其回送地址和 IPv4 地址使用本地 /etc/inet/hosts 文件。

图 10–1 以本地文件模式运行的系统的 /etc/inet/hosts 文件



ipnodes 数据库

注 – 在 Solaris 10 11/06 之后的发行版中，不再包括 ipnodes 数据库。在这些后续发行版中，ipnodes 的 IPv6 功能迁移到 hosts 数据库中。

/etc/inet/ipnodes 文件既可以存储 IPv4 地址，又可以存储 IPv6 地址。此外，也可以存储以传统的点分十进制或 CIDR 表示法表示的 IPv4 地址。此文件作为将主机名与其 IPv4 和 IPv6 地址进行关联的本地数据库。不要将主机名及其地址存储在静态文件中，例如 /etc/inet/ipnodes。但是，为了进行测试，可以按照在 /etc/inet/hosts 中存储 IPv4 地址的方式在文件中存储 IPv6 地址。ipnodes 文件与 hosts 文件使用相同的格式约定。有关 /etc/inet/hosts 的更多信息，请参阅第 207 页中的 “hosts 数据库”。有关 ipnodes 文件的说明，请参见 ipnodes(4) 手册页。

启用了 IPv6 的应用程序使用 /etc/inet/ipnodes 数据库。只包含 IPv4 地址的现有 /etc/hosts 数据库保持不变，以利于现有应用程序的运行。如果 ipnodes 数据库不存在，则启用了 IPv6 的应用程序使用现有的 hosts 数据库。

注 – 如果需要添加地址，必须将 IPv4 地址同时添加到 hosts 和 ipnodes 文件中，而只需将 IPv6 地址添加到 ipnodes 文件中。

示例 10–3 /etc/inet/ipnodes 文件

必须按主机名对主机名地址进行分组，如以下示例所示。

```
#
# Internet IPv6 host table
# with both IPv4 and IPv6 addresses
#
::1     localhost
2001:db8:3b4c:114:a00:20ff:fe78:f37c   farsite.com farsite farsite-v6
fe80::a00:20ff:fe78:f37c     farsite-11.com farsitell
192.168.85.87                    farsite.com farsite farsite-v4
2001:db8:86c0:32:a00:20ff:fe87:9aba   nearsite.com nearsite nearsite-v6
fe80::a00:20ff:fe87:9aba     nearsite-11.com nearsitell
10.0.0.177                   nearsite.com nearsite nearsite-v4 loghost
```

netmasks 数据库

仅当您在网络中设置了子网划分时，才需要在配置网络时编辑 *netmasks* 数据库。netmasks 数据库由网络及其关联的子网掩码的列表组成。

注 – 创建子网时，每个新网络必须是单独的物理网络。不能在单个物理网络中应用子网划分。

什么是子网划分？

在大型互联网络中，**子网划分**是一种最大程度地利用有限的 32 位 IPv4 地址空间并减小路由表大小的方法。借助地址类，子网划分提供了一种将部分主机地址空间分配给网络地址的方法，从而使您具有更多网络。分配给新网络地址的主机地址空间部分称为**子网号**。

除了更有效地利用 IPv4 地址空间之外，子网划分还具有多种管理方面的优势。随着网络数量的增长，路由过程将变得非常复杂。例如，某个小型组织可能为每个本地网络分配了一个 C 类网络号。随着组织的发展，管理大量不同的网络号可能变得异常复杂。一个更好的解决办法是将少量 B 类网络号分配给组织中的各个主要部门。例如，可以为工程部分配一个 B 类网络，为业务部分配一个 B 类网络，等等。然后，使用通过子网划分获得的附加网络号，将每一个 B 类网络划分为多个附加网络。这种划分方式还可以减少必须在路由器间传送的路由信息量。

为 IPv4 地址创建网络掩码

在进行子网划分的过程中，需要选择一个网络范围的**网络掩码**。网络掩码确定主机地址空间中有多少位以及哪些位表示子网号，有多少位以及哪些位表示主机号。请记住，完整的 IPv4 地址由 32 位组成。其中最多可用 24 位、最少可用 8 位表示主机地址空间，具体取决于地址类。网络掩码是在 `netmasks` 数据库中指定的。

如果计划使用子网，必须在配置 TCP/IP 之前确定网络掩码。如果计划将安装操作系统作为网络配置的一部分，Oracle Solaris 安装程序将要求您提供网络的网络掩码。

32 位 IP 地址由网络部分和主机部分组成，如第 51 页中的“设计 IPv4 寻址方案”中所述。32 位分为 4 个字节。根据网络类的不同，将每个字节分别指定给网络号或主机号。

例如，在 B 类 IPv4 地址中，左边的 2 个字节指定给网络号，而右边的 2 个字节指定给主机号。在 B 类 IPv4 地址 `172.16.10` 中，您可以将右边的 2 个字节指定给主机。

如果要实现子网划分，您需要将指定给主机号的字节中的某些位应用到子网地址。例如，16 位的主机地址空间可为 65,534 台主机提供地址。如果将第三个字节应用到子网地址，第四个字节应用到主机地址，则可以为最多 254 个网络提供地址，其中每个网络中最多具有 254 台主机。

主机地址字节中的哪些位应用到子网地址及主机地址是由**子网掩码**确定的。使用子网掩码从用作子网地址的字节中选择位。尽管网络掩码位必须是连续的，但它们不需要与字节边界对齐。

通过使用按位逻辑 AND 运算符，可以将网络掩码应用到 IPv4 地址。此操作将选出地址的网络号和子网号位置。

可以采用二进制表示法来说明网络掩码。可以使用计算器进行二进制到十进制的转换。以下示例显示了网络掩码的十进制格式和二进制格式。

如果将网络掩码 `255.255.255.0` 应用到 IPv4 地址 `172.16.41.101`，则结果为 IPv4 地址 `172.16.41.0`。

`172.16.41.101` & `255.255.255.0` = `172.16.41.0`

此操作的二进制格式如下所示：

10000001.10010000.00101001.01100101（IPv4 地址）

与

11111111.11111111.11111111.00000000（网络掩码）进行 AND 操作

现在，系统查找网络号 172.16.41，而不是 172.16。如果您的网络号是 172.16.41，则此编号便是系统检查并找到的编号。因为最多可以为 IPv4 地址空间的第三个字节指定 254 个值，所以通过子网划分可以为 254 个网络创建地址空间，之前地址空间只可用于一个网络。

如果只需要为两个附加网络提供地址空间，则可以使用以下子网掩码：

255.255.192.0

此网络掩码将生成以下结果：

11111111.11111111.1100000.00000000

此结果仍然保留 14 位，供主机地址使用。因为所有 0 和 1 都是保留的，所以必须至少为主机号保留 2 位。

/etc/inet/netmasks 文件

如果您的网络运行 NIS 或 LDAP，则提供这些名称服务的服务器将维护 netmasks 数据库。对于使用本地文件提供名称服务的网络，此信息在 /etc/inet/netmasks 文件中维护。

注 – 为了与基于 BSD 的操作系统兼容，/etc/netmasks 文件是指向 /etc/inet/netmasks 的符号链接。

以下示例显示 B 类网络的 /etc/inet/netmasks 文件。

示例 10–4 B 类网络的 /etc/inet/netmasks 文件。

```
# The netmasks file associates Internet Protocol (IPv4) address
# masks with IPv4 network numbers.
#
#     network-number    netmask
#
# Both the network-number and the netmasks are specified in
# “decimal dot” notation, e.g:
#
#        128.32.0.0   255.255.255.0
192.168.0.0  255.255.255.0
```

如果 /etc/netmasks 文件不存在，请使用文本编辑器创建此文件。使用以下语法：

network-number *netmask-number*

有关完整的详细信息，请参阅 netmasks(4) 手册页。

创建网络掩码号时，在 /etc/inet/netmasks 中键入由 ISP 或 Internet 注册机构指定的网络号（不是子网号）和网络掩码号。每个子网掩码应单占一行。

例如：

```
128.78.0.0        255.255.248.0
```

您还可以在 /etc/inet/hosts 文件中键入网络号的符号名称。然后，可以使用这些网络名称代替网络号作为命令参数。

inetd Internet 服务守护进程

inetd 守护进程在系统引导时将启动 Internet 标准服务，并可以在系统运行时重新启动服务。使用服务管理工具 (Service Management Facility, SMF) 可以修改标准 Internet 服务或由 inetd 守护进程启动其他服务。

使用以下 SMF 命令可以管理由 inetd 启动的服务：

svcadm	对服务的管理性操作，例如启用、禁用或重新启动。有关详细信息，请参阅 svcadm(1M) 手册页。
svcs	查询服务状态。有关详细信息，请参阅 svcs(1) 手册页。
inetadm	显示和修改服务的属性。有关详细信息，请参阅 inetadm(1M) 手册页。

inetadm 配置文件中针对特定服务的 proto 字段值指示该服务所基于的传输层协议。如果该服务只适用于 IPv4，则 proto 字段必须指定为 tcp、udp 或 sctp。

- 有关使用 SMF 命令的说明，请参阅《Oracle Solaris 管理：基本管理》中的“SMF 命令行管理实用程序”。
- 有关使用 SMF 命令添加在 SCTP 上运行的服务的任务信息，请参阅第 120 页中的 “如何添加使用 SCTP 协议的服务”。
- 有关添加同时处理 IPv4 和 IPv6 请求的服务的信息，请参阅第 214 页中的 “inetd Internet 服务守护进程”。

网络数据库和 nsswitch.conf 文件

网络数据库是提供配置网络所需信息的文件。网络数据库如下所示：

- hosts
- netmasks
- ethers 数据库
- bootparams
- protocols

- services
- networks

如果网络分为多个子网，则在配置过程中，需要编辑 hosts 数据库和 netmasks 数据库。bootparams 和 ethers 这两个数据库用于将系统配置为网络客户机。其余数据库由操作系统使用并很少需要编辑。

尽管 nsswitch.conf 文件不是网络数据库，但是您需要将此文件与相关的网络数据库一同进行配置。nsswitch.conf 指定要用于特定系统的名称服务：本地文件、NIS、DNS 或 LDAP。

名称服务如何影响网络数据库

网络数据库的格式取决于您为网络选择的名称服务的类型。例如，hosts 数据库至少包含本地系统的主机名和 IPv4 地址以及直接连接到本地系统的所有网络接口的主机名和 IPv4 地址。但是，hosts 数据库也可以包含其他 IPv4 地址和主机名，具体取决于网络中的名称服务类型。

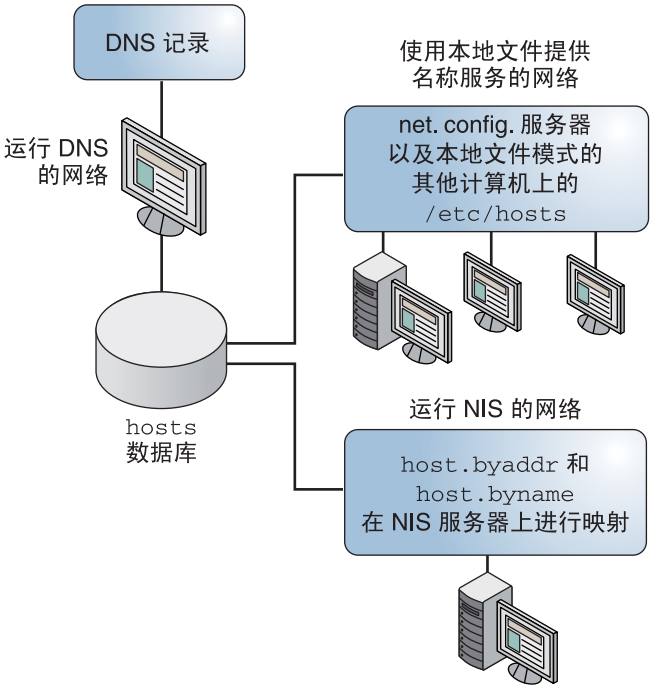
网络数据库按以下方式使用：

- 使用本地文件提供名称服务的网络依赖于 /etc/inet 和 /etc 目录中的文件。
- NIS 使用称为 NIS 映射的数据库。
- DNS 使用带有主机信息的记录。

注 – DNS 引导文件和数据文件不直接对应于网络数据库。

下图显示了这些名称服务使用 hosts 数据库的方式。

图 10–2 名称服务使用 hosts 数据库的方式



下表列出了网络数据库及其对应的本地文件和 NIS 映射。

注 – 在 Solaris 10 11/06 之后的 Oracle Solaris 发行版中，将删除 ipnodes 数据库。

表 10–1 网络数据库和对应的名称服务文件

网络数据库	本地文件	NIS 映射
hosts	/etc/inet/hosts	hosts.byaddr hosts.byname
netmasks	/etc/inet/netmasks	netmasks.byaddr
ethers	/etc/ethers	ethers.byname ethers.byaddr
bootparams	/etc/bootparams	bootparams
protocols	/etc/inet/protocols	protocols.byname protocols.bynumber
services	/etc/inet/services	services.byname
networks	/etc/inet/networks	networks.byaddr networks.byname

本书介绍的是使用本地文件提供名称服务的网络所查看的网络数据库。

- 第 207 页中的 “hosts 数据库”介绍了有关 hosts 数据库的信息。
- 第 211 页中的 “netmasks 数据库”介绍了有关 netmasks 数据库的信息。
- 对于 Solaris 10 11/06 及早期发行版，有关 ipnodes 数据库的信息，请参见第 210 页中的 “ipnodes 数据库”。

有关网络数据库在 NIS、DNS 和 LDAP 中的对应关系的信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

nsswitch.conf 文件

/etc/nsswitch.conf 文件定义了网络数据库的搜索顺序。基于在安装过程中指定的名称服务，Oracle Solaris 安装程序为本地系统创建缺省的 /etc/nsswitch.conf 文件。如果选择了 "None" 选项，表示使用本地文件提供名称服务，生成的 nsswitch.conf 文件与下面的示例类似。

示例 10–5 使用文件提供名称服务的网络的 nsswitch.conf

```
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file contains "switch.so" as a
# nametoaddr library for "inet" transports.

passwd:          files
group:           files
hosts:           files
networks:        files
protocols:       files
rpc:             files
ethers:          files
netmasks:        files
bootparams:      files
publickey:       files
# At present there isn't a 'files' backend for netgroup; the
# system will figure it out pretty quickly,
# and won't use netgroups at all.
netgroup:        files
automount:       files
aliases:         files
services:        files
sendmailvars:    files
```

nsswitch.conf(4) 手册页详细介绍了该文件。基本语法为：

database name-service-to-search

database 字段可以列出操作系统搜索的许多数据库类型之一。例如，此字段可以指示影响用户的数据库（例如 passwd、aliases ）或网络数据库。对于网络数据库，参数

name-service-to-search 的值可以是 files、nis 或 nis+。hosts 数据库也可以将 dns 作为要搜索的名称服务。您也可以列出多个名称服务，例如 nis+ 和 files。

在示例 10–5 中，指示的唯一搜索选项为 files。因此，除了网络数据库信息之外，本地系统还从 /etc 和 /etc/inet 目录中的文件获取安全信息和自动挂载信息。

更改 nsswitch.conf

/etc目录包含由 Oracle Solaris 安装程序创建的 nsswitch.conf文件。此目录还包含以下名称服务的模板文件：

- nsswitch.files
- nsswitch.nis

如果要从一个名称服务更改到另一个名称服务，可以将相应的模板复制到 nsswitch.conf，也可以有选择性地编辑 nsswitch.conf 文件，将缺省名称服务更改为搜索单个数据库。

例如，在运行 NIS 的网络中，可能必须更改网络客户机上的 nsswitch.conf 文件。bootparams 和 ethers 数据库的搜索路径必须将 files 列为第一个选项，然后是 nis。以下示例说明了正确的搜索路径。

示例 10–6 运行 NIS 的网络中客户机的 nsswitch.conf

```
# /etc/nsswitch.conf:#
.
.
passwd:         files nis
group:          files nis

# consult /etc "files" only if nis is down.
hosts:          nis     [NOTFOUND=return] files
networks:       nis     [NOTFOUND=return] files
protocols:      nis     [NOTFOUND=return] files
rpc:            nis     [NOTFOUND=return] files
ethers:         files   [NOTFOUND=return] nis
netmasks:       nis     [NOTFOUND=return] files
bootparams:     files   [NOTFOUND=return] nis
publickey:      nis
netgroup:       nis

automount:      files nis
aliases:        files nis

# for efficient getservbyname() avoid nis
services:       files nis
sendmailvars:   files
```

有关名称服务转换器的完整详细信息，请参阅《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

bootparams 数据库

bootparams 数据库包含配置为以网络客户机模式引导的系统所使用的信息。如果网络中具有网络客户机，则需要编辑此数据库。有关过程，请参见第 95 页中的 “配置网络客户机”。此数据库是根据 /etc/bootparams 文件中输入的信息生成的。

bootparams(4) 手册页介绍了此数据库的完整语法。基本语法为：

system-name file-key-server-name:pathname

对于每个网络客户机系统，该项可能包含以下信息：客户机名称、关键字列表、服务器名称以及路径名。每项的第一个条目都是客户机系统的名称。第一个条目以外的所有其他条目都是可选的。以下是一个示例。

示例 10–7 bootparams 数据库

```
myclient   root=myserver : /nfsroot/myclient  \
swap=myserver : /nfsswap//myclient \
dump=myserver : /nfsdump/myclient
```

在此示例中，dump= 告知客户机主机不要查找转储文件。

bootparams 的通配符项

在大多数情况下，应在编辑 bootparams 数据库时使用通配符项以支持客户机。该项为：

```
*  root=server:/path dump=:
```

星号 (*) 通配符指示该项应用于 bootparams 数据库中所有未特别命名的客户机。

ethers 数据库

ethers 数据库是根据在 /etc/ethers 文件中输入的信息构建的。此数据库将主机名与其**介质访问控制** (Media Access Control, MAC) 地址进行关联。仅当运行 RARP 守护进程时，才需要创建 ethers 数据库。也就是说，如果正在配置网络客户机，则需要创建此数据库。

RARP 使用该文件将 MAC 地址映射到 IP 地址。如果正在运行 RARP 守护进程 in.rarpd，则需要设置 ethers 文件并在运行此守护进程的所有主机上维护此文件，以便将所做的更改应用到网络中。

ethers(4) 手册页介绍了此数据库的完整语法。基本语法为：

MAC-address hostname #comment

MAC-address 主机的 MAC 地址

hostname 主机的正式名称

#comment 要附加到文件中某项的注释

设备制造商提供 MAC 地址。如果在系统引导过程中未显示 MAC 地址，请参见硬件手册中的相关帮助信息。

向 ethers 数据库添加项时，请确保主机名与 hosts（对于 Solaris 10 11/06 及早期发行版，为 ipnodes 数据库）中的主名称而不是别名相对应，如下所示。

示例 10–8 ethers 数据库中的项

```
8:0:20:1:40:16  fayoum
8:0:20:1:40:15  nubian
8:0:20:1:40:7   sahara    # This is a comment
8:0:20:1:40:14  tenere
```

其他网络数据库

很少需要编辑其余的网络数据库。

networks 数据库

networks 数据库将网络名称与网络号相关联，允许某些应用程序使用和显示网络名称而不是网络号。networks 数据库基于 /etc/inet/networks 文件中的信息。此文件包含通过路由器与您网络连接的所有网络的名称。

Oracle Solaris 安装程序配置初始 networks 数据库。但是，如果在现有网络拓扑中添加了新网络，必须更新此数据库。

networks(4) 手册页介绍了 /etc/inet/networks 的完整语法。基本格式为：

network-name network-number nickname(s) #comment

network-name 网络的正式名称

network-number 由 ISP 或 Internet 注册机构指定的编号

nickname 用于识别网络的任何其他名称

#comment 要附加到文件中某项的注释

您必须对 networks 文件进行维护。netstat 程序使用此数据库中的信息生成状态表。

以下是一个 /etc/networks 文件样例。

示例 10–9 /etc/networks 文件

```
#ident    "@(#)networks    1.4    92/07/14 SMI"    /* SVr4.0 1.1    */
#
# The networks file associates Internet Protocol (IP) network
# numbers with network names. The format of this file is:
#
#     network-name              network-number              nicnames . . .

# The loopback network is used only for intra-machine communication
loopback              127

#
# Internet networks
#
arpanet     10       arpa  # Historical
#
# local networks

eng   192.168.9 #engineering
acc   192.168.5 #accounting
prog  192.168.2 #programming
```

protocols 数据库

protocols 数据库列出了在系统上安装的 TCP/IP 协议及其协议编号。Oracle Solaris 安装程序自动创建此数据库。此文件很少需要进行管理。

protocols(4) 手册页介绍了此数据库的语法。以下是一个 /etc/inet/protocols 文件的示例。

示例 10–10 /etc/inet/protocols 文件

```
#
# Internet (IP) protocols
#
ip    0   IP    # internet protocol, pseudo protocol number
icmp  1   ICMP  # internet control message protocol
tcp   6   TCP   # transmission control protocol
udp  17   UDP   # user datagram protocol
```

services 数据库

services 数据库列出了 TCP 和 UDP 服务的名称及其已知的端口号。此数据库由调用网络服务的程序使用。在 Oracle Solaris 安装过程中自动创建 services 数据库。通常，此数据库不需要进行任何管理。

services(4) 手册页介绍了完整的语法信息。以下是从典型的 /etc/inet/services 文件中摘录的内容。

示例 10–11 /etc/inet/services 文件

```
#
# Network services
#
echo      7/udp
echo      7/tcp
echo      7/sctp6
discard   9/udp     sink null
discard   11/tcp
daytime   13/udp
daytime   13/tcp
netstat   15/tcp
ftp-data  20/tcp
ftp       21/tcp
telnet    23/tcp
time      37/tcp    timeserver
time      37/udp    timeserver
name      42/udp    nameserver
whois     43/tcp    nickname
```

Oracle Solaris 中的路由协议

本节介绍了 Oracle Solaris 中支持的两个路由协议：路由信息协议 (Routing Information Protocol, RIP) 和 ICMP 路由器搜索 (Router Discovery, RDISC)。RIP 和 RDISC 都是标准 TCP/IP 协议。有关 Oracle Solaris 中可用的路由协议的完整列表，请参阅表 5–1 和表 5–2。

路由信息协议 (Routing Information Protocol, RIP)

RIP 由系统引导时自动启动的路由选择守护进程 `in.routed` 来实现。如果在指定了 `s` 选项的情况下 `in.routed` 在路由器上运行，它将使用一个可到达每个可访问网络的路由填充内核路由表，并通过所有网络接口通告“可访问性”。

如果在指定了 `q` 选项的情况下 `in.routed` 在主机上运行，它将提取路由信息，但不会通告可访问性。在主机上，可以使用两种方法提取路由信息：

- 不指定 `S` 标志（大写 "S"：“空间节省模式”）。`in.routed` 完全按照它在路由器上的运行方式生成完整的路由表。
- 指定 `S` 标志。`in.routed` 创建一个最小内核表，其中包含每个可用路由器的一个缺省路由。

ICMP 路由器搜索 (Router Discovery, RDISC) 协议

主机使用 RDISC 从路由器获取路由信息。因此，当主机运行 RDISC 时，路由器也必须运行其他协议（例如 RIP）来交换路由器信息。

RDISC 由应该运行在路由器和主机上的 `in.routed` 实现。在主机上，`in.routed` 使用 RDISC 从通过 RDISC 通告自身状态的路由器中搜索缺省路由。在路由器上，`in.routed` 使用 RDISC 将缺省路由通告给直接相连的网络中的主机。请参见 `in.routed(1M)` 手册页和 `gateways(4)` 手册页。

网络类

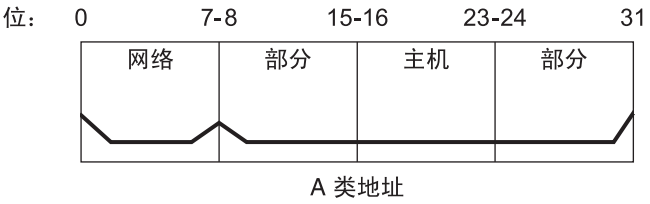
注 – 虽然很多旧网络仍然是基于网络类的，但是 IANA 不再提供基于类的网络号。

本节详细介绍了 IPv4 网络类。每个类以不同的方式使用 32 位 IPv4 地址空间，为网络地址部分提供或多或少的位。这些类是 A 类、B 类和 C 类。

A 类网络号

A 类网络号将 IPv4 地址的前 8 位用作其“网络部分”。其余的 24 位包含 IPv4 地址的主机部分，如下图所示。

图 10–3 A 类地址中的字节分配

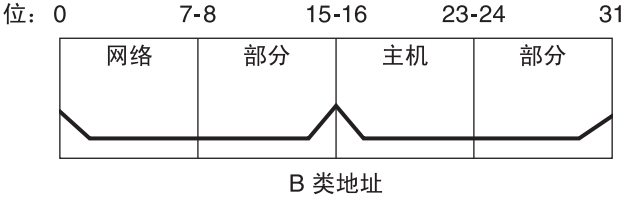


为 A 类网络号的第一个字节指定的值应该介于 0–127 范围内。以 IPv4 地址 `75.4.10.4` 为例。第一个字节中的值 75 指示主机位于 A 类网络中。其余字节 `4.10.4` 构成了主机地址。只有 A 类编号的第一个字节已向 IANA 注册。其余三个字节的使用完全由网络号的所有者决定。只有 127 个 A 类网络。每个 A 类网络都可以接纳最多 16,777,214 台主机。

B 类网络号

B 类网络号将 16 位用于网络号，并将 16 位用于主机号。B 类网络号的第一个字节介于 128–191 的范围内。在编号 172.16.50.56 中，前两个字节 172.16 向 IANA 注册，并构成了网络地址。最后两个字节 50.56 表示主机地址，由网络号的所有者根据自己的判断指定。下图说明了一个 B 类地址。

图 10–4 B 类地址中的字节分配



B 类地址

通常将 B 类指定给网络中包含许多主机的组织。

C 类网络号

C 类网络号将 24 位用于网络号，并将 8 位用于主机号。C 类网络号适用于主机很少的网络－最多 254 台。C 类网络号占用 IPv4 地址的前三个字节。只有第四字节由网络所有者根据自己的判断指定。下图说明了 C 类地址中的字节。

图 10–5 C 类地址中的字节分配

C 类地址

C 类网络号的第一个字节介于 192–223 范围内。第二和第三个字节分别介于 1–255 范围内。192.168.2.5 是一个典型的 C 类地址。前三个字节 192.168.2 构成了网络号。最后一个字节 5 是主机号。

◆ ◆ ◆ 第 11 章

11 IPv6 详解（参考）

本章包含以下有关 Oracle Solaris IPv6 实现的参考信息。

有关 IPv6 的概述，请参阅第 3 章，IPv6 介绍（概述）。有关配置启用 IPv6 的网络的任务，请参阅第 7 章，配置 IPv6 网络（任务）。

IPv6 中的新增功能详解

在 Solaris 10 7/07 中，`/etc/inet/ipnodes` 文件已过时。只能对早期 Solaris 10 发行版使用 `/etc/inet/ipnodes`，如以下各个过程中所述。

IPv6 寻址格式进阶

第 3 章，IPv6 介绍（概述）介绍了最常见的 IPv6 寻址格式：单播站点地址和链路本地地址。本节深入说明了第 3 章，IPv6 介绍（概述）中未详细介绍的寻址格式：

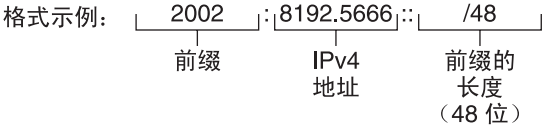
6to4 派生地址

如果计划从路由器或主机端点配置 6to4 隧道，则必须在端点系统上的 `/etc/inet/ndpd.conf` 文件中通告 6to4 站点前缀。有关配置 6to4 隧道的介绍和任务，请参阅第 167 页中的 “如何配置 6to4 隧道”。

下图显示了 6to4 站点前缀的各个部分。

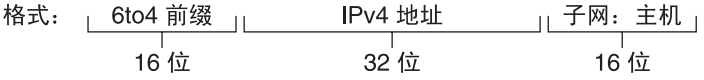
图 11–1 6to4 站点前缀的各个部分

6to4 地址示例： 2002:8192:5666::/48

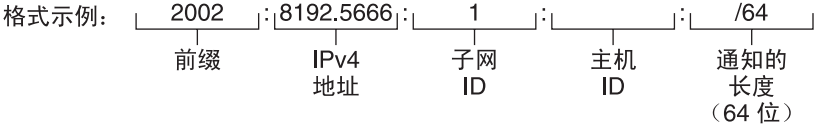


下图显示了 6to4 站点的子网前缀（如 `ndpd.conf` 文件中包含的子网前缀）的各个部分。

图 11–2 6to4 子网前缀的各个部分



6to4 地址示例： 2002:8192.5666:1: :/64



本表解释了 6to4 子网前缀的各个部分、这些部分各自的长度及其定义。

部分	长度	定义
前缀	16 位	6to4 前缀标签 2002 (0x2002)。

部分	长度	定义
IPv4 地址	32 位	已在 6to4 接口上配置的唯一的 IPv4 地址。对于通告，需要指定用十六进制形式表示的 IPv4 地址，而不要指定点分十进制形式的 IPv4 地址。
子网 ID	16 位	子网 ID，对于 6to4 站点上的链路必须唯一。

主机上的 6to4 派生地址

当 IPv6 主机以路由器通告的形式收到 6to4 派生前缀时，它会自动在接口上重新配置 6to4 派生地址。6to4 派生地址具有以下格式：

prefix:IPv4-address:subnet-ID:interface-ID/64

在具有 6to4 接口的主机上执行 `ifconfig -a` 命令所产生的输出类似于以下内容：

```
qfe1:3: flags=2180841<UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6>
 mtu 1500 index 7
        inet6 2002:8192:56bb:9258:a00:20ff:fea9:4521/64
```

在该输出中，`inet6` 后面是 6to4 派生地址。

本表解释了 6to4 派生地址的各个部分、这些部分的长度以及它们提供的信息。

地址部分	长度	定义
prefix	16 位	`2002`，6to4 前缀
IPv4-address	32 位	`8192:56bb`，在 6to4 路由器上配置的 6to4 伪接口的 IPv4 地址（用十六进制形式表示）
子网 *ID*	16 位	`9258`，此主机所属的子网的地址
接口 *ID*	64 位	`a00:20ff:fea9:4521`，为 6to4 配置的主机接口的接口 ID

IPv6 多点传送地址详解

IPv6 多播地址提供了一种将相同的信息或服务分发到一组已定义接口（称为**多播组**）的方法。通常，多播组的接口位于不同的节点上。一个接口可以属于任意数量的多播组。发送到多播地址的包将到达多播组的所有成员。例如，使用多播地址的一种情况就是广播信息，这与 IPv4 广播地址的功能相似。

下表显示了多播地址的格式。

表 11–1 IPv6 多播地址的格式

8 位	4 位	4 位	8 位	8 位	64 位	32 位

表 11–1 IPv6 多播地址的格式 （续）

11111111	*FLGS*	SCOP	*Reserved*	*Plen*	*Network prefix*	*Group ID*

下面是每个字段的内容摘要。

- 11111111－将地址标识为多播地址。
- *FLGS*－设置四个标志 0、0、P、T。前两个标志必须为零。P 字段具有下列值之一：
  - 0 = 不是基于网络前缀指定的多播地址
  - 1 = 基于网络前缀指定的多播地址

  如果 P 设置为 1，则 T 也必须为 1。
- *Reserved*－保留值，为零。
- *Plen*－对于基于站点前缀指定的多播地址，是站点前缀中标识子网的位数。
- *Group ID*－多播组的标识符（不变或动态改变）。

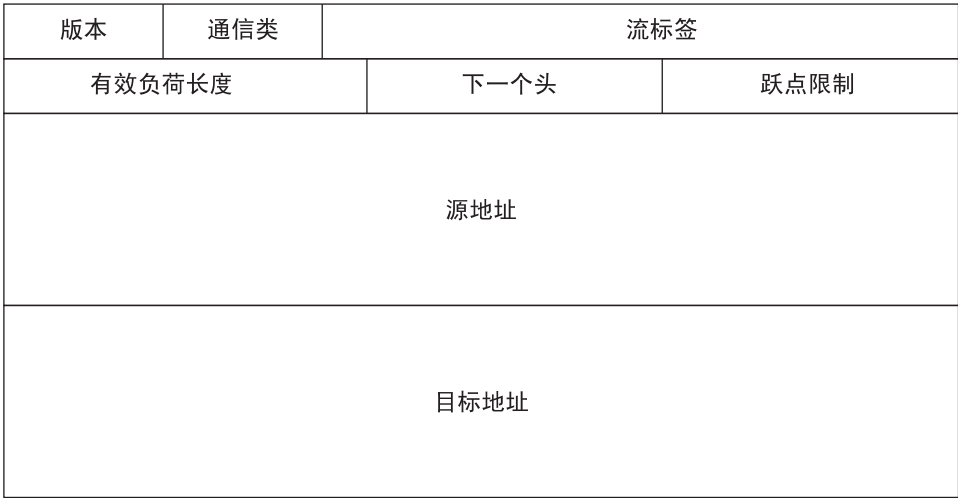
有关多播格式的完整详细信息，请参阅 RFC 3306，《Unicast-Prefix-based IPv6 Multicast Addresses》（《基于单播的前缀 IPv6 多播地址》）(ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt)。

某些 IPv6 多播地址是由 Internet 编号分配机构 (Internet Assigned Numbers Authority, IANA) 永久指定的。所有 IPv6 主机和 IPv6 路由器必需的 "All Nodes Multicast Addresses"（所有节点多播地址）和 "All Routers Multicast Addresses"（所有路由器多播地址）就是这样的地址。IPv6 多播地址也可以由系统动态分配。有关正确使用多播地址和组的更多信息，请参见 RFC 3307，《Allocation Guidelines for IPv6 Multicast Addresses》（《IPv6 多播地址分配指南》）。

IPv6 数据包头的格式

IPv6 协议定义一组数据包头，包括基本 IPv6 数据包头和 IPv6 扩展头。下图显示了 IPv6 数据包头中的字段以及这些字段的顺序。

图 11–3 IPv6 基本数据包头的格式



以下列表介绍了每个头字段的功能。

- **版本**－Internet 协议的 4 位版本号，此处为 6。
- **通信类**－8 位通信类字段。
- **流标签**－20 位字段。
- **有效负荷长度**－16 位无符号整数，这是紧随 IPv6 数据包头之后的其余数据包部分（用八位字节表示）。
- **下一个头**－8 位选定器。标识紧跟在 IPv6 数据包头后面的头的类型。使用与 IPv4 协议字段相同的值。
- **跃点限制**－8 位无符号整数。按转发包的每个节点逐一递减。如果跃点限制递减到零，包就会被丢弃。
- **源地址**－128 位。包初始发送者的地址。
- **目标地址**－128 位。包预定接收者的地址。如果存在可选的路由头，则预定接收者不一定就是接收者。

IPv6 扩展头

IPv6 选项位于包中的 IPv6 数据包头和传输层头之间的单独扩展头中。在包到达其最终目标之前，包传送路径中的任何路由器都不会检查或处理大多数 IPv6 扩展头。此功能显著改进了路由器对于包含选项的包的路由性能。在 IPv4 中，只要存在任何选项，就会要求路由器检查所有的选项。

与 IPv4 选项不同，IPv6 扩展头可以为任意长度。此外，一个包可承载的选项数量也不限于 40 字节。除了 IPv6 选项的处理方式，此功能还允许将 IPv6 选项用于那些在 IPv4 中不可行的功能。

为了在处理后续选项头以及随后的传输协议时提高性能，IPv6 选项始终设置为 8 个八位字节长度的整数倍。8 个八位字节长度的整数倍可以使后续的头保持对齐。

下面是目前已定义的 IPv6 扩展头：

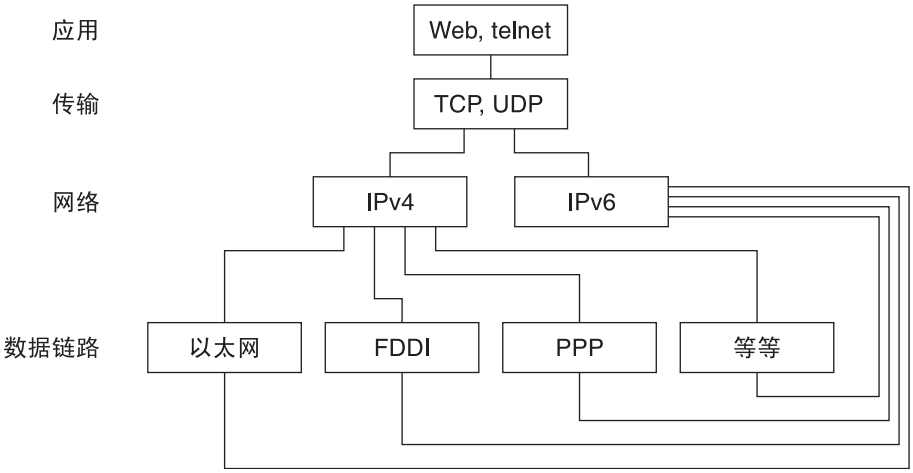
- **路由**－扩展路由，如 IPv4 稀疏源路由
- **分段**－分段和重新装配
- **验证**－完整性、验证和安全性
- **封装安全有效负荷**－保密性
- **逐跳寻径 (Hop-by-Hop) 选项**－需要进行逐跳寻径处理的特殊选项
- **目标选项**－由目标节点检查的可选信息

双栈协议

双栈通常是指协议栈中的所有级别（从应用层到网络层）都是重复的。同时运行 OSI 协议和 TCP/IP 协议的系统就是完全重复的示例。

Oracle Solaris 是**双栈**协议，即 Oracle Solaris 同时实现了 IPv4 和 IPv6 协议。安装操作系统时，可以选择在 IP 层启用 IPv6 协议，或者只使用缺省的 IPv4 协议。TCP/IP 栈的其余部分是相同的。因此，相同的 TCP、UDP 和 SCTP 传输协议可同时在 IPv4 和 IPv6 上运行。此外，相同的应用程序也可同时在 IPv4 和 IPv6 上运行。图 11–4 显示了 IPv4 和 IPv6 协议如何作为双栈协议在 Internet 协议套件的各个层中工作。

图 11–4 双栈协议体系结构



在双栈方案中，除支持 IPv4 之外，部分主机和路由器还应升级为支持 IPv6。双栈方案可以确保已升级的节点始终能够使用 IPv4 与仅支持 IPv4 的节点进行交互。

Oracle Solaris IPv6 实现

本节介绍在 Oracle Solaris 中启用 IPv6 的文件、命令和守护进程。

IPv6 配置文件

本节介绍属于 IPv6 实现的配置文件：

- 第 231 页中的 “ndpd.conf 配置文件”
- 第 235 页中的 “IPv6 接口配置文件”
- 第 235 页中的 “/etc/inet/ipaddrsel.conf 配置文件”

ndpd.conf 配置文件

/etc/inet/ndpd.conf 文件用于配置由 in.ndpd 相邻节点搜索守护进程使用的选项。对于路由器，主要使用 ndpd.conf 来配置要通告到链路上的站点前缀。对于主机，可使用 ndpd.conf 禁用地址自动配置功能或配置临时地址。

下表显示了 ndpd.conf 文件中使用的关键字。

表 11–2 /etc/inet/ndpd.conf 关键字

变量	说明
ifdefault	指定所有接口的路由器行为。使用以下语法设置路由器参数和相应的值： ifdefault [*variable-value*]
prefixdefault	指定前缀通告的缺省行为。使用以下语法设置路由器参数和相应的值： prefixdefault [*variable-value*]
if	设置每个接口的参数。使用以下语法： if *interface* [*variable-value*]
prefix	通告每个接口的前缀信息。使用以下语法： prefix *prefix/length interface* [*variable-value*]

在 ndpd.conf 文件中，可以将该表中的关键字与一组路由器配置变量结合使用。这些变量在 RFC 2461，《Neighbor Discovery for IP Version 6 (IPv6)》（《IP 版本 6 (IPv6) 的相邻节点搜索》）(http://www.ietf.org/rfc/rfc2461.txt?number=2461)中进行了详细定义。

下表显示了配置接口的变量及其简短定义。

表 11–3 /etc/inet/ndpd.conf 接口配置变量

变量	缺省	定义
AdvRetransTimer	0	指定路由器所发送的通告消息中 "Retrans Timer"（重新传输计时器）字段的值。
AdvCurHopLimit	Internet 的当前直径	指定路由器所发送的通告消息中当前跃点限制的值。
AdvDefaultLifetime	3 + MaxRtrAdvInterval	指定路由器通告的缺省生命周期。
AdvLinkMTU	0	指定路由器所发送的最大传输单元 (Maximum Transmission Unit, MTU) 值。零表示没有为路由器指定 MTU 选项。
AdvManaged Flag	False	指示路由器通告中 "Manage Address Configuration"（管理地址配置）标志的值。
AdvOtherConfigFlag	False	指示路由器通告中 "Other Stateful Configuration"（其他有状态配置）标志的值。
AdvReachableTime	0	指定路由器所发送的通告消息中 "Reachable Time"（可访问时间）字段的值。
AdvSendAdvertisements	False	指示节点是否应当发出通告并响应路由器请求。需要在 ndpd.conf 文件中将该变量明确设置为 "TRUE" 以启用路由器通告功能。有关更多信息，请参阅第 155 页中的 “如何配置启用了 IPv6 的路由器”。

表 11–3 /etc/inet/ndpd.conf 接口配置变量 （续）

变量	缺省	定义
DupAddrDetect Transmits	1	定义在对本地节点地址进行重复地址检测期间，相邻节点搜索协议应当发送的连续相邻节点请求消息的数量。
MaxRtrAdvInterval	600 秒	指定在两次发送未经请求的多播通告之间等待的最长时间。
MinRtrAdvInterval	200 秒	指定在两次发送未经请求的多播通告之间等待的最短时间。
StatelessAddrConf	True	控制节点是否通过无状态地址自动配置功能来配置节点的 IPv6 地址。如果在 ndpd.conf 中声明为 False，则必须手动配置地址。有关更多信息，请参阅第 161 页中的“如何配置用户指定的 IPv6 标记”。
TmpAddrsEnabled	False	指示是否为一个节点的所有接口或某个特定接口创建临时地址。有关更多信息，请参阅第 159 页中的“如何配置临时地址”。
TmpMaxDesyncFactor	600 秒	指定一个随机值，启动 in.ndpd 命令时会从首选的生命周期变量 TmpPreferredLifetime 中减去该值。TmpMaxDesyncFactor 变量用于防止网络上的所有系统同时重新生成它们的临时地址。TmpMaxDesyncFactor 允许您更改这个随机值的上界。
TmpPreferredLifetime	False	设置临时地址的首选生命周期。有关更多信息，请参阅第 159 页中的“如何配置临时地址”。
TmpRegenAdvance	False	为临时地址指定地址过时之前的前导时间。有关更多信息，请参阅第 159 页中的“如何配置临时地址”。
TmpValidLifetime	False	设置临时地址的有效生命周期。有关更多信息，请参阅第 159 页中的“如何配置临时地址”。

下表显示了用于配置 IPv6 前缀的变量。

表 11–4 /etc/inet/ndpd.conf 前缀配置变量

变量	缺省	定义
AdvAutonomousFlag	True	指定 "Prefix Information"（前缀信息）选项中 "Autonomous Flag"（自治标志）字段的值。
AdvOnLinkFlag	True	指定 "Prefix Information"（前缀信息）选项中“在链路 (on-link)”标记（“L 位”）的值。
AdvPreferredExpiration	未设置	指定首选的前缀失效日期。
AdvPreferredLifetime	604800 秒	指定 "Prefix Information"（前缀信息）选项中首选生命周期的值。
AdvValidExpiration	未设置	指定有效的前缀失效日期。

表 11–4 /etc/inet/ndpd.conf 前缀配置变量 （续）

变量	缺省	定义
AdvValidLifetime	2592000 秒	指定所配置的前缀的有效生命周期。

示例 11–1 /etc/inet/ndpd.conf 文件

以下示例显示了如何在 ndpd.conf 文件中使用关键字和配置变量。删除注释符号 (#) 可激活相应的变量。

```
# ifdefault       [variable-value ]*
# prefixdefault [variable-value ]*
# if ifname     [variable-value ]*
# prefix prefix/length  ifname
#
#  Per interface configuration variables
#
#DupAddrDetectTransmits
#AdvSendAdvertisements
#MaxRtrAdvInterval
#MinRtrAdvInterval
#AdvManagedFlag
#AdvOtherConfigFlag
#AdvLinkMTU
#AdvReachableTime
#AdvRetransTimer
#AdvCurHopLimit
#AdvDefaultLifetime
#
# Per Prefix:  AdvPrefixList configuration variables
#
#
#AdvValidLifetime
#AdvOnLinkFlag
#AdvPreferredLifetime
#AdvAutonomousFlag
#AdvValidExpiration
#AdvPreferredExpiration

ifdefault AdvReachableTime 30000 AdvRetransTimer 2000
prefixdefault AdvValidLifetime 240m AdvPreferredLifetime 120m

if qe0 AdvSendAdvertisements 1
prefix 2:0:0:56::/64 qe0
prefix fec0:0:0:56::/64 qe0

if qe1 AdvSendAdvertisements 1
prefix 2:0:0:55::/64 qe1
prefix fec0:0:0:56::/64 qe1

if hme1 AdvSendAdvertisements 1
prefix  2002:8192:56bb:1::/64 qfe0

if hme1 AdvSendAdvertisements 1
prefix  2002:8192:56bb:2::/64 hme1
```

IPv6 接口配置文件

IPv6 在启动时使用 `/etc/hostname6.`*interface* 文件自动定义 IPv6 逻辑接口。如果在安装 Oracle Solaris 的过程中选择 "IPv6 Enabled"（启用 IPv6）选项，除了 `/etc/hostname.`*interface* 文件之外，安装程序还会为主网络接口创建一个 `/etc/hostname6.`*interface* 文件。

如果在安装过程中检测到多个物理接口，系统将提示您是否要配置这些接口。安装程序会为指定的其他各个接口创建 IPv4 物理接口配置文件和 IPv6 逻辑接口配置文件。

与 IPv4 接口一样，您也可以在安装 Oracle Solaris 之后手动配置 IPv6 接口。请为新接口创建 `/etc/hostname6.`*interface* 文件。有关手动配置接口的说明，请参阅第 6 章，管理网络接口（任务）。

网络接口配置文件名称的语法如下所示：

```
hostname.interface
hostname6.interface
```

interface 变量的语法如下所示：

dev`[.`*module*`[.`*module* `...]]`*PPA*

dev 指示网络接口设备。设备可以是物理网络接口（如 `eri` 或 `qfe`），也可以是逻辑接口（如隧道）。有关更多详细信息，请参见第 235 页中的 “IPv6 接口配置文件”。

Module 列出一个或多个在检测设备时要推入设备的 `STREAMS` 模块。

PPA 指示物理连接点。

也可以使用语法 [.[.]]。

示例 11–2 IPv6 接口配置文件

以下是 IPv6 配置文件有效名称的示例：

```
hostname6.qfe0
hostname.ip.tun0
hostname.ip6.tun0
hostname6.ip6to4tun0
hostname6.ip.tun0
hostname6.ip6.tun0
```

`/etc/inet/ipaddrsel.conf` 配置文件

`/etc/inet/ipaddrsel.conf` 文件包含 IPv6 缺省地址选择策略表。如果在安装 Oracle Solaris 时启用了 IPv6，则该文件包含表 11–5 中所示的内容。

可以编辑 /etc/inet/ipaddrsel.conf 的内容。但是，在大多数情况下，应当避免修改此文件。如果一定要进行修改，请参阅第 197 页中的 “如何管理 IPv6 地址选择策略表”过程。有关 ippaddrsel.conf 的更多信息，请参阅第 236 页中的 “修改 IPv6 地址选择策略表的原因”和 ipaddrsel.conf(4) 手册页。

IPv6 相关命令

本节介绍实现 Oracle Solaris IPv6 时添加的命令，还会介绍为支持 IPv6 而对现有命令进行的修改。

ipaddrsel 命令

使用 ipaddrsel 命令，可以修改 IPv6 缺省地址选择策略表。

Oracle Solaris 内核使用 IPv6 缺省地址选择策略表为 IPv6 数据包头执行目标地址排序和源地址选择。/etc/inet/ipaddrsel.conf 文件包含该策略表。

下表列出了缺省地址的格式以及它们的策略表优先级。有关 IPv6 地址选择的技术详细信息，请参见 inet6(7P) 手册页。

表 11–5 IPv6 地址选择策略表

前缀	优先级	定义
::1/128	50	回送
::/0	40	缺省
2002::/16	30	6to4
::/96	20	与 IPv4 兼容
::ffff:0:0/96	10	IPv4

在该表中，IPv6 前缀（::1/128 和 ::/0）优先于 6to4 地址 (2002::/16)、IPv4 地址（::/96 和 ::ffff:0:0/96）。因此，在缺省情况下，内核将为转至另一个 IPv6 目标的包选择接口的全局 IPv6 地址。接口的 IPv4 地址具有较低的优先级，对于转至 IPv6 目标的包尤其如此。如果给出了选定的 IPv6 源地址，内核针对目标地址也使用 IPv6 格式。

修改 IPv6 地址选择策略表的原因

在许多情况下，您不必更改 IPv6 缺省地址选择策略表。如果确实需要管理策略表，请使用 ipaddrsel 命令。

在下列情况下，您可能希望修改策略表：

- 如果系统中有一个用于 6to4 隧道的接口，可以赋予 6to4 地址更高的优先级。
- 如果希望与特定的目标地址进行通信时仅使用特定的源地址，可以将这些地址添加到策略表中。然后，可以使用 `ifconfig` 将这些地址标记为首选地址。
- 如果希望 IPv4 地址优先于 IPv6 地址，可以将 `::ffff:0:0/96` 的优先级更改为较大的数字。
- 如果需要为过时的地址指定较高的优先级，可以将过时的地址添加到策略表中。例如，现在，本地站点地址在 IPv6 中已过时。这些地址的前缀为 `fec0::/10`。可以更改策略表，以便赋予本地站点地址更高的优先级。

有关 `ipaddrsel` 命令的详细信息，请参阅 `ipaddrsel(1M)` 手册页。

`6to4relay` 命令

使用 *6to4* **隧道连接**，可以在相互隔离的 6to4 站点之间进行通信。但是，要使用本地的非 6to4 IPv6 站点传输包，6to4 路由器必须使用 6to4 中继路由器建立一个隧道。然后，*6to4* **中继路由器**将 6to4 包转发到 IPv6 网络，并最终将其传输到本地 IPv6 站点。如果启用了 6to4 的站点必须与本地 IPv6 站点交换数据，请使用 `6to4relay` 命令启用相应的隧道。

由于使用中继路由器不太安全，因此 Oracle Solaris 在缺省情况下会禁用与中继路由器的隧道连接。在部署该方案之前，请认真考虑在建立通往 6to4 中继路由器的隧道时所涉及的问题。有关 6to4 中继路由器的详细信息，请参阅第 256 页中的 “6to4 中继路由器隧道的注意事项”。如果决定启用 6to4 中继路由器支持，可以参阅第 167 页中的 “如何配置 6to4 隧道”中的相关操作步骤。

6to4relay 的语法

`6to4relay` 命令的语法如下：

```
6to4relay -e [-a IPv4-address] -d -h
```

`-e`	在 6to4 路由器和某个任播 6to4 中继路由器之间启用隧道支持。隧道端点地址随后将设置为 `192.88.99.1`（6to4 中继路由器任播组的缺省地址）。
`-a` *IPv4-address*	在 6to4 路由器和具有指定 *IPv4-address* 的 6to4 中继路由器之间启用隧道支持。
`-d`	禁用对通往 6to4 中继路由器的隧道的支持，这是 Oracle Solaris 的缺省设置。
`-h`	显示 `6to4relay` 的帮助。

有关更多信息，请参阅 `6to4relay(1M)` 手册页。

示例 11–3 6to4 中继路由器支持的缺省状态

不带参数的 `6to4relay` 命令显示 6to4 中继路由器支持的当前状态。以下示例显示了在 Oracle Solaris 中实现的 IPv6 的缺省状态。

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is disabled
```

示例 11–4 在启用了 6to4 中继路由器支持的情况下所显示的状态

如果启用了中继路由器支持，`6to4relay` 将显示以下输出：

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is enabled
IPv4 destination address of Relay Router=192.88.99.1
```

示例 11–5 指定了 6to4 中继路由器时显示的状态

如果为 `-6to4relay` 命令指定了 `a` 选项和 IPv4 地址，将显示用 `-a` 提供的 IPv4 地址，而不显示 `192.88.99.1`。

`6to4relay` 不报告 `-d`、`-e` 和 `-a` *IPv4 address* 选项是否成功执行。但是，`6to4relay` 会显示在运行这些选项时可能生成的任何错误消息。

为支持 IPv6 而对 `ifconfig` 命令进行的扩展

可以使用 `ifconfig`命令对IPv6 接口和隧道连接模块进行检测。`ifconfig`使用一组经过扩展的 IOCtl 来配置 IPv4 和 IPv6 网络接口。下面说明了可支持 IPv6 操作的 `ifconfig` 选项。有关涉及 `ifconfig` 命令的一系列 IPv4 和 IPv6 任务，请参见第 178 页中的 “使用 ifconfig 命令监视接口配置”。

`index`	设置接口索引。
`tsrc/tdst`	设置隧道源或目标。
`addif`	创建下一个可用的逻辑接口。
`removeif`	删除具有特定 IP 地址的逻辑接口。
`destination`	设置接口的点对点目标地址。
`set`	为接口设置地址和/或网络掩码。
`subnet`	设置接口的子网地址。
`xmit/-xmit`	启用或禁用在接口上传输包。

第 7 章，配置 IPv6 网络（任务）提供了 IPv6 配置过程。

示例 11–6 在 `ifconfig` 命令中使用 `-addif` 选项添加 IPv6 逻辑接口

以下形式的 `ifconfig` 命令创建 `hme0:3` 逻辑接口：

示例 11–6 在 ifconfig 命令中使用 -addif 选项添加 IPv6 逻辑接口 （续）

```
# ifconfig hme0 inet6 addif up
Created new logical interface hme0:3
```

以下形式的 ifconfig 可验证是否创建了新接口：

```
# ifconfig hme0:3 inet6
hme0:3: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6  inet6 fe80::203:baff:fe11:b321/10
```

示例 11–7 在 ifconfig 命令中使用 -removeif 选项删除 IPv6 逻辑接口

以下形式的 ifconfig 命令删除 hme0:3 逻辑接口：

```
# ifconfig hme0:3 inet6 down

# ifconfig hme0 inet6 removeif 1234::5678
```

示例 11–8 使用 ifconfig 配置 IPv6 隧道源

```
# ifconfig ip.tun0 inet6 plumb index 13
```

打开要与物理接口名称相关联的隧道。

```
# ifconfig ip.tun0 inet6
ip.tun0: flags=2200850<POINTOPOINT,RUNNING,MULTICAST,NONUD,
#IPv6> mtu 1480 index 13
        inet tunnel src 0.0.0.0
        inet6 fe80::/10 --> ::
```

配置 TCP/IP 使用隧道设备并报告设备状态所需要的流。

```
# ifconfig ip.tun0 inet6 tsrc 120.46.86.158 tdst 120.46.86.122
```

配置隧道的源地址和目标地址。

```
# ifconfig ip.tun0 inet6
ip.tun0: flags=2200850<POINTOPOINT,RUNNING,MULTICAST,NONUD,
IPv6> mtu 1480 index 13
        inet tunnel src 120.46.86.158  tunnel dst 120.46.86.122
        inet6 fe80::8192:569e/10 --> fe80::8192:567a
```

在配置之后报告设备的新状态。

示例 11–9 通过 ifconfig 配置 6to4 隧道（长格式）

以下示例显示在 6to4 伪接口配置中子网 ID 为 1，并以十六进制形式指定主机 ID。

```
# ifconfig ip.6to4tun0 inet6 plumb
# ifconfig ip.6to4tun0 inet tsrc 129.146.86.187 \
2002:8192:56bb:1::8192:56bb/64 up
```

示例 11–9 通过 ifconfig 配置 6to4 隧道（长格式） （续）

```
# ifconfig ip.6to4tun0 inet6
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6>mtu 1480 index 11
        inet tunnel src 129.146.86.187
        tunnel hop limit 60
        inet6 2002:8192:56bb:1::8192:56bb/64
```

示例 11–10 通过 ifconfig 配置 6to4 隧道（短格式）

以下示例显示了配置 6to4 隧道的短格式：

```
# ifconfig ip.6to4tun0 inet6 plumb
# ifconfig ip.6to4tun0 inet tsrc 129.146.86.187 up

# ifconfig ip.6to4tun0 inet6
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6>mtu 1480 index 11
        inet tunnel src 129.146.86.187
        tunnel hop limit 60
        inet6 2002:8192:56bb::1/64
```

为支持 IPv6 而对 netstat 命令进行的修改

netstat 命令显示 IPv4 和 IPv6 网络状态。可通过在 /etc/default/inet_type 文件中设置 DEFAULT_IP 值或者使用 -f 命令行选项来选择要显示的协议信息。如果永久设置 DEFAULT_IP，则可以确保 netstat 仅显示 IPv4 信息。可以使用 -f 选项来覆盖该设置。有关 inet_type 文件的更多信息，请参见 inet_type(4) 手册页。

netstat 命令的 -p 选项显示 net-to-media 表。对于 IPv4，该表是 ARP 表；对于 IPv6，该表是相邻节点高速缓存。有关详细信息，请参见 netstat(1M) 手册页。有关使用此命令的过程的说明，请参见第 185 页中的 “如何显示套接字的状态”。

为支持 IPv6 而对 snoop 命令进行的修改

snoop 命令可以捕获 IPv4 和 IPv6 包。此命令可以显示 IPv6 数据包头、IPv6 扩展头、ICMPv6 数据包头和相邻节点搜索协议数据。缺省情况下，snoop 命令既可以显示 IPv4 包又可以显示 IPv6 包。如果您指定了 ip 或 ip6 协议关键字，snoop 命令将只显示 IPv4 包或 IPv6 包。使用 IPv6 的过滤选项，可以对所有的 IPv4 和 IPv6 包进行过滤，以便仅显示 IPv6 包。有关详细信息，请参见 snoop(1M) 手册页。有关使用 snoop 命令的过程，请参见第 196 页中的 “如何监视 IPv6 网络通信”。

为支持 IPv6 而对 route 命令进行的修改

route 命令既作用于 IPv4 路由又作用于 IPv6 路由，IPv4 路由是缺省设置。如果在命令行中紧跟 route 命令之后使用 -inet6 选项，系统将针对 IPv6 路由执行操作。有关详细信息，请参见 route(1M) 手册页。

为支持 IPv6 而对 ping 命令进行的修改

ping 命令既可以使用 IPv4 协议又可以使用 IPv6 协议来探测目标主机。具体选择哪个协议取决于由特定目标主机的名称服务器所返回的地址。缺省情况下，如果名称服务器返回目标主机的 IPv6 地址，ping 命令将使用 IPv6 协议。如果名称服务器仅返回 IPv4 地址，ping 命令将使用 IPv4 协议。可以使用 -A 命令行选项指定要使用的协议以覆盖该操作。

有关详细信息，请参见 ping(1M) 手册页。有关使用 ping 的过程，请参阅第 188 页中的 “使用 ping 命令探测远程主机”。

为支持 IPv6 而对 traceroute 命令进行的修改

可以使用 traceroute 命令跟踪到特定主机的 IPv4 和 IPv6 路由。从协议的角度看，traceroute 与 ping 使用相同的算法。使用 -A 命令行选项可覆盖此选择。使用 -a 命令行选项，可以跟踪到多宿主主机的每个地址的各个单独路由。

有关详细信息，请参见 traceroute(1M) 手册页。有关使用 traceroute 的过程，请参阅第 192 页中的 “使用 traceroute 命令显示路由信息”。

与 IPv6 相关的守护进程

本节讨论与 IPv6 相关的守护进程。

用于相邻节点搜索功能的 in.ndpd 守护进程

in.ndpd 守护进程可实现 IPv6 相邻节点搜索协议和路由器搜索。该守护进程还可实现 IPv6 的地址自动配置功能。下面显示了 in.ndpd 支持的选项。

-d　启用调试功能。

-D　针对特定事件启用调试功能。

-f　指定要从中读取配置数据的文件，而不是缺省的 /etc/inet/ndpd.conf 文件。

-I　列显与每个接口相关的信息。

-n　不回送路由器通告。

-r　忽略收到的包。

-v　指定详细模式，报告各种类型的诊断消息。

-t　启用包跟踪功能。

in.ndpd 守护进程由在 /etc/inet/ndpd.conf 配置文件中设置的参数以及 /var/inet/ndpd_state.*interface* 启动文件中适用的参数来控制。

如果 /etc/inet/ndpd.conf 文件存在，系统将解析该文件并使用它将节点配置为路由器。表 11–2 列出了此文件中可能出现的有效关键字。当主机引导之后，路由器可能无法立即使用。由路由器通告的包可能会被丢弃，当然，它们可能将无法送达到主机。

/var/inet/ndpd_state.*interface* 文件是一个状态文件。由每个节点定期更新。当该节点失败并重新启动之后，该节点可以在没有路由器的情况下配置其接口。此文件包含接口地址、上次更新文件的时间以及文件的有效期。此文件还包含从以前的路由器通告中“获知”的其他参数。

注 – 您不必修改状态文件的内容，in.ndpd 守护进程会自动维护状态文件。

有关配置变量和可允许值的列表，请参见 in.ndpd(1M) 手册页和 ndpd.conf(4) 手册页。

用于 IPv6 路由的 in.ripngd 守护进程

in.ripngd 守护进程可实现用于 IPv6 路由器的下一代路由信息协议 (Routing Information Protocol next-generation, RIPng)。RIPng 定义 IPv6 中与 RIP 等效的协议。在使用 routeadm 命令配置 IPv6 路由器并启用 IPv6 路由时，in.ripngd 守护进程可在路由器上实现 RIPng。

下面显示了 RIPng 支持的选项：

-p *n*　*n* 指定用于收发 RIPng 包的备用端口号。

-q　禁止显示路由信息。

-s　强制显示路由信息，即使该守护进程充当路由器也是如此。

-P　禁止使用毒性逆转 (poison reverse)。

-S　如果 in.ripngd 不充当路由器，该守护进程将只输入每个路由器的缺省路由。

inetd 守护进程和 IPv6 服务

启用了 IPv6 的服务器应用程序可以既处理 IPv4 请求又处理 IPv6 请求，也可以仅处理 IPv6 请求。服务器始终通过 IPv6 套接字处理请求。另外，服务器还与相应的客户机使用相同的协议。

要为 IPv6 添加或修改服务，请使用服务管理工具 (Service Management Facility, SMF) 中的命令。

- 有关 SMF 命令的信息，请参阅《Oracle Solaris 管理：基本管理》中的“SMF 命令行管理实用程序”。
- 有关使用 SMF 配置在 SCTP 上运行的 IPv4 服务清单的示例任务，请参阅第 120 页中的 “如何添加使用 SCTP 协议的服务”。

要配置 IPv6 服务，必须确保该服务 inetadm 配置文件中的 proto 字段中列出了相应的值：

- 对于既处理 IPv4 请求又处理 IPv6 请求的服务，请选择 tcp6、udp6 或 sctp6。如果 proto 的值为 tcp6、udp6 或 sctp6，则会导致 inetd 向服务器传递 IPv6 套接字。服务器中包含映射到 IPv4 的地址以备 IPv4 客户机发出请求。
- 对于仅处理 IPv6 请求的服务，请选择 tcp6only 或 udp6only。如果 proto 的值为 tcp6only 或 udp6only，inetd 会向服务器传递 IPv6 套接字。

如果用其他实现来替代 Oracle Solaris 命令，则必须验证所实现的服务是否支持 IPv6。如果该服务不支持 IPv6，则必须将 proto 值指定为 tcp、udp 或 sctp。

下面是针对 echo 服务清单运行 inetadm 时生成的配置文件，该服务清单既支持 IPv4 又支持 IPv6，并且在 SCTP 上运行：

```
# inetadm -l svc:/network/echo:sctp_stream
    SCOPE    NAME=VALUE      name="echo"
             endpoint_type="stream"
             proto="sctp6"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
             user="root"
    default  bind_addr=""
    default  bind_fail_max=-1
    default  bind_fail_interval=-1
    default  max_con_rate=-1
    default  max_copies=-1
    default  con_rate_offline=-1
    default  failrate_cnt=40
    default  failrate_interval=60
    default  inherit_env=TRUE
    default  tcp_trace=FALSE
    default  tcp_wrappers=FALSE
```

要更改 proto 字段的值，请使用以下语法：

```
# inetadm -m FMRI proto="transport-protocols"
```

随 Oracle Solaris 软件一起提供的所有服务器都只需要一个用来将 proto 指定为 tcp6、udp6 或 sctp6 的配置文件项。但是，远程 shell 服务器 (shell) 和远程执行服务器 (exec) 现在由单个服务实例组成，该服务实例要求 proto 值中同时包含 tcp 和 tcp6only 值。例如，要为 shell 设置 proto 值，可发出以下命令：

```
# inetadm -m network/shell:default proto="tcp,tcp6only"
```

有关写入使用套接字且启用了 IPv6 的服务器的更多详细信息，请参见《编程接口指南》中的“套接字 API 的 IPv6 扩展”。

在为 IPv6 配置服务时的注意事项

在为 IPv6 添加或修改服务时，请记住以下几点：

- 需要将 `proto` 值指定为 `tcp6`、`sctp6` 或 `udp6`，以便既支持 IPv4 连接又支持 IPv6 连接。如果将 `proto` 值指定为 `tcp`、`sctp` 或 `udp`，则该服务仅使用 IPv4。
- 尽管可以为 `inetd` 添加使用一对多样式的 SCTP 套接字的服务实例，但是建议不要这样做。`inetd` 不能处理一对多样式的 SCTP 套接字。
- 如果某个服务因其 `wait-status` 或 `exec` 属性不同而需要两项，则必须从初始服务创建两个实例/服务。

IPv6 相邻节点搜索协议

IPv6 引入了相邻节点搜索协议，如 RFC 2461，《Neighbor Discovery for IP Version 6 (IPv6)》（《IP 版本 6 (IPv6) 的相邻节点搜索》）(`http://www.ietf.org/rfc/rfc2461.txt?number=2461`)中所述。有关相邻节点搜索的主要功能的概述，请参阅第 72 页中的 “IPv6 相邻节点搜索协议概述”。

本节讨论相邻节点搜索协议的以下功能：

- 第 244 页中的 “相邻节点搜索功能中的 ICMP 消息”
- 第 245 页中的 “自动配置过程”
- 第 246 页中的 “相邻节点请求和不可访问性”
- 第 247 页中的 “重复地址检测算法”
- 第 247 页中的 “相邻节点搜索协议与 ARP 和相关 IPv4 协议的比较”

相邻节点搜索功能中的 ICMP 消息

相邻节点搜索功能定义了五种新的 Internet 控制消息协议 (Internet Control Message Protocol, ICMP) 消息。这些消息具有以下用途：

- **路由器请求** – 当接口变为启用状态时，主机可以发送路由器请求消息。这种请求要求路由器立即生成路由器通告，而不是在下次预定时间生成。
- **路由器通告** – 路由器通告其存在状态、各种链路参数和 Internet 参数。路由器会定期或在响应路由器请求消息时发出通告。路由器通告包含用于确定是否在链路中或用于配置地址的前缀以及建议的跃点限制值等。
- **相邻节点请求** – 节点发送相邻节点请求消息，以确定相邻节点的链路层地址，并验证相邻节点是否仍可以通过缓存的链路层地址进行访问。相邻节点请求还可用于检测重复地址。
- **相邻节点通告** – 节点发送相邻节点通告消息以响应相邻节点请求消息。节点还可以发送未经请求的相邻节点通告以公布链路层地址更改。

- **重定向**－路由器使用重定向消息来通告主机：对于某个目标有一个更好的第一个跃点，或者该目标在同一个链路上。

自动配置过程

本节概述在自动配置过程中由接口执行的典型步骤。自动配置仅在能够进行多播的链路上执行。

1. 启用能够进行多播的接口，例如，在启动某个节点上的系统时启用该接口。
2. 节点在执行自动配置过程时首先为接口生成链路本地地址。

   链路本地地址是根据接口的介质访问控制 (Media Access Control, MAC) 地址构造的。
3. 节点发送相邻节点请求消息，其中包含暂定为目标的链路本地地址。

   发送此消息的目的在于验证要使用的地址未由链路上的其他节点占用。在验证之后，可以将链路本地地址指定给接口。

   a. 如果建议的地址已被另一个节点使用，则该节点将返回一条相邻节点通告，声明该地址正在使用中。

   b. 如果另一个节点也正在尝试使用同一地址，则该节点也会针对该目标发送一条相邻节点请求。

      相邻节点请求传输或重新传输的数量以及连续请求之间的延迟与链路有关。如有必要，可以设置这些参数。
4. 如果某个节点发现它要使用的链路本地地址不唯一，则自动配置过程会停止。此时，您必须手动配置该接口的链路本地地址。

   要简化恢复操作，可以提供一个备用接口 ID 来覆盖缺省标识符。这样，自动配置机制就可以使用这个可能唯一的新接口 ID 继续工作。
5. 如果某个节点发现它要使用的链路本地地址唯一，该节点会将此地址指定给这个接口。

   此时，该节点与相邻节点具有 IP 级别的连通性。其余的自动配置步骤只能由主机执行。

获取路由器通告

自动配置的下一个阶段涉及到获取路由器通告或者确定是否没有路由器存在。如果存在路由器，路由器会发送路由器通告，以指定主机应当执行哪种类型的自动配置。

路由器定期发送路由器通告。但是，相邻通告之间的延迟通常比执行自动配置的主机可以等待的时间要长。为了快速获取通告，主机可以向所有路由器多播组发送一个或多个路由器请求。

前缀配置变量

路由器通告还包含前缀变量，其中包含无状态地址自动配置用于生成前缀的信息。路由器通告中 "Stateless Address Autoconfiguration"（无状态地址自动配置）字段是单独处理的。"Address Autoconfiguration"（地址自动配置）标志是一个包含前缀信息的选项字段，它指示该选项是否可以应用于无状态自动配置过程。如果该选项字段确实适用，则其他选项字段中包含具有生命周期值的子网前缀。这些值指示根据前缀创建的地址保持优先和有效的时间长度。

因为路由器会定期生成路由器通告，所以主机会不断收到新通告。启用了 IPv6 的主机可处理包含在每个通告中的信息。主机会添加到这些信息中，还会刷新在以前的通告中收到的信息。

地址唯一性

出于安全方面的考虑，在将每个地址指定给接口之前，必须测试它们是否唯一。对于通过无状态自动配置过程创建的地址，情况会有所不同。地址是否唯一主要由地址中基于接口 ID 创建的那一部分来确定。因此，如果经过验证，节点的链路本地地址唯一，则无需再逐一测试其他地址。这些地址必须是根据同一个接口 ID 创建的。与之相反，对于手动获取的所有地址，必须逐一测试它们是否唯一。某些站点的系统管理员认为执行重复地址检测得不偿失。对于这些站点，可通过设置每接口配置标志来禁用重复地址检测功能。

为了加速自动配置过程，主机可以生成其链路本地地址，并在等待路由器通告的同时验证该地址是否唯一。路由器可能会延迟几秒来响应路由器请求。因此，如果连续执行两个步骤，则完成自动配置所必需的总时间可能会非常长。

相邻节点请求和不可访问性

相邻节点搜索功能使用**相邻节点请求**消息来确定是否可以向同一个单播地址指定多个节点。**相邻节点不可访问性检测**功能检测相邻节点或到相邻节点的转发路径中是否有故障。该检测功能要求确认发送到某个相邻节点的包能够实际到达该相邻节点，还确定节点的 IP 层是否能够正确处理这些包。

相邻节点不可访问性检测功能使用来自以下两个来源的确认：上层协议和相邻节点请求消息。如有可能，上层协议会确认某个连接正在执行**转发**。例如，当收到新的 TCP 确认时，上层协议会确认以前发送的数据已正确传送。

如果某个节点没有收到来自上层协议的肯定确认，该节点将发送单播相邻节点请求消息。这些消息会请求相邻节点通告，并根据此通告确认下一个跃点的可访问性。为了减少不必要的网络通信流量，探测消息只会发送到该节点将包实际发送到的相邻节点。

重复地址检测算法

为了确保所有已配置的地址在特定链路上的唯一性，节点需要针对这些地址运行**重复地址检测**算法。在将这些地址指定给接口之前，必须先针对节点运行该算法。重复地址检测算法是针对所有地址执行的。

本节中描述的自动配置过程仅适用于主机，而不适用于路由器。因为主机自动配置过程使用由路由器通告的信息，所以路由器需要通过其他方法进行配置。但是，路由器可使用本章中描述的机制来生成链路本地地址。另外，在将地址指定给接口之前，路由器应当能够通过针对所有地址的重复地址检测算法。

代理通告

代表目标地址接受包的路由器可以发出不可覆盖的相邻节点通告。路由器可以接受无法响应相邻节点请求的目标地址的包。目前未指定对于代理的使用。但是，代理通告有可能会用来处理诸如已移出链路的移动节点之类的情况。请注意，在处理未实现此协议的节点时，不应将使用代理作为一般机制。

传入负载平衡

具有复制接口的节点可能需要在同一个链路上的多个网络接口之间，对所收到的传入包进行负载平衡。这样的节点会将多个链路本地地址指定给同一个接口。例如，单个网络驱动程序可以将多个网络接口卡表示为具有多个链路本地地址的单个逻辑接口。

负载平衡的处理方式如下：允许路由器忽略来自路由器通告包的源链路本地地址。因此，相邻节点必须使用相邻节点请求消息来获知路由器的链路本地地址。于是，所返回的相邻节点通告消息可能包含链路本地地址，这些地址会因请求发出者而异。

链路本地地址更改

已知其链路本地地址发生更改的节点可以发出未经请求的多播相邻节点通告包。该节点可以向所有的节点发送多播包，从而更新所缓存的已无效的链路本地地址。发送未经请求的通告仅是为了提高性能。相邻节点不可访问性检测算法可确保所有的节点都能够可靠地搜索新地址，尽管延迟时间可能会稍长些。

相邻节点搜索协议与 ARP 和相关 IPv4 协议的比较

IPv6 相邻节点搜索协议的功能与下列 IPv4 协议的组合相对应：地址解析协议 (Address Resolution Protocol, ARP)、Internet 控制消息协议 (Internet Control Message Protocol, ICMP)、路由器搜索和 ICMP 重定向。IPv4 对于相邻节点不可访问性检测没有公认的协

议或机制。但是，主机要求确实为停用网关检测指定了一些可能的算法。停用网关检测所解决的问题是相邻节点不可访问性检测所能解决的问题的一部分。

以下是对相邻节点搜索协议和一组相关 IPv4 协议进行的比较。

- 路由器搜索是基础 IPv6 协议集的一部分。IPv6 主机无需针对路由协议执行 snoop 即可查找路由器。IPv4 使用 ARP、ICMP 路由器搜索和 ICMP 重定向来搜索路由器。
- IPv6 路由器通告传输链路本地地址。无需进行其他包交换即可解析路由器的链路本地地址。
- 路由器通告传输链路的站点前缀。无需像在 IPv4 中那样使用单独的机制来配置网络掩码。
- 路由器通告功能允许自动配置地址。在 IPv4 中未实现自动配置过程。
- 相邻节点搜索允许 IPv6 路由器通告主机要在链路上使用的 MTU。因此，在缺乏完善定义的 MTU 的链路上，所有的节点都使用相同的 MTU 值。同一个网络上的 IPv4 主机可能具有不同的 MTU。
- 与 IPv4 广播地址不同的是，IPv6 地址解析多播分布到 40 亿 (2^32) 个多播地址上，这会大大减少目标以外的节点上与地址解析有关的中断。而且，非 IPv6 计算机根本就不应当中断。
- IPv6 重定向包含新的第一个跃点的链路本地地址。在接收重定向消息时无需进行单独的地址解析。
- 多个站点前缀可以与同一个 IPv6 网络相关联。缺省情况下，主机可以从路由器通告中获知所有的本地站点前缀。但是，可以将路由器配置为忽略来自路由器通告的部分或全部前缀。在这种情况下，主机会假定目标位于远程网络上。因此，主机会向路由器发送通信。路由器随后可以根据需要发出重定向命令。
- 与 IPv4 不同的是，IPv6 重定向消息的接收者假定下一个新跃点位于本地网络上。在 IPv4 中，主机根据网络掩码会忽略那些指定下一个跃点不在本地网络上的重定向消息。IPv6 重定向机制与 IPv4 中的 Xredirect 功能相似。重定向机制在非广播链路和共享介质链路上非常有用。在这些网络上，节点不应当检查本地链路目标的所有前缀。
- IPv6 相邻节点不可访问性检测改进了在路由器存在故障时的包传送能力。此功能改进了包在部分故障链路或分区链路上的传送能力，还改进了包在可更改其链路本地地址的节点上的传送能力。例如，移动节点可移出本地网络，而不会因存在过时的 ARP 高速缓存而失去任何连通性。IPv4 没有与相邻节点不可访问性检测相对应的方法。
- 与 ARP 不同的是，相邻节点搜索功能使用相邻节点不可访问性检测机制来检测半链路故障。相邻节点搜索功能可避免在没有双向连通性的情况下向相邻节点发送通信。
- IPv6 主机使用链路本地地址来唯一标识路由器，从而可以维护路由器关联。对于路由器通告和重定向消息，这种路由器标识功能是必需的。如果站点使用新的全局前缀，主机需要维护路由器关联。IPv4 没有与路由器标识功能相对应的方法。

- 因为相邻节点搜索消息在接收时的跃点限制为 255，所以，相邻节点搜索协议不会受到来自链路外节点的欺骗攻击。与之相反，IPv4 链路外节点可以发送 ICMP 重定向消息。IPv4 链路外节点还可以发送路由器通告消息。
- 将地址解析放在 ICMP 层，使得相邻节点搜索比 ARP 更加独立于介质。因此可以使用标准的 IP 验证和安全机制。

IPv6 路由

在无类域间路由 (Classless Inter-Domain Routing, CIDR) 情况下，IPv6 中的路由与 IPv4 路由几乎完全相同。唯一的区别在于地址是 128 位 IPv6 地址，而非 32 位 IPv4 地址。通过非常简单的扩展，所有的 IPv4 路由算法，如 OSPF（开放式最短路径优先）、RIP（路由信息协议）、IDRP（域间路由协议）和 IS-IS（中间系统对中间系统），都可以用来路由 IPv6。

IPv6 还包括可支持功能强大的新路由功能的简单路由扩展。以下是对新路由功能的描述：

- 基于策略、性能、成本等因素选择提供器
- 主机灵活性，可路由到当前位置
- 自动重新寻址，可路由到新地址

通过创建可使用 IPv6 路由选项的 IPv6 地址序列，可以获取新路由功能。IPv6 源使用路由选项列出在通往包目标的途中访问的中间节点（一个或多个）或拓扑组。此功能与 IPv4 的稀疏源路由选项和记录路由选项非常相似。

在大多数情况下，为了使地址序列成为一般功能，必须使用 IPv6 主机将主机所接收包中的路由反向。包必须使用 IPv6 验证头成功地进行验证。包中必须包含地址序列才能将包返回到其始发者。此方法会强制所实现的 IPv6 主机支持对源路由进行处理和反向。对源路由进行处理和反向非常重要，因为它使提供者能够使用实现了新 IPv6 功能（如提供器选择和扩展地址）的主机。

路由器通告

在能够进行多播的链路和点对点链路上，每个路由器都定期向多播组发送一个路由器通告包来公布其可用性。主机将从所有的路由器接收路由器通告，并创建缺省路由器的列表。路由器会频繁生成路由器通告，以便主机可以在几分钟内获知路由器是否存在。但是，路由器进行通告的频率不太高，因此不能依赖通告是否存在来检测路由器故障。可以通过用来确定相邻节点不可访问性的单独的检测算法来检测路由器故障。

路由器通告前缀

路由器通告中包含一系列子网前缀，这些前缀用来确定主机是否与路由器处在同一个链路上（在链路 (on-link)），还可用来配置自治地址。与前缀相关联的标志用来指定特定前缀的预定用法。主机使用通告的“在链路 (on-link)”前缀来创建和维护一个列表，该列表用于确定包的目标是在链路上还是在路由器外部。即使目标没有包含在所通告的任何“在链路 (on-link)”前缀中，目标也可以位于链路上。在这种情况下，路由器可以发送重定向消息。重定向功能通告发送者目标是相邻节点。

路由器通告和每前缀标志使路由器能够通告主机如何执行无状态地址自动配置。

路由器通告消息

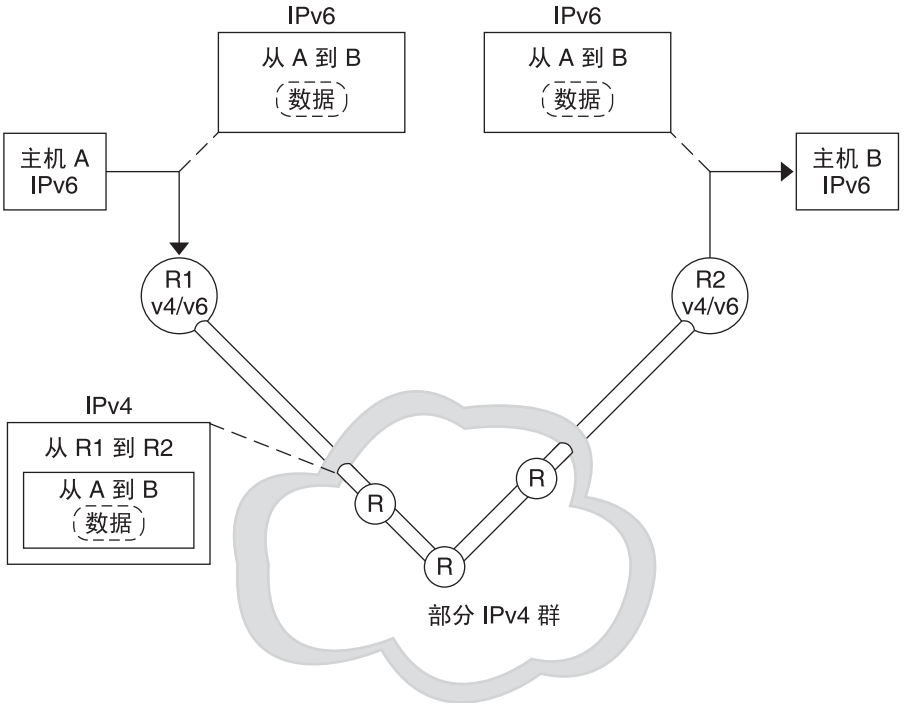
路由器通告消息中还包含主机应当在外发包中使用的 Internet 参数（如跃点限制）。路由器通告消息中还可以包含链路参数，如链路 MTU。此功能允许对临界参数进行集中管理。这些参数可以针对路由器设置，它们可自动传播到所连接的全部主机。

节点可通过向多播组发送相邻节点请求以要求目标节点返回其链路层地址来完成地址解析。多播相邻节点请求消息会发送到目标地址中请求节点的多播地址。目标会在单播相邻节点通告消息中返回其链路层地址。对于启动器和目标来说，一个包请求/响应对就足以解析对方的链路层地址。启动器的相邻节点请求中包括其链路层地址。

IPv6 隧道

为了在双栈 IPv4/IPv6 站点中尽可能减少相关性，两个 IPv6 节点之间路径中的路由器不必都支持 IPv6。可支持类似网络配置的机制称作**隧道**。本质上，IPv6 包放在 IPv4 包中，IPv4 包随后通过 IPv4 路由器进行路由，下图说明通过 IPv4 路由器（在该图中用 "R" 指示）的隧道连接机制。

图 11–5 IPv6 隧道连接机制



Oracle Solaris IPv6 实现中包括两种类型的隧道连接机制：

- 在两个路由器之间配置的隧道，如图 11–5 所示
- 在端点主机终止的自动隧道

出于其他方面的考虑，目前在 Internet 上，例如，在 IPv4 MBONE（多播主干）上，使用已配置的隧道。从操作的角度看，隧道由两个路由器组成，这两个路由器配置为可通过 IPv4 网络建立虚拟的点对点链路。在可以预见的将来，这种隧道很有可能要用在 Internet 的某些部分上。

自动隧道要求使用与 IPv4 兼容的地址。当 IPv6 路由器不可用时，自动隧道可用来连接多个 IPv6 节点。这些隧道可以通过配置自动隧道连接网络接口来从双栈主机或双栈路由器发起。这些隧道总是在双栈主机终止。它们的工作方式如下：通过从与 IPv4 兼容的目标地址中提取目标 IPv4 地址（隧道的端点）来动态确定目标地址。

已配置的隧道

隧道连接接口具有以下格式：

```
ip.tun ppa
```

ppa 是物理连接点。

在系统启动时，隧道连接模块 (tun) 会由 ifconfig 命令推入 IP 的顶层，从而创建一个虚拟接口。推送操作可通过创建相应的 hostname6.* 文件来实现。

例如，要创建通过 IPv4 网络封装 IPv6 包的隧道（即 IPv6 over IPv4 隧道），可以创建以下文件名：

```
/etc/hostname6.ip.tun0
```

在对接口进行检测之后，此文件的内容将传递到 ifconfig。此文件内容将变成配置点对点隧道所必需的参数。

示例 11–11 IPv6 Over IPv4 隧道的 hostname6.ip.tun0 文件

下面举例说明 hostname6.ip.tun0 文件中的各项：

```
tsrc 10.10.10.23 tdst 172.16.7.19 up
addif 2001:db8:3b4c:1:5678:5678::2 up
```

在此示例中，源和目标 IPv4 地址用作自动配置 IPv6 链路本地地址的标记。这些地址分别是 ip.tun0 接口的源和目标。配置了两个接口。配置了 ip.tun0 接口。还配置了一个逻辑接口 ip.tun0:1。该逻辑接口的源和目标 IPv6 地址由 addif 命令指定。

当系统在多用户模式下启动后，这些配置文件的内容将会原封不动地传递到 ifconfig。示例 11–11 中的项与以下各项等效：

```
# ifconfig ip.tun0 inet6 plumb
# ifconfig ip.tun0 inet6 tsrc 10.0.0.23 tdst 172.16.7.19 up
# ifconfig ip.tun0 inet6 addif 2001:db8:3b4c:1:5678:5678::2 up
```

下面显示了此隧道的 ifconfig -a 输出：

```
ip.tun0: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,
  NONUD,IPv6> mtu 1480 index 6
        inet tunnel src 10.0.0.23  tunnel dst 172.16.7.19
        inet6 fe80::c0a8:6417/10 --> fe80::c0a8:713
ip.tun0:1: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
  index 5
        inet6 2001:db8:3b4c:1:5678:5678::2
```

可以使用以下语法，通过向配置文件中添加一些行来配置更多的逻辑接口：

```
addif IPv6-source IPv6-destination up
```

注 – 当隧道的任意一端是通过隧道通告一个或多个前缀的 IPv6 路由器时，不必在隧道配置文件中使用 `addif` 命令。可能只有 `tsrc` 和 `tdst` 才是必需的，因为所有其他地址都是自动配置的。

在某些情况下，需要为特定的隧道手动配置特定的源和目标链路本地地址。可通过更改配置文件的第一行来包括这些链路本地地址。下面是示例行：

```
tsrc 10.0.0.23 tdst 172.16.7.19 fe80::1/10 fe80::2 up
```

请注意，源链路本地地址的前缀长度为 10。在此示例中，`ip.tun0` 接口与以下内容类似：

```
ip.tun0: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
index 6
        inet tunnel src 10.0.0.23  tunnel dst 172.16.7.19
        inet6 fe80::1/10 --> fe80::2
```

要建立通过 IPv6 网络封装 IPv6 包的隧道（即 IPv6 over IPv6 隧道），可以创建以下文件名：

```
/etc/hostname6.ip6.tun0
```

示例 11–12 IPv6 over IPv6 隧道的 `hostname6.ip6.tun0` 文件

下面举例说明通过 IPv6 网络进行的 IPv6 封装的 `hostname6.ip6.tun0` 文件中的各项：

```
tsrc 2001:db8:3b4c:114:a00:20ff:fe72:668c
        tdst 2001:db8:15fa:25:a00:20ff:fe9b:a1c3
fe80::4 fe80::61 up
```

要建立通过 IPv6 网络封装 IPv4 包的隧道（即 IPv4 over IPv6 隧道），可以创建以下文件名：

```
/etc/hostname.ip6.tun0
```

示例 11–13 IPv4 Over IPv6 隧道的 `hostname.ip6.tun0` 文件

下面举例说明通过 IPv6 网络进行的 IPv4 封装的 `hostname.ip6.tun0` 文件中的各项：

```
tsrc 2001:db8:3b4c:114:a00:20ff:fe72:668c
         tdst 2001:db8:15fa:25:a00:20ff:fe9b:a1c3
10.0.0.4 10.0.0.61 up
```

要建立通过 IPv4 网络封装 IPv4 包的隧道（即 IPv4 over IPv4 隧道），可以创建以下文件名：

```
/etc/hostname.ip.tun0
```

示例 11–14 IPv4 Over IPv4 隧道的 `hostname.ip.tun0` 文件

下面举例说明通过 IPv4 网络进行的 IPv4 封装的 `hostname.ip.tun0` 文件中的各项：

```
tsrc 172.16.86.158 tdst 192.168.86.122
10.0.0.4 10.0.0.61 up
```

有关 `tun` 的具体信息，请参见 `tun(7M)` 手册页。有关在转换到 IPv6 的过程中隧道连接概念的一般说明，请参见第 74 页中的 “IPv6 隧道概述”。有关隧道配置过程的说明，请参见第 164 页中的 “针对 IPv6 支持配置隧道所需的任务（任务列表）”。

6to4 自动隧道

Oracle Solaris 中包括 6to4 隧道，这是用来从 IPv4 寻址过渡到 IPv6 寻址的临时首选方法。6to4 隧道允许 IPv6 隔离站点通过一个经由 IPv4 网络（不支持 IPv6）的自动隧道进行通信。要使用 6to4 隧道，必须将 IPv6 网络上的边界路由器配置为 6to4 自动隧道的一个端点。这样，6to4 路由器便可以参与通往另一个 6to4 站点的隧道，如果需要的话，还可以参与通往本地非 6to4 IPv6 站点的隧道。

本节提供有关下列 6to4 主题的参考资料：

- 6to4 隧道的拓扑
- 6to4 寻址（包括通告格式）
- 通过 6to4 隧道的包流的说明
- 6to4 路由器和 6to4 中继路由器之间隧道的拓扑
- 配置 6to4 中继路由器支持之前的注意事项

下表介绍用来配置 6to4 隧道的额外任务，以及获得额外有用信息的资源。

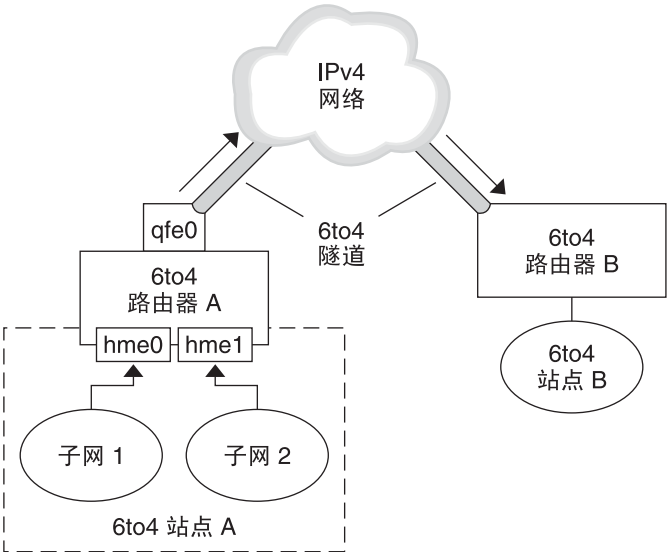
任务或详细信息	参考
用来配置 6to4 隧道的任务	第 167 页中的 “如何配置 6to4 隧道”
与 6to4 相关的 RFC（互联网信息文档和标准）	RFC 3056，《Connection of IPv6 Domains via IPv4 Clouds》（《经 IPv4 云连接 IPv6 域》）(`http://www.ietf.org/rfc/rfc3056.txt`)
有关 `6to4relay` 命令（该命令启用对通往 6to4 中继服务器的隧道的支持）的详细信息	`6to4relay(1M)`

任务或详细信息	参考
6to4 安全问题	《Security Considerations for 6to4》（《6to4 安全考虑》）(http://www.ietf.org/rfc/rfc3964.txt)

6to4 隧道的拓扑

6to4 隧道提供到任何位置的所有 6to4 站点的 IPv6 连接。如果 6to4 隧道配置为向中继路由器转发，该隧道也同样提供到所有 IPv6 站点的连接（包括本地 IPv6 Internet）。下图显示了 6to4 隧道如何提供两个 6to4 站点之间的连接。

图 11–6 两个 6to4 站点之间的隧道



该图描述了两个隔离的 6to4 网络，即站点 A 和站点 B。每个站点都配置了具有到 IPv4 网络的外部连接的路由器。跨 IPv4 网络的 6to4 隧道可以连接两个 6to4 站点。

必须至少配置一个路由器接口来支持 6to4，才能让 IPv6 站点成为 6to4 站点。此接口必须提供与 IPv4 网络的外部连接。在 `qfe0` 上配置的地址必须全局唯一。在上图中，边界路由器 A 的 `qfe0` 接口将站点 A 连接到 IPv4 网络。只有在用 IPv4 地址配置 `qfe0` 接口后，才能将 `qfe0` 配置为 6to4 伪接口。

在上图中，6to4 站点 A 由两个子网组成，这些子网连接到路由器 A 上的 `hme0` 和 `hme1` 接口。站点 A 的任一子网上的所有 IPv6 主机会在接收到来自路由器 A 的通告时重新配置为具有 6to4 派生的地址。

站点 B 是另一个隔离的 6to4 站点。为了正确地从站点 A 接收通信，必须在站点 B 上配置边界路由器以支持 6to4。否则，路由器从站点 A 接收的包将因无法识别而被丢弃。

通过 6to4 隧道的包流

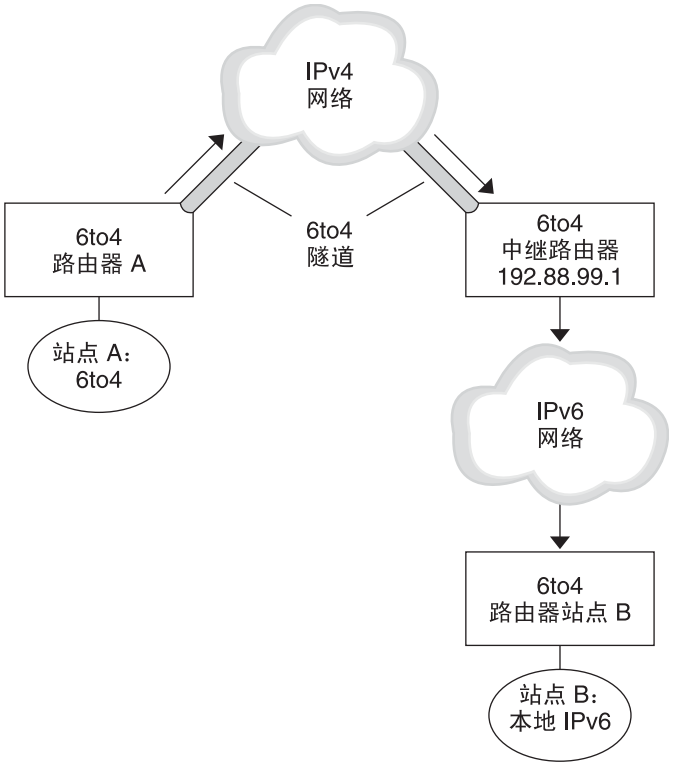
本节介绍从一个 6to4 站点上的主机到远程 6to4 站点上的主机之间的包流。此方案使用图 11–6 中显示的拓扑。而且，它还假定已经配置了 6to4 路由器和 6to4 主机。

1. 6to4 站点 A 的子网 1 上的主机发送传输请求，6to4 站点 B 上的主机作为目标。每个数据包头中均有 6to4 派生的源地址和 6to4 派生的目标地址。
2. 站点 A 的路由器将每个 6to4 包封装到 IPv4 数据包头中。在该过程中，路由器将 IPv4 封装数据包头的目标地址设置为站点 B 的路由器地址。对于每个流经隧道接口的 IPv6 包，其 IPv6 目标地址同时也包含 IPv4 目标地址。因此，路由器将能够确定 IPv4 封装数据包头上设置的 IPv4 目标地址。路由器随后使用标准的 IPv4 路由过程，通过 IPv4 网络转发包。
3. 包通过的任何 IPv4 路由器都使用包的 IPv4 目标地址进行转发。此地址是路由器 B 上某个接口的全局唯一 IPv4 地址，该接口还充当 6to4 伪接口。
4. 来自站点 A 的包到达路由器 B，路由器 B 对 IPv4 数据包头中的 IPv6 包取消封装。
5. 路由器 B 随后使用 IPv6 包中的目标地址将包转发到站点 B 上的接收主机。

6to4 中继路由器隧道的注意事项

6to4 中继路由器充当某些隧道的一个端点，这些隧道的另一个端点是需要与本地非 6to4 IPv6 网络通信的 6to4 路由器。本质上，中继路由器是 6to4 站点和本地 IPv6 站点之间的桥梁。因为此解决方案可能很不安全，所以，在缺省情况下，Oracle Solaris 不启用对 6to4 中继路由器的支持。但是，如果站点需要这样的隧道，可以使用 `6to4relay` 命令来启用下面的隧道连接方案。

图 11–7 从 6to4 站点到 6to4 中继路由器的隧道



在图 11–7 中，6to4 站点 A 需要与本机 IPv6 站点 B 上的节点通信。该图说明了从站点 A 到 IPv4 网络上 6to4 隧道的通信路径。该隧道将 6to4 路由器 A 和 6to4 中继路由器作为其端点。IPv6 站点 B 所连接到的 IPv6 网络位于 6to4 中继路由器的外部。

6to4 站点和本地 IPv6 站点之间的包流

本节介绍从 6to4 站点到本地 IPv6 站点之间的包流。此方案使用图 11–7 中显示的拓扑。

1. 6to4 站点 A 上的主机发送一个将本地 IPv6 站点 B 上的主机指定为目标的传输信号。每个包头中具有作为其源地址的 6to4 派生地址。目标地址是标准的 IPv6 地址。
2. 站点 A 的 6to4 路由器将每个包封装到 IPv4 数据包头中，该 IPv4 数据包头将 6to4 中继路由器的 IPv4 地址作为其目标地址。6to4 路由器随后使用标准的 IPv4 路由过程，通过 IPv4 网络转发包。包遇到的任何 IPv4 路由器都会将包转发到 6to4 中继路由器。
3. 物理位置距离站点 A 最近的任播 6to4 中继路由器检索以 `192.88.99.1` 任播组为目标的包。

注 – 6to4 中继路由器属于 6to4 中继路由器任播组，它的 IP 地址为 `192.88.99.1`。此任播地址是 6to4 中继路由器的缺省地址。如果您需要使用特定的 6to4 中继路由器，则可以覆盖缺省设置并指定该路由器的 IPv4 地址。

4. 该中继路由器会对 6to4 包中的 IPv4 数据包头取消封装，并显示本地 IPv6 目标地址。
5. 然后，中继路由器仅将 IPv6 包发送到 IPv6 网络中，站点 B 中的路由器最终将在该网络中检索到这些包。然后，路由器将这些包转发到目标 IPv6 节点。

Oracle Solaris 名称服务的 IPv6 扩展

本节介绍在实现 IPv6 时引入的命名更改。可以将 IPv6 地址存储在任何 Oracle Solaris 名称服务（如 NIS、LDAP、DNS 和 files）中。还可以使用 NIS over IPv6 RPC 传输机制来检索任何 NIS 数据。

DNS 的 IPv6 扩展

AAAA 资源记录是 IPv6 特定的资源记录，它已在 RFC 1886《DNS Extensions to Support IP Version 6》中指定。该 AAAA 记录将主机名映射到 128 位 IPv6 地址。IPv6 仍使用 PTR（指针）记录将 IP 地址映射为主机名。系统为 IPv6 地址保留了 128 位地址中的 32 个半字节（四位）。每个半字节都转换为与其相对应的十六进制 ASCII 值，然后再附加 `ip6.int`。

对 `nsswitch.conf` 文件的更改

对于 Solaris 10 11/06 及早期发行版，除了通过 `/etc/inet/ipnodes` 查找 IPv6 地址的功能之外，还向 NIS、LDAP 和 DNS 名称服务中添加了 IPv6 支持。因此，已经对 `nsswitch.conf` 文件进行了修改以支持 IPv6 查找功能。

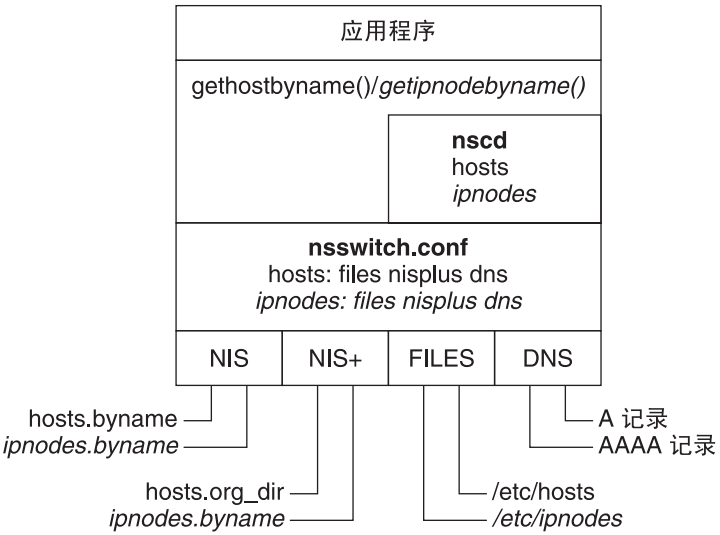
```
hosts:  files dns nisplus [NOTFOUND=return]
ipnodes: files dns nisplus [NOTFOUND=return]
```

注 – 在将 `/etc/nsswitch.conf` 文件更改为在多个名称服务中搜索 `ipnodes` 之前，请使用 IPv4 和 IPv6 地址填充这些 `ipnodes` 数据库。否则，不必要的延迟（包括可能的引导计时延迟）可能会导致解析主机地址。

下图显示了对于使用 `gethostbyname` 和 `getipnodebyname` 命令的应用程序，`nsswitch.conf` 文件和新名称服务数据库之间的新关系。斜体项为新

项。gethostbyname命令仅检查存储在 /etc/inet/hosts 中的 IPv4 地址。在 Solaris 10 11/06 及早期发行版中，getipnodebyname 命令会查阅在 nsswitch.conf 文件的 ipnodes 项中指定的数据库。如果查找失败，该命令会检查在 nsswitch.conf 文件中的 hosts 项中指定的数据库。

图 11–8 nsswitch.conf 和名称服务之间的关系



有关名称服务的更多信息，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

名称服务命令的更改

为了支持 IPv6，可以用现有的名称服务命令查找 IPv6 地址。例如，ypmatch 命令可用于新的 NIS 映射。nslookup 命令可以在 DNS 中查找新的 AAAA 记录。

NFS 和 RPC IPv6 支持

NFS 软件和远程过程调用 (Remote Procedure Call, RPC) 软件以无缝方式支持 IPv6。与 NFS 服务相关的现有命令没有任何改变。而且大多数 RPC 应用程序无需任何更改即可运行于 IPv6 上。某些涉及到传输的高级 RPC 应用程序可能需要进行更新。

IPv6 Over ATM（异步传输模式）支持

Oracle Solaris 支持 IPv6 over ATM、永久虚拟电路 (Permanent Virtual Circuit, PVC) 和静态交换式虚拟电路 (Switched Virtual Circuit, SVC)。

第 3 部分

DHCP

本部分介绍有关动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 的概念性信息，以及 DHCP 服务在规划、配置、管理和故障排除方面所涉及的任务。

◆ ◆ ◆ 第 12 章

关于 DHCP（概述）

本章介绍动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 并解释了该协议的基础概念，同时也说明了在网络中使用 DHCP 的优点。

本章包含以下信息：

关于 DHCP 协议

使用 DHCP 协议，可以使 TCP/IP 网络中的主机系统在引导时针对网络自动进行配置。DHCP 使用客户机/服务器机制。服务器为客户机存储和管理配置信息，并根据客户机的请求提供此信息。该信息中包含客户机的 IP 地址，以及有关客户机可使用的网络服务的信息。

DHCP 由早期的 BOOTP 协议发展而来，后者设计用于在 TCP/IP 网络上引导系统。对于客户机和服务器之间传送的消息，DHCP 使用与 BOOTP 相同的格式。然而，与 BOOTP 消息不同，DHCP 消息可包含客户机的网络配置数据。

DHCP 的主要优点是它能够通过租用来管理 IP 地址的指定。可通过**租用**回收未使用的 IP 地址。这些回收的 IP 地址可以重新指定给其他客户机。使用 DHCP 的站点所用的 IP 地址池小于为所有客户机指定永久性 IP 地址时所需的 IP 地址池。

使用 DHCP 的优势

在设置 TCP/IP 网络以及进行该网络的日常管理时，某些任务很耗时，而 DHCP 可以帮助您处理这些任务。请注意，在 Oracle Solaris 实现中，DHCP 只能与 IPv4 配合使用。

DHCP 拥有下列优点：

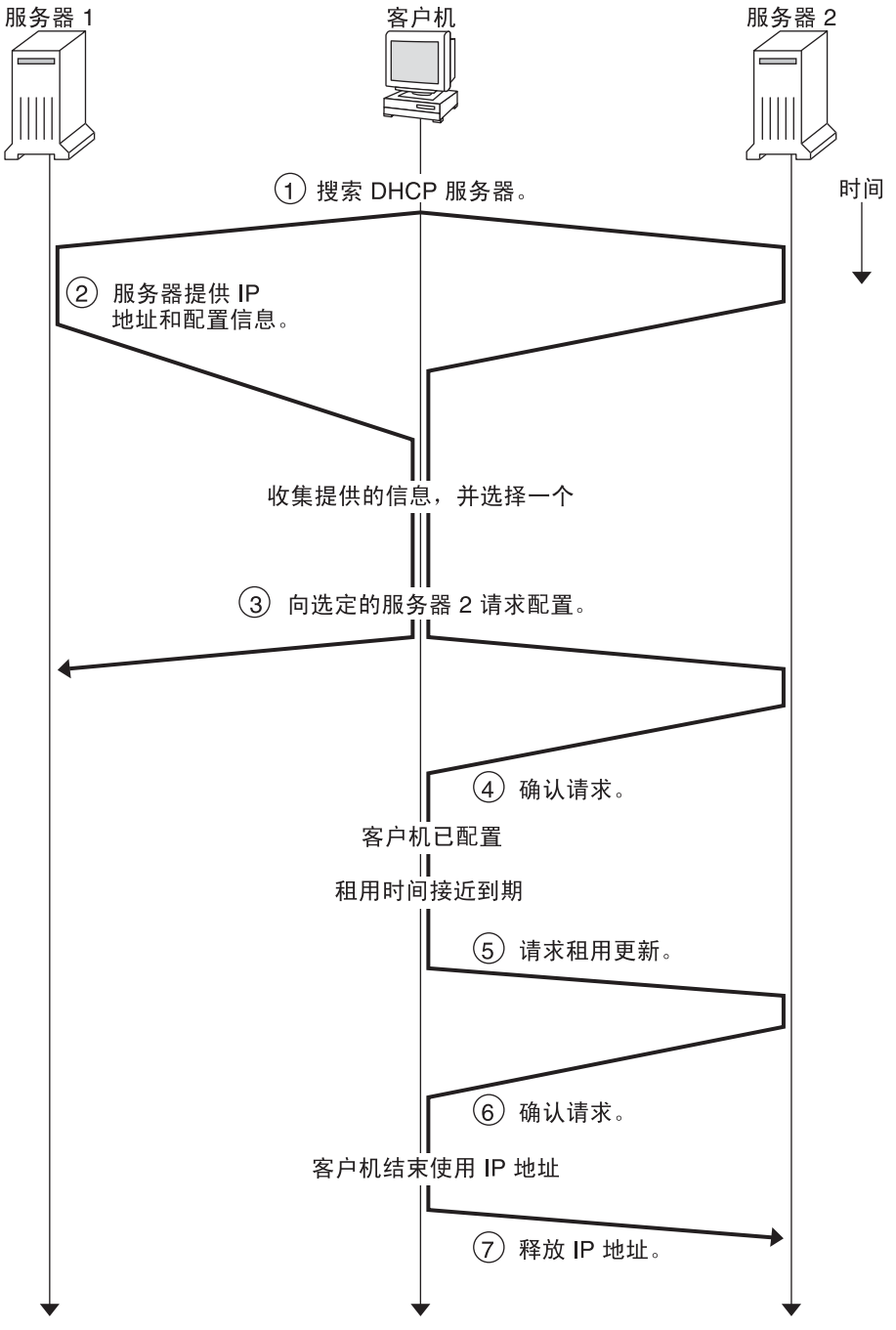
- **IP 地址管理**－DHCP 的一个主要优点是更易于管理 IP 地址。在不使用 DHCP 的网络中，您必须手动指定 IP 地址。您必须小心地为每台客户机指定唯一的 IP 地址并单独配置每台客户机。如果客户机移动到其他网络，您必须为该客户机执行手动修改。启用 DHCP 后，DHCP 服务器便会管理和指定 IP 地址，而无需管理员介入。客户机无需重新手动配置便可移动到其他子网，因为它们从 DHCP 服务器中获取了适用于新网络的新的客户机信息。
- **网络客户机集中配置**－您可以为某些客户机或某些类型的客户机创建定制的配置。配置信息存储在 DHCP 数据存储中的某个位置。您无需登录到客户机更改其配置。可以仅通过更改数据存储中的信息来更改多个客户机的配置。
- **支持 BOOTP 客户机**－BOOTP 服务器和 DHCP 服务器都可以侦听并响应来自客户机的广播。除了响应 DHCP 客户机之外，DHCP 服务器还可以响应来自 BOOTP 客户机的请求。BOOTP 客户机从服务器接收引导所需的 IP 地址和信息。
- **支持本地客户机和远程客户机**－BOOTP 提供了从一个网络到另一个网络的消息中继。DHCP 通过数种方法利用 BOOTP 的中继功能。大多数网络路由器可以配置为充当 BOOTP 中继代理角色，用于将 BOOTP 请求传送到不在客户机网络上的服务器。DHCP 请求也可以通过相同的方式转发，因为对于路由器而言，无法区别 DHCP 请求与 BOOTP 请求。当支持 BOOTP 中继的路由器不可用时，DHCP 服务器也可以配置为充当 BOOTP 中继代理的角色。
- **网络引导**－客户机可以使用 DHCP 而不是 RARP（Reverse Address Resolution Protocol，反向地址解析协议）和 `bootparams` 文件来获取从网络上的服务器进行引导所需的信息。DHCP 服务器可以为客户机提供运行所需的所有信息，包括 IP 地址、引导服务器和网络配置等信息。由于 DHCP 请求可以在子网间转发，因此，在使用 DHCP 网络引导时，可以减少在网络中部署的引导服务器的数量。RARP 引导要求每个子网都有一台引导服务器。
- **大型网络支持**－拥有数百万台 DHCP 客户机的网络可以使用 DHCP。DHCP 服务器使用多线程来同时处理大量客户机请求。该服务器也支持为处理大量数据而优化的数据存储。数据存储访问由单独的处理模块来处理。借助于这种数据存储方法，您可以添加对任何所需的数据库的支持。

DHCP 的工作原理

您必须先安装和配置 DHCP 服务器。在配置过程中，您需要指定客户机在网络上运行时所需的有关网络的信息。具备该信息后，客户机便可请求并接收网络信息。

下图显示了 DHCP 服务事件的序列。圆圈中的数字与图表之后的说明中的编号相对应。

图 12–1 DHCP 服务事件的序列



上图显示了以下步骤：

1. 客户机通过将**搜索消息**广播到本地子网上的有限广播地址 (255.255.255.255) 来搜索 DHCP 服务器。如果存在路由器并已将其配置为充当 BOOTP 中继代理的角色，请求便会传送到不同子网上的其他 DHCP 服务器。客户机的**广播**包括其唯一的 ID，在 Oracle Solaris 中的 DHCP 实现中，此 ID 由客户机的介质访问控制 (Media Access Control, MAC) 地址派生而来。在以太网上，MAC 地址与以太网地址相同。

   接收搜索消息的 DHCP 服务器可以通过查看以下信息来确定客户机的网络：

   - 请求来自哪个网络接口？服务器会确定客户机是位于通过接口连接的网络上，还是正在使用连接到该网络的 BOOTP 中继代理。
   - 请求中包含 BOOTP 中继代理的 IP 地址吗？当请求通过中继代理时，中继代理会将其地址插入到请求标头中。当服务器检测到**中继代理地址**时，服务器知道该地址的网络部分表示客户机的网络地址，这是因为中继代理必须连接到客户机的网络。
   - 客户机的网络有子网吗？服务器会查看 `netmasks` 表，来查找由中继代理地址或接收请求的网络接口地址指定的网络使用的子网掩码。服务器一旦知道了所用的子网掩码，就可以确定网络地址的哪一部分表示主机地址，然后可以选择适用于客户机的 IP 地址。有关 `netmasks` 的信息，请参见 `netmasks(4)` 手册页。

2. 在 DHCP 服务器确定客户机的网络之后，便会选择相应的 IP 地址并确认该地址尚未使用。然后，DHCP 服务器通过广播一条**提供消息**来响应客户机。该提供消息中包含选定的 IP 地址以及有关可为客户机配置的服务的信息。每台服务器都会暂时保留所提供的 IP 地址，直到客户机确定是否使用该 IP 地址为止。
3. 客户机会根据所提供的服务的数量和类型来选择最佳内容。客户机会广播一条请求来指定提供最佳内容的服务器的 IP 地址。该广播确保所有发出响应的 DHCP 服务器都知道客户机已经选择了一台服务器。未被选中的服务器会取消对之前提供的 IP 地址的保留。
4. 被选中的服务器会为客户机分配 IP 地址，并将信息存储到 DHCP 数据存储中。该服务器还会向客户机发送一条确认消息 (acknowledgement message, ACK)。**确认消息**包含客户机的网络配置参数。客户机使用 `ping` 实用程序测试此 IP 地址，以确保它没有被其他系统使用。然后，客户机会继续引导以加入网络。
5. 客户机会监视租用时间。当设定的时间段过去时，客户机会向所选的服务器发送一条新消息来延长租用时间。
6. 接收请求的 DHCP 服务器会延长租用时间，前提是租用仍然遵循由管理员设置的本地租用策略。如果服务器在 20 秒内没有响应，客户机便会广播一条请求，以便其他 DHCP 服务器之一可以延长租用期。
7. 当客户机不再需要 IP 地址时，便会通知服务器已释放了 IP 地址。此通知可以在正常关机时发送，也可以手动发送。

DHCP 服务器

DHCP 服务器在主机系统上作为 Oracle Solaris 中的守护进程运行。该服务器有两个基本功能：

- **管理 IP 地址**－DHCP 服务器控制一组 IP 地址，并将它们永久性地或在一段特定时间内分配给客户机。服务器使用租用机制来确定客户机可以使用非永久性地址的时间。当地址不再使用时，会返回到池中并可重新指定。服务器将 IP 地址与客户机的绑定信息保留在其 DHCP 网络表中，以确保地址不会被多台客户机使用。
- **为客户机提供网络配置**－DHCP 服务器指定 IP 地址，并提供网络配置的其他信息，例如主机名、广播地址、网络子网掩码、缺省网关、名称服务，以及可能的更多信息。这些网络配置信息从服务器的 `dhcptab` 数据库中获取。

DHCP 服务器也可配置为执行以下附加功能：

- **响应 BOOTP 客户机请求**－服务器会侦听 BOOTP 客户机所发出的搜索 BOOTP 服务器的广播，并为 BOOTP 客户机提供 IP 地址和引导参数。这些信息必须已经由管理员进行静态配置。DHCP 服务器可同时充当 BOOTP 服务器和 DHCP 服务器。
- **转发请求**－服务器会将 BOOTP 和 DHCP 请求转发到其他子网上的相应服务器。当服务器被配置为 BOOTP 中继代理时，便不能提供 DHCP 或 BOOTP 服务。
- **为 DHCP 客户机提供网络引导支持**－服务器可以为 DHCP 客户机提供通过网络引导所需的以下信息：IP 地址、引导参数和网络配置信息。服务器也可以提供 DHCP 客户机通过广域网 (Wide Area Network, WAN) 引导和安装所需的信息。
- **为提供主机名的客户机更新 DNS 表**－对于在其 DHCP 服务请求中提供 `Hostname` 选项和值的客户机，服务器可以代表它们尝试进行 DNS 更新。

DHCP 服务器管理

您能够以超级用户身份使用 DHCP 管理程序或命令行实用程序（如第 271 页中的 “DHCP 命令行实用程序”中所述），来启动、停止和配置 DHCP 服务器。通常，DHCP 服务器配置为在系统引导时自动启动，在系统关闭时自动停止。一般情况下，您无需手动启动和停止服务器。

DHCP 数据存储

DHCP 服务器使用的所有数据都保存在数据存储库内。数据存储可能包含纯文本文件、NIS+ 表或二进制格式文件。当配置 DHCP 服务时，请选择要使用的数据存储的类型。第 282 页中的 “选择 DHCP 数据存储”一节中介绍了数据存储类型之间的差异。您可以使用 DHCP 管理程序或 `dhcpconfig` 命令将数据存储从一种格式转换为另一种格式。

您还可以将数据从一台 DHCP 服务器的数据存储移动到另一台服务器的数据存储。您可以使用用于数据存储的导出和导入实用程序，即使服务器正在使用不同的数据存储格式时也是如此。借助 DHCP 管理程序或 `dhcpconfig` 命令，您可以导出和导入数据存储的全部内容或其中一部分内容。

注 – 如果您开发自己的代码模块来创建 DHCP（服务器和管理工具）和数据库之间的接口，则可以针对 DHCP 数据存储使用任何数据库或文件格式。有关更多信息，请参见《Solaris DHCP Service Developer's Guide》。

在 DHCP 数据存储库内有两种类型的表。您可以使用 DHCP 管理程序或命令行实用程序来查看和管理这些表的内容。数据表的类型如下：

- `dhcptab` **表** – 该表包含可传送给客户机的配置信息。
- **DHCP 网络表** – 该表包含驻留在表名所指定的网络上的 DHCP 客户机和 BOOTP 客户机的相关信息。例如，网络 `192.168.32.0` 可以有一个名称包括 `192_168_32_0` 的表。

dhcptab 表

`dhcptab` 表包含客户机可以从 DHCP 服务器获取的所有信息。DHCP 服务器在每次启动时都会扫描 `dhcptab` 表。`dhcptab` 表的文件名根据所用的数据存储的不同而异。例如，由 NIS+ 数据存储 `SUNWnisplus` 创建的 `dhcptab` 表为 `SUNWnisplus1_dhcptab`。

DHCP 协议定义了许多可传送到客户机的标准信息项。这些项被称为参数、符号或选项。在 DHCP 协议中，选项由数字代码和文本标签定义，但没有值。下表显示了一些常用的标准选项。

表 12–1 DHCP 标准选项样例

代码	标签	说明
1	`Subnet`	子网掩码 IP 地址
3	`Router`	路由器的 IP 地址
6	`DNSserv`	DNS 服务器的 IP 地址
12	`Hostname`	客户机主机名的文本字符串
15	`DNSdmain`	DNS 域名

当您在配置服务器的过程中提供信息时，会自动为一些选项指定值。您也可以在以后明确为其他选项指定值。各个选项及其值将被传送到客户机以提供配置信息。例如，选项/值对 `DNSdmain=Georgia.Peach.COM` 会将客户机的 DNS 域名设置为 `Georgia.Peach.COM`。

这些选项和其他选项可组合成名为**宏**的容器，此容器简化了将信息传送到客户机的过程。在服务器配置过程中会自动创建一些宏，这些宏包含在配置过程中被指定值的选项。宏也可以包含其他宏。

dhcptab 表的格式在 dhcptab(4) 手册页中进行了介绍。在 DHCP 管理程序中，在 "Options"（选项）和 "Macros"（宏）选项卡中显示的所有信息都来自 dhcptab 表。有关选项的更多信息，请参见第 273 页中的 “关于 DHCP 选项”。有关宏的更多信息，请参见第 273 页中的 “关于 DHCP 宏”。

请注意，不应手动编辑 dhcptab 表。您应该使用 dhtadm 命令或 DHCP 管理程序来创建、删除或修改选项和宏。

DHCP 网络表

DHCP 网络表将客户机标识符映射到 IP 地址和与每个地址关联的配置参数。网络表的格式在 dhcp_network(4) 手册页中进行了介绍。在 DHCP 管理程序中，在 "Addresses"（地址）选项卡中显示的所有信息都来自网络表。

DHCP 管理程序

DHCP 管理程序是可用于执行与 DHCP 服务关联的所有管理功能的图形用户界面 (Graphical User Interface, GUI) 工具。此工具除了用于管理服务器之外，还可用于管理服务器所用的数据。您必须以超级用户身份运行 DHCP 管理程序。

可以通过以下方法在服务器上使用 DHCP 管理程序：

- 配置和取消配置 DHCP 服务器
- 启动、停止和重新启动 DHCP 服务器
- 禁用和启用 DHCP 服务
- 定制 DHCP 服务器设置

使用 DHCP 管理程序，您可以通过以下方法管理 IP 地址、网络配置宏和网络配置选项：

- 在 DHCP 管理下添加和删除网络
- 在 DHCP 管理下查看、添加、修改、删除和释放 IP 地址
- 查看、添加、修改和删除网络配置宏
- 查看、添加、修改和删除非标准网络配置选项

DHCP 管理程序允许您通过以下方法管理 DHCP 数据存储：

- 将数据转换为一种新的数据存储格式
- 通过将 DHCP 数据从一台 DHCP 服务器导出并将其导入另一台服务器来在 DHCP 服务器之间移动该数据

DHCP 管理程序提供了使用此工具可执行的过程的全面联机帮助。有关更多信息，请参见第 300 页中的 “关于 DHCP 管理程序”。

DHCP 命令行实用程序

所有 DHCP 管理功能都可以通过命令行实用程序来执行。如果您以超级用户身份登录或以指定给 DHCP 管理配置文件的用户身份登录，就可以运行实用程序。请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

下表列出了实用程序，并说明了每个实用程序的用途。

表 12–2 DHCP 命令行实用程序

命令	说明和用途	手册页链接
`in.dhcpd`	DHCP 服务守护进程。使用命令行参数，您可以设置数个运行时选项。	`in.dhcpd(1M)`
`dhcpconfig`	用于配置和取消配置。使用该实用程序，您可以从命令行执行许多 DHCP 管理程序功能。该实用程序主要在需要自动执行某些配置功能的站点的脚本中使用。`dhcpconfig` 从服务器系统的网络拓扑文件中收集信息，来创建对初始配置有用的信息。	`dhcpconfig(1M)`
`dhtadm`	用于添加、删除和修改 DHCP 客户机的配置选项和宏。使用该实用程序，您可以间接编辑 `dhcptab` 表，以确保 `dhcptab` 表的格式正确。不能直接编辑 `dhcptab` 表。	`dhtadm(1M)`
`pntadm`	用于管理 DHCP 网络表。您可以使用该实用程序来执行以下任务： ■ 在 DHCP 管理下添加和删除 IP 地址和网络。 ■ 修改指定的 IP 地址的网络配置。 ■ 在 DHCP 管理下显示有关 IP 地址和网络的信息。	`pntadm(1M)`

基于角色的 DHCP 命令访问控制

`dhcpconfig`、`dhtadm` 和 `pntadm` 命令的安全性由基于角色的访问控制 (Role-Based Access Control, RBAC) 设置决定。在缺省情况下，这些命令只能由超级用户运行。如果您想在其他用户名下使用这些命令，必须按照第 303 页中的 “设置用户访问 DHCP 命令的权限”中所述，将该用户名指定到 DHCP 管理配置文件中。

DHCP 服务器配置

当您在要运行 DHCP 服务器的系统上首次运行 DHCP 管理程序时，需要对 DHCP 服务器进行配置。

DHCP 管理程序服务器配置对话框将提示您输入在网络上启用并运行 DHCP 服务器时所需的基本信息。某些缺省值可从现有的系统文件中获取。如果您没有为网络配置系统，就没有缺省值。DHCP 管理程序会提示输入以下信息：

- 服务器的角色，充当 DHCP 服务器或充当 BOOTP 中继代理
- 数据存储类型（文件、二进制文件、NIS+ 或某些特定于您的站点的类型）
- 所选数据存储类型的数据存储配置参数
- 用于更新主机记录的名称服务（如果有），例如 `/etc/hosts`、NIS+ 或 DNS
- 租用时间的长度及客户机是否能够更新租用
- DNS 服务器的 DNS 域名和 IP 地址
- 针对 DHCP 服务配置的第一个网络的网络地址和子网掩码
- 网络类型，局域网 (Local Area Network, LAN) 或点对点网络
- 特定路由器的路由器搜索或 IP 地址
- NIS 服务器的 NIS 域名和 IP 地址
- NIS+ 服务器的 NIS+ 域名和 IP 地址

您也可以使用 `dhcpconfig` 命令来配置 DHCP 服务器。该实用程序自动从现有的系统文件中收集信息来提供有用的初始配置。因此，您必须确保在运行 `dhcpconfig` 之前这些文件是正确的。有关 `dhcpconfig` 用来获取信息的各种文件的信息，请参见 `dhcpconfig(1M)` 手册页。

IP 地址分配

DHCP 服务器支持以下类型的 IP 地址分配：

- **手动分配**－服务器为特定的 DHCP 客户机提供您为其选择的特定 IP 地址。该地址不可回收或指定给其他客户机。
- **自动或永久性分配**－服务器提供没有截止时间的 IP 地址，使其与客户机永久性地关联，直到您更改了这种指定方式或客户机释放了该地址。
- **动态分配**－服务器向发出请求的客户机提供可租用特定一段时间的 IP 地址。当租用到期时，该地址可由服务器收回并可指定给其他客户机。具体期限由为服务器配置的租用时间决定。

网络配置信息

您需要确定要提供给 DHCP 客户机的信息。在配置 DHCP 服务器时，需要提供有关网络的基本信息。然后，您可以添加更多想要提供给客户机的信息。

DHCP 服务器以选项/值对和宏的形式将网络配置信息存储在 `dhcptab` 表中。选项是指您要提供给客户机的网络数据的关键字。值被指定到选项并以 DHCP 消息的形式传送到客户机。例如，NIS 服务器地址通过一个名为 `NISservs` 的选项传送。`NISservs` 选项有一个与 DHCP 服务器指定的 IP 地址列表相同的值。宏提供了一种将要提供给客户机

的任意数目的选项组合起来的便捷方法。您可以使用 DHCP 管理程序创建宏来组合选项并将值指定到选项。如果您更喜欢使用命令行工具，则可以通过 DHCP 配置表管理实用程序 dhtadm 来使用选项和宏。

关于 DHCP 选项

在 DHCP 中，**选项**是要传送到客户机的一条网络信息。DHCP 介绍中也将选项称为**符号**或**标记**。选项由数字代码和文本标签进行定义。在 DHCP 服务中使用选项时，便会为其赋值。

DHCP 协议为通常会指定的网络数据定义了大量的标准选项：Subnet、Router、Broadcst、NIS+dom、Hostname 和 LeaseTim 便是其中的几个示例。dhcp_inittab(4) 手册页中给出了标准选项的完整列表。您绝不能修改标准选项关键字。但是，您可以在将选项纳入宏时为与网络相关的选项指定值。

您可以为不是由标准选项表示的数据创建新的选项。您所创建的选项必须归为以下三种类别之一：

- **Extended**（扩展）－已成为标准 DHCP 选项但还没有纳入 DHCP 服务器实现的选项。如果您知道要使用的标准选项，但又不想升级您的 DHCP 服务器，可以使用扩展选项。
- **Site**（站点）－您的站点独有的选项。这些选项由您创建。
- **Vendor**（供应商）－只能应用于特定类别（如硬件或供应商平台）的客户机的选项。DHCP 实现包括大量用于 Oracle Solaris 客户机的供应商选项。例如，选项 SrootIP4 用于指定从网络引导的客户机应该用作其根 (/) 文件系统的服务器的 IP 地址。

第 15 章，管理 DHCP（任务）介绍了创建、修改和删除 DHCP 选项的过程。

关于 DHCP 宏

在 DHCP 服务中，**宏**是指网络配置选项以及您为这些选项指定的值的集合。创建宏是为了组合要传送到特定的客户机或客户机类型的选项。例如，专用于特定子网中所有客户机的宏可能包含子网掩码、路由器 IP 地址、广播地址、NIS+ 域和租用时间的选项/值对。

DHCP 服务器的宏处理

当 DHCP 服务器处理宏时，它将宏中定义的网络选项及值放在 DHCP 消息中传送给客户机。对于特定类型的客户机，服务器会自动处理一些宏。

要让服务器自动处理宏，宏的名称必须符合下表中所示的类别之一。

表 12–3 自动处理的 DHCP 宏类别

宏类别	说明
客户机类	宏名称与某类客户机相匹配，客户机类由客户机类型或操作系统分别指明或者由二者共同指明。例如，如果服务器有一个名为 SUNW.Sun-Blade-100 的宏，则所有硬件实现为 SUNW.Sun-Blade-100 的客户机都会自动收到 SUNW.Sun-Blade-100 宏中的值。
网络地址	宏名称与 DHCP 管理的网络 IP 地址相匹配。例如，如果服务器有一个名为 10.53.224.0 的宏，则所有连接到 10.53.224.0 网络的客户机都会自动收到 10.53.224.0 宏中的值。
客户机 ID	宏名称与客户机的某个唯一标识符相匹配，该标识符通常是从以太网地址或 MAC 地址中派生而来。例如，如果服务器有一个名为 08002011DF32 的宏，则客户机 ID 为 08002011DF32（从以太网地址 8:0:20:11:DF:32 中派生而来）的客户机会自动收到名为 08002011DF32 的宏中的值。

仅当以下条件之一成立时，才能处理其名称未使用表 12–3 中列出的任一类别的宏：

- 宏映射到 IP 地址。
- 宏包含在另一个自动处理的宏中。
- 宏包含在另一个映射到 IP 地址的宏中。

注 – 当配置服务器时，在缺省情况下会创建一个名称与服务器名称相匹配的宏。系统**不**会为任何客户机自动处理该服务器宏，因为没有使用可引发自动处理的任一名称类型对该客户机进行命名。以后在服务器上创建 IP 地址时，便会将这些 IP 地址映射为在缺省情况下使用该服务器宏。

宏处理的顺序

当 DHCP 客户机请求 DHCP 服务时，DHCP 服务器会确定与客户机相匹配的宏。服务器处理这些宏，并使用宏类别来确定处理的顺序。会首先处理最常见的类别，最后处理最特殊的类别。宏的处理顺序如下：

1. 客户机类宏－最常见的类别
2. 网络地址宏－比客户机类稍特殊
3. 映射到 IP 地址的宏－比网络地址宏稍特殊
4. 客户机 ID 宏－最特殊的类别，与某台客户机有关

包含在其他宏中的宏会作为容器宏的一部分来进行处理。

如果多个宏中包含同一选项，则会使用最特殊类别的宏中的该选项的值，因为它是最后处理的。例如，如果网络地址宏包含值为 24 小时的租用时间选项，而客户机 ID 宏包含值为 8 小时的租用时间选项，则客户机收到的租用时间为 8 小时。

DHCP 宏的大小限制

为宏中所有选项指定的值的总长度不得超过 255 个字节，包括选项代码和长度信息。此限制由 DHCP 协议指定。

最可能受此限制影响的是那些用于将路径传送到 Oracle Solaris 安装服务器上的文件的宏。一般情况下，应该能够传送所需的最小量的供应商信息。在需要输入路径名的选项中，应该使用简短的路径名。如果创建指向长路径的符号链接，则可以传送更简短的链接名。

DHCP 客户机

“客户机”一词有时用来指代在网络上充当客户机角色的物理计算机。但是，本文档中介绍的 DHCP 客户机是一种软件实体。DHCP 客户机是在配置为从 DHCP 服务器接收网络配置的系统上的 Oracle Solaris 中运行的守护进程 (`dhcpagent`)。其他供应商提供的 DHCP 客户机也可以使用 DHCP 服务器的服务。但是，本文档仅介绍 DHCP 客户机。

有关 DHCP 客户机的详细信息，请参见第 16 章，配置和管理 DHCP 客户机。

◆ ◆ ◆ 第 13 章

规划 DHCP 服务（任务）

您可以在正在创建的或已经存在的网络中使用 DHCP 服务。如果正在建立网络，请在尝试设置 DHCP 服务之前先阅读第 2 章，规划 TCP/IP 网络（任务）。如果网络已经存在，请继续阅读本章内容。

本章介绍在网络上设置 DHCP 服务之前需要执行的操作。虽然您还可以使用命令行实用程序 `dhcpconfig` 来设置 DHCP 服务，但是此信息用于 DHCP 管理程序。

本章包含以下信息：

为 DHCP 服务准备网络（任务列表）

在将网络设置为使用 DHCP 之前，必须收集信息以帮助您确定是配置一台还是多台服务器。使用下表中的任务列表来识别为使用 DHCP 服务而需要进行的网络准备工作。下表列出了各项任务、对任务内容的说明，以及对任务步骤进行详细描述的章节。

任务	说明	参考
映射网络拓扑。	确定并找到系统上可用的服务。	第 278 页中的“映射网络拓扑”
确定所需的 DHCP 服务器数量。	根据预期的 DHCP 客户机数量确定所需的 DHCP 服务器数量。	第 279 页中的“确定DHCP 服务器的数量”

任务	说明	参考
更新系统文件和 netmasks 表。	准确反映网络拓扑。	第 279 页中的 “更新系统文件和网络掩码表”

映射网络拓扑

如果尚未映射网络拓扑，则应映射网络的物理结构。指明路由器和客户机的位置，以及提供网络服务的服务器的位置。此网络拓扑图可以帮助您确定用于提供 DHCP 服务的服务器。该图还可帮助确定 DHCP 服务器可以提供给客户机的配置信息。

有关规划网络的更多信息，请参见第 2 章，规划 TCP/IP 网络（任务）。

DHCP 配置过程可以从服务器的系统和网络文件中收集一些网络信息。第 279 页中的 “更新系统文件和网络掩码表”中介绍了这些文件。但是，您可能需要为客户机提供其他必须输入服务器宏的服务信息。检查网络拓扑时，请记录需要客户机识别的任何服务器的 IP 地址。例如，以下服务器可能会在系统上提供服务。DHCP 配置不会搜索这些服务器。

- 时间服务器
- 日志服务器
- 打印服务器
- 安装服务器
- 引导服务器
- Web 代理服务器
- 交换服务器
- X 窗口字体服务器
- 简单文件传输协议 (Trivial File Transfer Protocol, TFTP) 服务器

要避免的网络拓扑

在一些 IP 网络环境中，多个局域网 (Local Area Network, LAN) 共享同一网络硬件介质。这些网络可能会使用多个网络硬件接口或多个逻辑接口。DHCP 在这种共享介质网络中不能很好地工作。当多个 LAN 在同一物理网络上运行时，DHCP 客户机的请求便会发送到所有网络硬件接口。这种效果就好像将客户机同时连接到所有 IP 网络上一样。

DHCP 必须能够确定客户机的网络地址，以便为客户机指定一个适当的 IP 地址。如果在硬件介质上提供多个网络，则服务器无法确定客户机的网络。服务器在不知道网络号的情况下不能指定 IP 地址。

可以仅在其中一个网络上使用 DHCP。如果一个网络不能满足您的 DHCP 需求，则必须重新配置网络。您应该考虑以下建议：

- 在子网上使用一个可变长度子网掩码 (Variable Length Subnet Mask, VLSM)，以便更好地利用现有的 IP 地址空间。您可能无需在同一物理网络上运行多个网络。有关实现可变长度子网划分的信息，请参见 `netmasks(4)` 手册页。有关无类域间路由 (Classless Inter-Domain Routing, CIDR) 和 VLSM 的更多详细信息，请参见`http://www.ietf.org/rfc/rfc1519.txt`。
- 将交换机上的端口配置为将设备指定到不同的物理 LAN。此技术可根据 DHCP 的需要保留 LAN 到 IP 网络的一对一映射。有关端口配置的信息，请参见有关交换机的文档。

确定DHCP 服务器的数量

您选择的数据存储选项会直接影响支持 DHCP 客户机所必需的服务器数量。下表显示了针对每种数据存储，一台 DHCP 服务器可以支持的 DHCP 和 BOOTP 客户机的最大数量。

表 13–1 一台 DHCP 服务器支持的客户机最大估计数量

数据存储类型	支持的客户机最大数量
文本文件	10,000
NIS+	40,000
二进制文件	100,000

此最大数量是一个一般原则，不是绝对数量。一台 DHCP 服务器的客户机容量很大程度上依赖于服务器每秒必须处理的事务数量。租用时间和使用模式对事务处理率有重要影响。例如，假定租用设置为 12 小时并且用户在夜间关闭系统。如果许多用户在早上同一时间打开系统，服务器就会遇到事务处理高峰，因为许多客户机会同时请求租用。在这种环境中，DHCP 服务器只能支持较少的客户机；而在租用时间较长的环境或者由持续连接的设备（如电缆调制解调器）构成的环境中，DHCP 服务器可以支持更多客户机。

第 282 页中的 “选择 DHCP 数据存储”一节对各种数据存储类型进行了比较。

更新系统文件和网络掩码表

在 DHCP 配置过程中，DHCP 工具会扫描服务器上的各种系统文件，以查找可用于配置服务器的信息。

在运行 DHCP 管理程序或 dhcpconfig 配置服务器之前，必须确保系统文件中的信息是最新的。如果在配置服务器之后发现错误，请使用 DHCP 管理程序或 dhtadm 修改服务器上的宏。

下表列出了在 DHCP 服务器配置过程中收集到的一些信息及信息来源。请确保在服务器上配置 DHCP 之前已经正确设置了此信息。如果配置服务器之后对系统文件进行更改，则应重新配置服务以反映这些更改。

表 13–2 用于 DHCP 配置的信息

信息	源	注释
时区	系统日期、时区设置	在安装 Oracle Solaris 的过程中初始设置的日期和时区。您可以使用 date 命令更改日期。您可以通过将 svc:/system/environment:init SMF 服务中的 timezone/localtime 属性设置为 TZ 环境变量来更改时区。有关更多信息，请参见 TIMEZONE(4) 手册页。
DNS 参数	/etc/resolv.conf	DHCP 服务器使用 /etc/resolv.conf 文件获取 DNS 参数，如 DNS 域名和 DNS 服务器地址。有关 resolv.conf 的更多信息，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》或 resolv.conf(4) 手册页。
NIS 或 NIS+ 参数	系统域名、nsswitch.conf、NIS 或 NIS+	DHCP 服务器使用 domainname 命令获取服务器系统的域名。nsswitch.conf 文件可指示服务器在何处查找基于域的信息。如果服务器系统是 NIS 或 NIS+ 客户机，则 DHCP 服务器会执行查询以获取 NIS 或 NIS+ 服务器的 IP 地址。有关更多信息，请参见 nsswitch.conf(4) 手册页。
缺省路由器	系统路由表、用户提示符	DHCP 服务器搜索网络路由列表来为连接到本地网络的客户机查找缺省路由器。如果客户机不在同一网络上，则 DHCP 服务器一定会提示您有关信息。
子网掩码	网络接口、netmasks 表	DHCP 服务器查看自己的网络接口来确定本地客户机的网络掩码和广播地址。如果请求已由中继代理转发，则服务器会在中继代理的网络上获取 netmasks 表中的子网掩码。

表 13–2 用于 DHCP 配置的信息 （续）

信息	源	注释
广播地址	网络接口、netmasks 表	对于本地网络，DHCP 服务器通过查询网络接口来获取广播地址。对于远程网络，服务器使用 BOOTP 中继代理的 IP 地址和远程网络的网络掩码来计算网络的广播地址。

为 DHCP 服务器配置做出决定（任务列表）

本节介绍在网络上配置第一台 DHCP 服务器之前要做出的一些决定。下表将指导您做出配置网络以使用 DHCP 所需的决定，并为每项任务提供相关链接，使其跳转至说明执行每项任务的步骤的章节。

任务	说明	参考
为 DHCP 选择一台服务器。	决定服务器是否满足系统运行 DHCP 服务的要求。	第 281 页中的 “选择运行 DHCP 服务的主机”
选择数据存储。	比较数据存储类型，确定最适用于站点的数据存储。	第 282 页中的 “选择 DHCP 数据存储”
设置租用策略。	了解 IP 地址租用，以帮助确定适用于站点的租用策略。	第 282 页中的 “设置租用策略”
选择路由器地址或路由器搜索。	确定 DHCP 客户机是使用路由器搜索还是使用特定的路由器。	第 283 页中的 “确定用于 DHCP 客户机的路由器”

选择运行 DHCP 服务的主机

了解网络拓扑之后，可以根据以下系统要求来选择设置 DHCP 服务器的主机。

主机必须满足以下要求：

- 主机必须运行 Solaris 2.6 发行版或更高版本。如果需要支持大量客户机，则必须安装 Solaris 8 7/01 发行版或更高版本。
- 所有包含计划使用 DHCP 的客户机的网络都必须能够访问主机，无论是直接在网络上访问还是通过 BOOTP 中继代理访问。
- 必须将主机配置为使用路由。
- 主机必须具有可以反映网络拓扑的正确配置的 netmasks 表。

选择 DHCP 数据存储

您可以选择在文本文件、二进制文件或 NIS+ 目录服务中存储 DHCP 数据。下表总结了每种数据存储类型的功能，并指明了使用每种数据存储类型的环境。

表 13–3 DHCP 数据存储的比较

数据存储类型	性能	维护	共享	环境
二进制文件	高性能、高容量	低维护成本，无需数据库服务器。必须使用 DHCP 管理程序或 `dhtadm` 和 `pntadm` 来查看内容。建议对常规文件进行备份。	不能在 DHCP 服务器之间共享数据存储。	包含许多网络的中型到大型环境，其中每个网络都包含数以千计的客户机。适用于小型到中型 ISP。
NIS+	中等性能和容量，依赖于 NIS+ 服务的性能和容量	DHCP 服务器系统必须配置为 NIS+ 客户机。需要对 NIS+ 服务进行维护。必须使用 DHCP 管理程序或 `dhtadm` 和 `pntadm` 来查看内容。建议使用 `nisbackup` 进行常规备份。	DHCP 数据分布在 NIS+ 中，并且多台服务器可以访问相同容器。	小型到中型环境，其中每个网络最多可包含 5000 台客户机。
文本文件	中等性能、低容量	低维护成本，无需数据库服务器。ASCII 格式是可读格式，不需要使用 DHCP 管理程序、`dhtadm` 或 `pntadm` 来查看。建议对常规文件进行备份。	DHCP 服务器之间可以共享数据存储，前提是 DHCP 数据存储在一个通过 NFS 挂载点导出的文件系统上。	包含少于 10,000 台客户机的小型环境，其中每个网络包含数百台到一千台客户机。

由于 NIS 不支持快速增量更新，因此不会提供传统的 NIS 作为数据存储选项。如果网络使用的是 NIS，则应使用文本文件或二进制文件进行数据存储。

设置租用策略

租用期可指定 DHCP 服务器允许 DHCP 客户机使用特定 IP 地址的时间。在服务器的初始配置过程中，必须指定站点范围的租用策略。**租用策略**可指明租用时间，并指定客户机是否可以续订其租约。服务器会使用您提供的信息设置缺省宏中的选项值，这些宏在配置过程中由服务器创建。可以通过在创建的配置宏中设置选项来为特定的客户机或客户机类型设置不同的租用策略。

租用时间以小时数、天数或星期数的形式指定租用有效的时间。为客户机指定 IP 地址或者重新协商 IP 地址的租用时，会计算租用的失效日期和时间。同时还会将租用时间的小时数添加到客户机的 DHCP 确认的时间戳中。例如，假定 DHCP 确认的时间戳为

2005 年 9 月 16 日上午 9:15，并且租用时间为 24 小时。本示例中的租用截止时间为 2005 年 9 月 17 日上午 9:15。租用截止时间存储在客户机的 DHCP 网络记录中，可以使用 DHCP 管理程序或通过 `pntadm` 实用程序查看该时间。

租用时间值应该相对较小，以便快速回收过期地址，另外，该值还应该足够大，以使租用时间长于 DHCP 服务中断的时间。修复运行 DHCP 服务的系统后，客户机应当能够正常工作。一般原则是指定的时间应该是预期的系统停机时间的两倍。例如，如果需要四个小时来获取和更换故障部件并重新引导系统，请将租用时间指定为八个小时。

租用协商选项可确定客户机是否可以在租用过期之前与服务器重新协商租用。如果允许协商租用，则客户机会跟踪租用中剩余的时间。当租用时间经过一半时，客户机会请求 DHCP 服务器将租用延长到最初的租用时间。在系统多于 IP 地址的环境中，应禁用租用协商。然后，对于 IP 地址的使用强制执行时间限制。如果 IP 地址足够多，则应启用租用协商，以避免在租用到期时强制客户机关闭网络接口。如果使客户机获取新的租用，则客户机的 TCP 连接（如 NFS 和 telnet 会话）可能会中断。可以在服务器配置过程中针对所有客户机启用租用协商。您可以通过使用配置宏中的 `LeaseNeg` 选项针对特定的客户机或客户机类型启用租用协商。

注 – 在网络上提供服务的系统应保留它们的 IP 地址。这类系统不应受短期租用的限制。如果为这类系统指定保留的手动 IP 地址而不是永久租用的 IP 地址，则可以在这些系统中使用 DHCP。然后，可以检测何时不再使用系统的 IP 地址。

确定用于 DHCP 客户机的路由器

主机系统使用路由器来进行本地网络之外的任何网络通信。主机必须知道这些路由器的 IP 地址。

配置 DHCP 服务器时，必须通过两种方法之一为 DHCP 客户机提供路由器地址。其中一种方法是为路由器提供特定的 IP 地址。但是，首选方法是指定客户机应通过路由器搜索协议来查找路由器。

如果网络上的客户机可以执行路由器搜索，则即使只有一个路由器，也应该使用路由器搜索协议。通过路由器搜索，客户机可轻松适应网络中路由器的变化。例如，假定一个路由器出现故障并由具有新地址的路由器取代。客户机可以自动搜索新地址，而不必获取新的网络配置以得到新路由器地址。

为 IP 地址管理做出决定（任务列表）

作为 DHCP 服务设置的一部分，需要确定服务器要管理的 IP 地址的几个方面。如果网络需要多台 DHCP 服务器，可将某些 IP 地址的任务指定给每一台服务器。必须决定如何划分地址的责任。下表是一个任务列表，描述在网络上使用 DHCP 时管理 IP 地址的任务。该表还包括指向详细说明如何执行每项任务的相关章节的链接。

任务	说明	参考
指定服务器应管理的地址。	确定希望 DHCP 服务器管理的地址及其数量。	第 284 页中的 “IP 地址的数目和范围”
决定服务器是否应自动为客户机生成主机名。	了解如何生成客户机主机名，以便决定是否生成主机名。	第 284 页中的 “生成客户机主机名”
确定要指定给客户机的配置宏。	了解客户机配置宏，以便为客户机选择一个合适的宏。	第 285 页中的 “缺省客户机的配置宏”
确定要使用的租用类型。	了解租用类型，以帮助确定最适合您的 DHCP 客户机的类型。	第 285 页中的 “动态和永久租用类型”

IP 地址的数目和范围

在服务器的初始配置过程中，可以使用 DHCP 管理程序，通过指定块中的地址总数和第一个地址来添加一个 IP 地址块（或称地址范围）。DHCP 管理程序可根据此信息添加一个连续地址的列表。如果有几个非连续地址块，则可以通过在初始配置之后再次运行 DHCP 管理程序的地址向导来添加其他地址。

在配置 IP 地址之前，需要了解待添加地址的初始块中的地址数目，以及地址范围中第一个地址的 IP 地址。

生成客户机主机名

DHCP 的动态性质意味着 IP 地址不会与使用它的系统的主机名永久关联。如果选择此选项，DHCP 管理工具可生成一个客户机名称来与每个 IP 地址关联。客户机名称由一个前缀（或称作根名称）、一个连字符和一个由服务器指定的编号组成。例如，如果根名称为 `charlie`，则客户机名称为 `charlie-1`、`charlie-2`、`charlie-3`，依此类推。

缺省情况下，生成的客户机名称以管理客户机的 DHCP 服务器的名称开头。此策略在包含多台 DHCP 服务器的环境下非常有用，因为您可以在 DHCP 网络表中快速查看任何给定的 DHCP 服务器所管理的客户机。不过，可以将根名称更改为您所选择的任何名称。

配置 IP 之前，请确定是否需要 DHCP 管理工具生成客户机名称，如果是，请再确定要使用的根名称。

如果指定在配置 DHCP 的过程中注册主机名，则可以将生成的客户机名称映射到 `/etc/inet/hosts` 、DNS 或 NIS+ 中的 IP 地址。有关更多信息，请参见第 315 页中的 “客户机主机名注册”。

缺省客户机的配置宏

在 DHCP 中，**宏**是指网络配置选项及其指定值的集合。DHCP 服务器使用宏来确定要发送到 DHCP 客户机的网络配置信息。

配置 DHCP 服务器时，管理工具将通过两种方法收集信息：一种是从系统文件中收集，另一种是通过用户指定的提示符或命令行选项直接从用户处收集。使用此信息，管理工具可创建以下宏：

- **网络地址宏** – 网络地址宏的名称与客户机网络的 IP 地址相匹配。例如，如果网络为 `192.68.0.0`，则网络地址宏的名称也是 `192.68.0.0`。该宏包含作为网络的一部分的任何客户机所需的信息，如子网掩码、网络广播地址、缺省路由器或路由器搜索标记，以及 NIS/NIS+ 域和服务器等。此外，可能还包括其他适用于您的网络的选项。对于位于该网络上的所有客户机，会自动处理网络地址宏，如第 274 页中的 “宏处理的顺序”中所述。
- **语言环境宏** – 语言环境宏的名称为 `Locale`。该宏包含指定本地时区时相对于世界标准时间 (UTC) 的偏移（以秒为单位）。语言环境宏不会自动进行处理，但会包含在服务器宏中。
- **服务器宏** – 服务器宏的名称与服务器的主机名相匹配。例如，如果服务器的名称是 `pineola`，则服务器宏的名称也是 `pineola`。服务器宏包含有关租用策略、时间服务器、DNS 域和 DNS 服务器的信息，并且还可能包含配置程序可从系统文件中获取的其他信息。服务器宏包含语言环境宏，因此 DHCP 服务器会将语言环境宏作为服务器宏的一部分来处理。

  配置第一个网络的 IP 地址时，必须为使用正在配置的地址的所有 DHCP 客户机选择一个要使用的客户机配置宏。您选择的宏会映射到 IP 地址。缺省情况下，会选择服务器宏，因为该宏包含所有使用此服务器的客户机所需的信息。

客户机会先接收包含在网络地址宏中的选项，然后再接收映射到 IP 地址的宏中的选项。此处理顺序使服务器宏中的选项优先级高于网络地址宏中的任何冲突的选项。有关宏处理顺序的更多信息，请参见第 274 页中的 “宏处理的顺序”。

动态和永久租用类型

租用类型可确定租用策略是否适用于您所配置的 IP 地址。在服务器的初始配置过程中，可以通过 DHCP 管理程序为正在添加的地址选择动态或永久租用。如果使用 `dhcpconfig` 命令配置 DHCP 服务器，则租用为动态租用。

如果某个 IP 地址具有**动态租用**，则 DHCP 服务器可管理该地址。DHCP 服务器可以将 IP 地址分配给客户机、延长租用时间、检测地址不再使用的时间以及回收该地址。如

果某个 IP 地址具有**永久租用**，则 DHCP 服务器只能分配该地址。然后，客户机将一直拥有该地址，直到将其明确释放为止。释放该地址时，服务器可将其指定给另一台客户机。只要将地址配置为永久租用类型，它即可不受租用策略的限制。

配置 IP 地址范围时，选择的租用类型会应用于此范围内的所有地址。要最大限度地利用 DHCP，应针对大多数地址使用动态租用。如有必要，可随后将各个地址修改为永久租用。但是，应该将永久租用的总数保持在最小程度。

保留的 IP 地址和租用类型

可以通过将 IP 地址手动指定给特定的客户机来保留它们。保留的地址既可以与永久租用关联，也可以与动态租用关联。将保留的地址指定给一个永久租用时，以下语句成立：

- 该地址只能分配给与其绑定的客户机。
- DHCP 服务器不能将该地址分配给另一台客户机。
- DHCP 服务器不能回收该地址。

如果将保留的地址指定给动态租用，则只能将该地址分配给与其绑定的客户机。但是，客户机必须像地址未保留那样跟踪租用时间并协商延长租用期。使用此策略，您可以通过查看网络表来跟踪客户机使用地址的时间。

在初始配置过程中，不能为所有 IP 地址创建保留的地址。保留的地址少量地用于单个地址。

规划多台 DHCP 服务器

如果要配置多台 DHCP 服务器来管理 IP 地址，请考虑以下几条原则：

- 划分 IP 地址池，以便每台服务器负责一个地址范围，并且责任没有重叠。
- 如果 NIS+ 可用，请选择其作为数据存储。如果不可用，请选择文本文件并为数据存储的绝对路径指定一个共享目录。不能共享二进制文件数据存储。
- 分别配置每台服务器，以便正确分配地址所有权，并且可自动创建基于服务器的宏。
- 将服务器设置为按指定间隔扫描 `dhcptab` 表中的选项和宏，以便服务器可以使用最新信息。可以使用 DHCP 管理程序安排自动读取 `dhcptab`，如第 316 页中的 “定制 DHCP 服务器的性能选项”中所述。
- 确保所有客户机均可以访问所有 DHCP 服务器，以便服务器可以互相支持。当拥有客户机地址的服务器无法访问时，具有有效 IP 地址租用的客户机可能会尝试检验其配置或延长租用期。当一台客户机尝试连接主服务器的时间达到 20 秒时，另一台服务器便会响应此客户机。如果客户机请求一个特定的 IP 地址，并且拥有该地址的服务器不可用，则其他服务器将处理此请求。在这种情况下，客户机不会接收请求的地址。客户机将接收一个由做出响应的 DHCP 服务器所拥有的 IP 地址。

规划远程网络的 DHCP 配置

完成 DHCP 的初始配置之后，即可将 IP 地址放入 DHCP 管理的远程网络中。但是，由于系统文件不是服务器的本地文件，并且 DHCP 管理程序和 `dhcpconfig` 无法查找提供缺省值的信息，因此必须提供此信息。尝试配置远程网络之前，请确保您了解以下信息：

- 远程网络的 IP 地址。
- 远程网络的子网掩码。此信息可从名称服务中的 `netmasks` 表中获取。如果网络使用本地文件，请查看网络中某个系统上的 `/etc/netmasks`。如果网络使用的是 NIS+，请使用 `niscat netmasks.org_dir` 命令。如果网络使用的是 NIS+，请使用 `ypcat -k netmasks.byaddr` 命令。请确保 `netmasks` 表中包含要管理的所有子网的全部拓扑信息。
- 网络类型。客户机通过局域网 (Local Area Network, LAN) 连接或点对点协议 (Point-to-Point Protocol, PPP) 连接到网络。
- 路由信息。客户机是否可以使用路由器搜索？如果不能，则必须确定客户机可使用的路由器的 IP 地址。
- NIS 域和 NIS 服务器（如果适用）。
- NIS+ 域和 NIS+ 服务器（如果适用）。

有关添加 DHCP 网络的过程，请参见第 320 页中的 “添加 DHCP 网络”。

选择用于配置 DHCP 的工具

收集信息和规划 DHCP 服务之后，即可准备配置 DHCP 服务器。您可以使用 DHCP 管理程序或命令行实用程序 `dhcpconfig` 配置服务器。使用 DHCP 管理程序，可以选择选项并指定之后用于创建 DHCP 服务器所用的 `dhcptab` 和网络表的数据。`dhcpconfig` 实用程序要求使用命令行选项指定数据。

DHCP 管理程序功能

DHCP 管理程序是一种基于 Java™ 技术的 GUI 工具，可提供 DHCP 配置向导。第一次在未配置为 DHCP 服务器的系统上运行 DHCP 管理程序时，配置向导会自动启动。DHCP 配置向导提供了一系列对话框，提示您配置服务器所需的基本信息：数据存储格式、租用策略、DNS/NIS/NIS+ 服务器和域，以及路由器地址。向导从系统文件中获取某些信息，您只需确认信息是否正确，或者在需要时更正信息。

完成对话框并确认信息之后，DHCP 服务器守护进程便会在服务器系统上启动。然后，系统将提示您启动 "Add Addresses Wizard"（添加地址向导）来配置网络的 IP 地址。最初仅会为 DHCP 配置服务器的网络，并且为其他服务器选项指定缺省值。完成初始配置之后，可以再次运行 DHCP 管理程序，以添加网络并修改其他服务器选项。

有关 DHCP 配置向导的更多信息，请参见第 289 页中的 “使用 DHCP 管理程序来配置和取消配置 DHCP 服务器”。有关 DHCP 管理程序的更多详细信息，请参见第 300 页中的 “关于 DHCP 管理程序”。

dhcpconfig 功能

dhcpconfig 实用程序支持可用于配置和取消配置 DHCP 服务器，转换为新的数据存储以及从其他 DHCP 服务器中导入/导出数据的选项。使用 dhcpconfig 实用程序配置 DHCP 服务器时，此实用程序可从第 279 页中的 “更新系统文件和网络掩码表”中介绍的系统文件中获取信息。您不能像使用 DHCP 管理程序那样来查看和确认从系统文件中获取的信息。因此，在运行 dhcpconfig 之前更新系统文件非常重要。您也可以使用命令行选项覆盖 dhcpconfig 在缺省情况下从系统文件中获取的值。dhcpconfig 命令可以在脚本中使用。有关更多信息，请参见 dhcpconfig(1M) 手册页。

DHCP 管理程序与 dhcpconfig 的比较

下表总结了两种服务器配置工具之间的差异。

表 13–4 DHCP 管理程序与 dhcpconfig 命令的比较

功能	DHCP 管理程序	带有选项的 dhcpconfig
从系统中收集的网络信息。	允许您查看从系统文件中收集的信息，并在需要时更改此信息。	您可以使用命令行选项指定网络信息。
配置速度。	通过省略不必要的服务器选项的提示并改用缺省值来加快配置过程。您可以在完成初始配置之后更改不必要的选项。	最快的配置过程，但是您可能需要指定许多选项值。

第 14 章，配置 DHCP 服务（任务）介绍了通过 DHCP 管理程序或 dhcpconfig 实用程序来配置服务器时可使用的过程。

◆ ◆ ◆ 第 14 章

配置 DHCP 服务（任务）

当您在网络上配置 DHCP 服务时，需要配置并启动第一台 DHCP 服务器。其他 DHCP 服务器可以在以后添加，而且如果数据存储支持共享数据，这些 DHCP 服务器可以从共享位置访问相同的数据。本章介绍有关配置 DHCP 服务器并将网络及其关联 IP 地址纳入 DHCP 管理的任务，同时还介绍了取消配置 DHCP 服务器的方法。

每个任务都包括一个帮助您在 DHCP 管理程序下执行该任务的过程和一个使用 `dhcpconfig` 实用程序执行等效任务的过程。本章包含以下信息：

- 第 289 页中的 “使用 DHCP 管理程序来配置和取消配置 DHCP 服务器”
- 第 295 页中的 “使用 `dhcpconfig` 命令来配置和取消配置 DHCP 服务器”

如果您在配置 DHCP 服务时遇到困难，请参见第 17 章，对 DHCP 问题进行故障排除（参考）。

在配置 DHCP 服务后，请参见第 15 章，管理 DHCP（任务）以获取有关管理 DHCP 服务的信息。

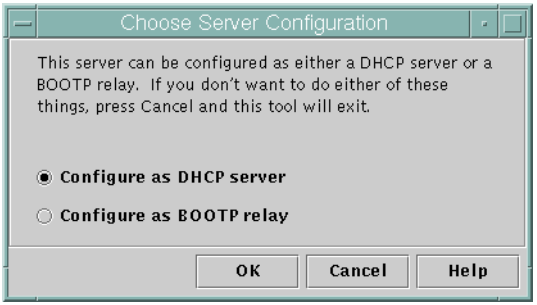
使用 DHCP 管理程序来配置和取消配置 DHCP 服务器

本节介绍通过 DHCP 管理程序来帮助您配置和取消配置 DHCP 服务器的过程。请注意，您必须运行 X Window 系统（例如，CDE 或 GNOME）才能使用 DHCP 管理程序。

可以超级用户身份使用 `/usr/sadm/admin/bin/dhcpmgr` 命令来运行 DHCP 管理程序。有关实用程序的常规信息，请参见第 300 页中的 “关于 DHCP 管理程序”。有关运行 DHCP 管理程序的更多详细信息，请参见第 305 页中的 “如何启动和停止 DHCP 服务（DHCP 管理程序）”。

当您在没有配置 DHCP 的服务器上运行 DHCP 管理程序时，屏幕上会显示以下信息。您可以指定是要配置 DHCP 服务器还是要配置 BOOTP 中继代理。

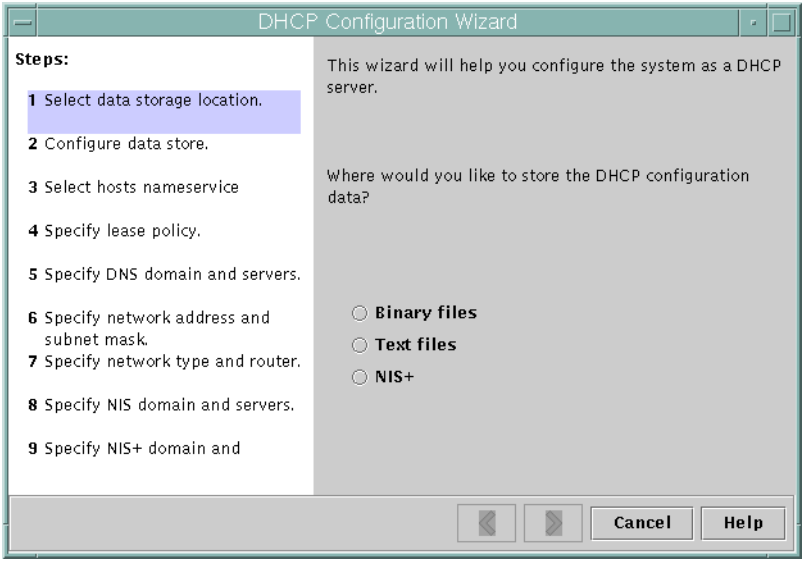
图 14–1 DHCP 管理程序中的 "Choose Server Configuration"（选择服务器配置）对话框



配置 DHCP 服务器

在配置 DHCP 服务器时，DHCP 管理程序会启动 "DHCP Configuration Wizard"（DHCP 配置向导）来提示您提供配置服务器所需的信息。下图显示了该向导的初始屏幕。

图 14–2 "DHCP Configuration Wizard"（DHCP 配置向导）的初始屏幕



完成对向导提示的回答时，DHCP 管理程序会创建下表列出的项。

表 14–1 在 DHCP 服务器配置过程中创建的项

项	说明	内容
服务配置文件 `/etc/inet/dhcpsvc.conf`	记录服务器配置选项的关键字和值。	数据存储类型、位置以及与 `in.dhcpd` 一起用来在引导系统时启动 DHCP 守护进程的选项。请勿手动编辑此文件。必须使用 `dhcpmgr` 或 `dhcpconfig` 来修改 DHCP 配置信息。
`dhcptab` 表	DHCP 管理程序会在 `dhcptab` 表不存在的情况下创建该表。	具有指定值的宏和选项。
名为 `Locale` 的语言环境宏（可选）	包含本地时区相对于标准时间 (Universal time, UTC) 的偏移（以秒为单位）。	具有指定秒数的 `UTCoffst` 选项。
服务器宏（其名称与服务器的节点名称相匹配）	包含由配置 DHCP 服务器的管理员的输入来决定值的选项。这些选项适用于所有使用该服务器所拥有的地址的客户机。	`Locale` 宏以及以下选项： ■ `Timeserv`，设置为指向服务器的主 IP 地址。 ■ `LeaseTim`，设置为租用的秒数。 ■ `LeaseNeg`，前提是您选择了可协商租用。 ■ `DNSdmain` 和 `DNSserv`，前提是配置了 DNS。 ■ `Hostname`，**不得**为它指定值。该选项的出现表明了主机名必须从名称服务中获取。
网络地址宏（其名称与客户机网络的网络地址相同）	包含由配置 DHCP 服务器的管理员的输入来决定值的选项。这些选项适用于所有驻留在宏名称所指定的网络上的客户机。	包括以下选项： ■ `Subnet`，设置为本地子网的子网掩码 ■ `Router`，设置为路由器的 IP 地址，或 `RDiscvyF`，让客户机可以使用路由器搜索功能 ■ `Broadcst`，设置为广播 IP 地址。该选项仅在网络不是点对点网络时出现。 ■ `MTU`，即最大传输单元 ■ `NISdmain` 和 `NISservs`，前提是配置了 NIS ■ `NIS+dom` 和 `NIS+serv`，前提是配置了 NIS+
网络的网络表	在您为该网络创建 IP 地址之前，会创建一个空白的表。	添加 IP 地址后才会有内容。

▼ 如何配置 DHCP 服务器（DHCP 管理程序）

开始之前 请确保在配置 DHCP 服务器之前已阅读第 13 章，规划 DHCP 服务（任务）。特别是，应参阅第 281 页中的“为 DHCP 服务器配置做出决定（任务列表）”中的说明执行下列任务：

- 选择要用作 DHCP 服务器的系统。
- 做出有关数据存储、租用策略和路由器信息的决定。

1 以超级用户的身份登录服务器系统。

2 **启动 DHCP 管理程序。**

```
#/usr/sadm/admin/bin/dhcpmgr &
```

3 **选择 "Configure as DHCP Server"（配置为 DHCP 服务器）选项。**

将启动 "DHCP Configuration Wizard"（DHCP 配置向导），帮助您配置服务器。

4 **根据在规划阶段所做的决定来选择选项或键入所需的信息。**

如果有问题，请单击向导窗口中的 "Help"（帮助）来打开 Web 浏览器并显示 "DHCP Configuration Wizard"（DHCP 配置向导）的帮助信息。

5 **在结束指定所需信息时，请单击 "Finish"（完成）来结束服务器配置。**

6 **在 "Start Address Wizard"（启动地址向导）提示下，单击 "Yes"（是）来配置服务器的 IP 地址。**

使用 "Add Addresses to Network"（添加地址到网络）向导，您可以指定纳入 DHCP 控制的地址。

7 **按照您在规划阶段所作的决定来回答提示。**

有关更多信息，请参见第 284 页中的 “为 IP 地址管理做出决定（任务列表）”。如果有问题，请单击向导窗口中的 "Help"（帮助）来打开 Web 浏览器并显示 "Add Addresses to Network"（添加地址到网络）向导的帮助信息。

8 **检查您的选择，然后单击 "Finish"（完成），将 IP 地址添加到网络表中。**

网络表将根据您所指定范围内的每个地址的记录进行更新。

另请参见 您可以使用 "Network Wizard"（网络向导）将更多的网络添加到 DHCP 服务器，如第 320 页中的 “添加 DHCP 网络”中所述。

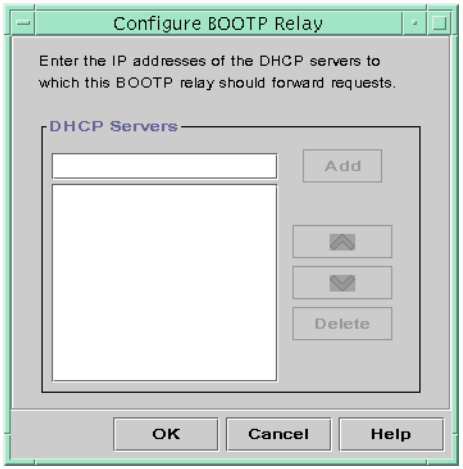
配置 BOOTP 中继代理

在配置 BOOTP 中继代理时，DHCP 管理程序将执行以下操作：

- 提示您输入向其转发请求的一台或多台 DHCP 服务器的 IP 地址。
- 存储 BOOTP 中继服务所需的设置

下图显示了在选择配置 BOOTP 中继代理时的屏幕显示。

图 14–3 DHCP 管理程序中的 "Configure BOOTP Relay"（配置 BOOTP 中继）对话框



▼ 如何配置 BOOTP 中继代理（DHCP 管理程序）

开始之前 请确保在配置 BOOTP 中继代理之前已阅读第 13 章，规划 DHCP 服务（任务）。特别是，应参阅第 281 页中的“选择运行 DHCP 服务的主机”中的说明选择要使用的系统。

1. **以超级用户的身份登录服务器系统。**

2. **启动 DHCP 管理程序。**

   ```
   #/usr/sadm/admin/bin/dhcpmgr &
   ```

   如果系统没有配置为 DHCP 服务器或 BOOTP 中继代理，将启动 "DHCP Configuration Wizard"（DHCP 配置向导）。如果系统已经配置为 DHCP 服务器，则必须首先取消配置服务器。请参见第 294 页中的“取消配置 DHCP 服务器和 BOOTP 中继代理”。

3. **选择 "Configure as BOOTP Relay"（配置为 BOOTP 中继）。**

   将打开 "Configure BOOTP Relay"（配置 BOOTP 中继）对话框。

4. **键入一台或多台 DHCP 服务器的 IP 地址或主机名，然后单击 "Add"（添加）。**

   指定的 DHCP 服务器必须配置为可以处理该 BOOTP 中继代理接收的 BOOTP 或 DHCP 请求。

5. **单击 "OK"（确定）退出对话框。**

   请注意，DHCP 管理程序只提供用于退出应用程序的 "File"（文件）菜单和用于管理服务器的 "Service"（服务）菜单。禁用的菜单选项仅在 DHCP 服务器上有用。

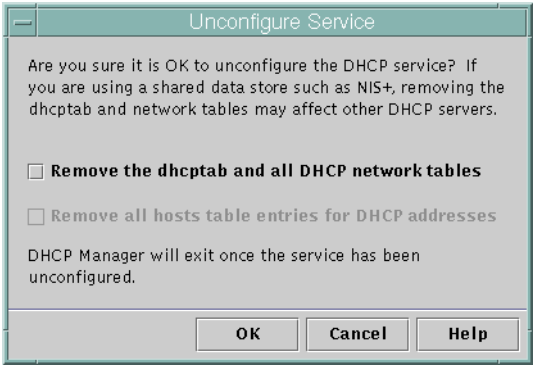
取消配置 DHCP 服务器和 BOOTP 中继代理

在取消配置 DHCP 服务器或 BOOTP 中继代理时，DHCP 管理程序将执行以下操作：

- 停止 DHCP 守护进程 (`in.dhpcd`)
- 删除 `/etc/inet/dhcpsvc.conf` 文件，该文件记录了有关守护进程启动和数据存储位置的信息

下图显示了在选择取消配置 DHCP 服务器时的屏幕显示。

图 14–4 DHCP 管理程序中的 "Unconfigure Service"（取消配置服务）对话框



已取消配置的服务器上的 DHCP 数据

在取消配置 DHCP 服务器时，您必须决定如何处理 `dhcptab` 表和 DHCP 网络表。如果数据在服务器之间共享，则不能删除 `dhcptab` 和 DHCP 网络表。如果删除这些表，便无法在网络上使用 DHCP。数据可通过 NIS+ 或在导出的本地文件系统上共享。文件 `/etc/inet/dhcpsvc.conf` 记录了所用的数据存储及其位置。

您可以取消配置 DHCP 服务器，但按原样保留数据，方法是不选择任何删除数据的选项。如果您取消配置服务器并按原样保留数据，则会禁用 DHCP 服务器。

如果您希望其他 DHCP 服务器拥有这些 IP 地址，则必须将 DHCP 数据移动到该 DHCP 服务器。必须在取消配置当前服务器之前移动该数据。有关更多信息，请参见第 367 页中的 “在 DHCP 服务器之间移动配置数据（任务列表）”。

如果您确定要删除该数据，则可以选择一个选项来删除 `dhcptab` 和网络表。如果生成了 DHCP 地址的客户机名，您仍可以选择从主机表中删除那些项。客户机名项可以从 DNS、`/etc/inet/hosts` 或 NIS+ 中删除。

在取消配置 BOOTP 中继代理之前，请确保没有依赖于该代理向 DHCP 服务器转发请求的客户机。

▼ 如何取消配置 DHCP 服务器或 BOOTP 中继代理（DHCP 管理程序）

1. **成为超级用户。**
2. **启动 DHCP 管理程序。**

   ```
   #/usr/sadm/admin/bin/dhcpmgr &
   ```

3. **在 "Service"（服务）菜单中选择 "Unconfigure"（取消配置）。**

   将显示 "Unconfigure Service"（取消配置服务）对话框。如果服务器为 BOOTP 中继代理，则可以使用对话框确认取消配置中继代理。如果服务器为 DHCP 服务器，则必须决定如何处理 DHCP 数据并在对话框中做出选择。请参见图 14–4。

4. **可选选择选项删除数据。**

   如果服务器使用通过 NIS+ 共享的数据，或使用通过 NFS 共享的文件中的共享数据，则不能选择任何删除数据的选项。如果服务器没有使用共享数据，则可选择两个选项中的一个或两个全选来删除数据。

   有关删除数据的更多信息，请参见第 294 页中的 “已取消配置的服务器上的 DHCP 数据”。

5. **单击 "OK"（确定）取消配置服务器。**

   将关闭 "Unconfigure Service"（取消配置服务）对话框和 DHCP 管理程序。

使用 dhcpconfig 命令来配置和取消配置 DHCP 服务器

本节介绍使用 dhcpconfig 和命令行选项来帮助您配置和取消配置 DHCP 服务器或 BOOTP 中继代理的过程。

▼ 如何配置 DHCP 服务器 (dhcpconfig -D)

开始之前 请确保在配置 DHCP 服务器之前已阅读第 13 章，规划 DHCP 服务（任务）。特别是，应参阅第 281 页中的 “为 DHCP 服务器配置做出决定（任务列表）”中的说明执行下列任务：

- 选择要用作 DHCP 服务器的系统。
- 做出有关数据存储、租用策略和路由器信息的决定。

1. **登录到要进行 DHCP 服务器配置的系统。**

2. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

3. **通过键入以下格式的命令来配置 DHCP 服务器：**

   #/usr/sbin/dhcpconfig -D -r *datastore* **-p** *location*

   datastore 为以下类型之一： `SUNWfiles`、`SUNWbinfiles` 或 `SUNWnisplus`。

   location 为与数据存储有关的位置，您要在此处存储 DHCP 数据。对于 `SUNWfiles` 和 `SUNWbinfiles`，该位置必须是绝对路径名。对于 `SUNWnisplus`，该位置必须是完全指定的 NIS+ 目录。

   例如，您可能键入以下类似命令：

   ```
   dhcpconfig -D -r SUNWbinfiles -p /var/dhcp
   ```

   `dhcpconfig` 实用程序使用主机的系统文件和网络文件来确定用于配置 DHCP 服务器的值。有关可用于覆盖缺省值的其他 `dhcpconfig` 命令选项的信息，请参见 `dhcpconfig(1M)` 手册页。

4. **向 DHCP 服务添加一个或多个网络。**

   有关添加网络的过程，请参见第 322 页中的 “如何添加 DHCP 网络 (dhcpconfig)”。

▼ 如何配置 BOOTP 中继代理 (dhcpconfig -R)

开始之前 根据第 281 页中的 “选择运行 DHCP 服务的主机”中列出的要求选择要用作 BOOTP 中继代理的系统。

1. **登录到要配置为 BOOTP 中继代理的服务器。**

2. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

3. **通过键入以下格式的命令来配置 BOOTP 中继代理：**

   # /usr/sbin/dhcpconfig -R *server-addresses*

为要将请求转发到的 DHCP 服务器指定一个或多个 IP 地址。如果您指定多个地址，请使用逗号分隔这些地址。

例如，您可能键入以下类似命令：

```
/usr/sbin/dhcpconfig -R 192.168.1.18,192.168.42.132
```

▼ 如何取消配置 DHCP 服务器或 BOOTP 中继代理 (dhcpconfig -U)

1. **登录到您要取消配置的 DHCP 服务器或 BOOTP 中继代理系统。**

2. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

3. **取消配置 DHCP 服务器或 BOOTP 中继代理：**

   ```
   # /usr/sbin/dhcpconfig -U
   ```

   如果服务器不使用共享数据，也可使用 -x 选项来删除 dhcptab 和网络表。如果服务器使用共享数据，则不要使用 -x 选项。-h 选项可用于从主机表中删除主机名。有关 dhcpconfig 选项的更多信息，请参见 dhcpconfig(1M) 手册页。

   有关删除数据的更多信息，请参见第 294 页中的 “已取消配置的服务器上的 DHCP 数据”。

◆ ◆ ◆ 第 15 章

15 管理 DHCP（任务）

本章介绍的任务将会对您管理 DHCP 服务有所帮助，其中包括针对服务器、BOOTP 中继代理和客户机的任务。每个任务都包含一个帮助您在 DHCP 管理程序中执行任务的过程，以及一个使用 DHCP 命令行实用程序执行等效任务的过程。手册页中更全面地介绍了这些 DHCP 命令行实用程序。

在使用本章之前，您应该已完成了 DHCP 服务和初始网络的初始配置。第 14 章，配置 DHCP 服务（任务）介绍了 DHCP 配置。

本章包含以下信息：

关于 DHCP 管理程序

DHCP 管理程序是一种图形用户界面 (Graphical User Interface, GUI) 工具，可用于执行针对 DHCP 服务的管理任务。

DHCP 管理程序窗口

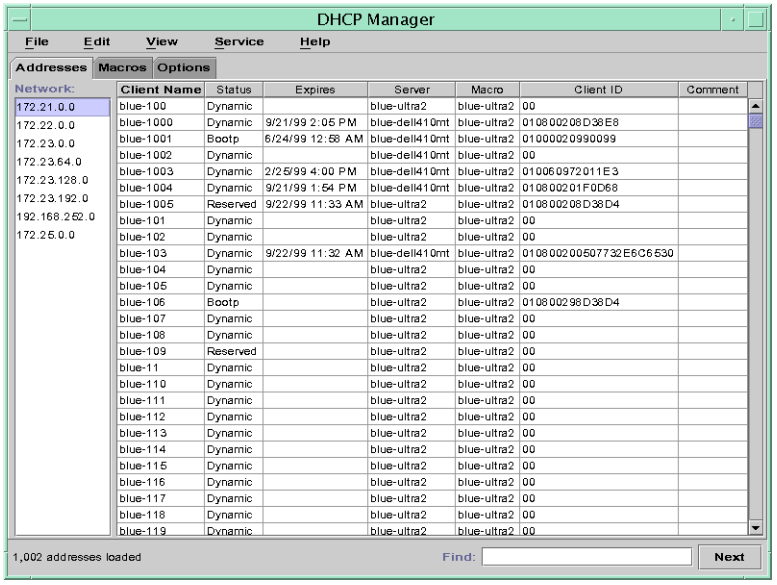
DHCP 管理程序窗口的外观取决于在运行 DHCP 管理程序的系统上配置 DHCP 服务器的方式。

将系统配置为 DHCP 服务器后，DHCP 管理程序可使用基于选项卡的窗口。您可以针对要处理的信息类型选择一个选项卡。DHCP 管理程序包含以下选项卡：

- **Addresses**（地址）选项卡－列出由 DHCP 管理的所有网络和 IP 地址。通过 "Addresses"（地址）选项卡，您可以处理网络和 IP 地址；可以单个或成批地添加或删除项；还可以修改各个网络或 IP 地址的属性，或者同时对一批地址执行相同的属性修改。启动 DHCP 管理程序时，首先会打开 "Addresses"（地址）选项卡。
- **Macros**（宏）选项卡－列出 DHCP 配置表 (`dhcptab`) 中的所有可用宏以及宏中包含的选项。通过 "Macros"（宏）选项卡，您可以创建或删除宏，还可以通过添加选项并为选项提供值来修改宏。
- **Options**（选项）选项卡－列出已针对此 DHCP 服务器定义的所有选项。此选项卡上列出的选项并不是在 DHCP 协议中定义的标准选项，而是标准选项的扩展，并分为 "Extended"（扩展）、"Vendor"（供应商）或 "Site"（站点）类。由于不能以任何方式对标准选项进行更改，因此该选项卡上不会列出标准选项。

下图显示了在启动 DHCP 服务器上的 DHCP 管理程序时，DHCP 管理程序窗口可能的外观。

图 15–1 DHCP 服务器系统上的 DHCP 管理程序



将服务器配置为 BOOTP 中继代理之后，DHCP 管理程序窗口便不会显示这些选项卡。BOOTP 中继代理不需要相同的信息。您只能使用 DHCP 管理程序来修改 BOOTP 中继代理的属性以及停止或启动 DHCP 守护进程。下图显示了在配置为 BOOTP 中继代理的系统上，DHCP 管理程序的可能外观。

图 15–2 BOOTP 中继代理上的 DHCP 管理程序

DHCP 管理程序菜单

DHCP 管理程序菜单包括以下各项：

- **File**（文件）－退出 DHCP 管理程序。
- **Edit**（编辑）－针对网络、地址、宏和选项执行管理任务。
- **View**（视图）－更改当前选定的选项卡的外观。

- **Service**（服务）－管理 DHCP 守护进程和数据存储。
- **Help**（帮助）－打开 Web 浏览器并显示 DHCP 管理程序的帮助信息。

当 DHCP 管理程序在 BOOTP 中继代理上运行时，"Edit"（编辑）和 "View"（视图）菜单处于禁用状态。

所有的 DHCP 管理任务都通过 "Edit"（编辑）和 "Service"（服务）菜单完成。

可以使用 "Edit"（编辑）菜单中的命令，在选定的选项卡中创建、删除和修改项。这些项可以包括网络、地址、宏和选项。选择 "Addresses"（地址）选项卡之后，"Edit"（编辑）菜单还会列出向导。这些向导由多个对话框集组成，可帮助您创建多个网络和 IP 地址。

"Service"（服务）菜单列出可用于管理 DHCP 守护进程的命令。通过 "Service"（服务）菜单，您可以执行以下任务：

- 启动和停止 DHCP 守护进程。
- 启用和禁用 DHCP 守护进程。
- 修改服务器配置。
- 取消配置服务器。
- 转换数据存储。
- 在服务器上导出和导入数据。

启动和停止 DHCP 管理程序

您必须以超级用户的身份在 DHCP 服务器系统上运行 DHCP 管理程序。如果您必须从远程运行 DHCP 管理程序，则可以使用 X 窗口远程显示功能将显示内容发送到您的系统。

▼ 如何启动和停止 DHCP 管理程序

1. **以超级用户的身份登录 DHCP 服务器系统。**

2. **可选如果您是远程登录到 DHCP 服务器系统，请按如下方式在本地系统上显示 DHCP 管理程序。**

   a. **在本地系统上键入以下命令：**

      ```
      # xhost +server-name
      ```

   b. **在远程 DHCP 服务器系统上键入以下命令：**

      ```
      # DISPLAY=local-hostname;export DISPLAY
      ```

3. **启动 DHCP 管理程序。**

   ```
   # /usr/sadm/admin/bin/dhcpmgr &
   ```

将打开 DHCP 管理程序窗口。如果服务器配置为 DHCP 服务器，则此窗口将显示 "Addresses"（地址）选项卡。如果服务器配置为 BOOTP 中继代理，则此窗口将不显示任何选项卡。

4 **要停止 DHCP 管理程序，请从 "File"（文件）菜单中选择 "Exit"（退出）。**

将关闭 DHCP 管理程序窗口。

设置用户访问 DHCP 命令的权限

缺省情况下，只有 `root` 用户才能执行 `dhcpconfig`、`dhtadm` 和 `pntadm` 命令。如果您希望非 `root` 用户使用这些命令，可以针对这些命令设置基于角色的访问控制 (Role-Based Access Control, RBAC)。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

以下手册页也可能对您非常有用：`rbac(5)`、`exec_attr(4)` 和 `user_attr(4)`。

以下过程说明如何指定DHCP 管理配置文件，用户可通过此配置文件执行 DHCP 命令。

▼ 如何授予用户访问 DHCP 命令的权限

1 **以超级用户的身份登录 DHCP 服务器系统。**

2 **将用户或角色添加到 `/etc/user_attr` 文件。**

编辑文件 `/etc/user_attr` 以添加如下格式的项。针对每个应该管理 DHCP 服务的用户或角色添加一项。

username`::::type=normal;profiles=DHCP Management`

例如，对于用户 `ram`，将添加以下项：

```
ram::::type=normal;profiles=DHCP Management
```

DHCP 服务器任务

▼ 如何配置 ISC DHCP 服务器

可以按照以下步骤对 ISC DHCP 服务器进行初始配置。

1. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **编辑 DHCP 配置文件。**

   创建 `/etc/dhcp/dhcpd4.conf` 或 `/etc/dhcp/dhcpd6.conf` 文件。有关更多信息，请参见 `dhcpd.conf`(5) 手册页。

3. **启用需要的服务。**

   ```
   # svcadm enable service
   ```

   service 可以为下列值之一：

`svc:/network/dhcp/server:ipv4`	提供来自 IPv4 客户机的 DHCP 和 BOOTP 请求
`svc:/network/dhcp/server:ipv6`	提供来自 IPv6 客户机的 DHCP 和 BOOTP 请求
`svc:/network/dhcp/relay:ipv4`	将来自 IPv4 客户机的 DHCP 和 BOOTP 请求中继到包含 DHCP 服务器的网络
`svc:/network/dhcp/relay:ipv6`	将来自 IPv6 客户机的 DHCP 和 BOOTP 请求中继到包含 DHCP 服务器的网络

▼ 如何修改 DHCP 服务的配置

1. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **编辑 DHCP 配置文件。**

   编辑 `/etc/dhcp/dhcpd4.conf` 或 `/etc/dhcp/dhcpd6.conf` 文件。有关更多信息，请参见 `dhcpd.conf`(5) 手册页。

3. **刷新 SMF 数据。**

   ```
   # svcadm refresh service
   ```

启动和停止 DHCP 服务

本节介绍如何使用 DHCP 管理程序和 dhcpconfig 命令来启动和停止 DHCP 服务。还可以使用服务管理工具 (Service Management Facility, SMF) 命令来启动和停止 DHCP 服务。有关将 SMF 命令用于 DHCP 服务的更多信息，请参见第 307 页中的 “DHCP 服务和服务管理工具”。

启动和停止 DHCP 服务包含几种级别的操作，您可以执行这些操作来影响 DHCP 守护进程的运行。您必须了解每种操作的含义，以便选择正确的过程来获取所需的结果。这些操作的术语如下：

- **启动、停止和重新启动命令**仅影响当前会话的守护进程。例如，如果您停止 DHCP 服务，则守护进程会终止，但在重新引导系统时重新启动。停止 DHCP 服务时，DHCP 数据表不会受到影响。您可以使用 DHCP 管理程序或 SMF 命令来临时启动和停止 DHCP 服务，而无需启用和禁用此服务。
- **启用和禁用命令**影响当前会话和将来会话的守护进程。如果禁用 DHCP 服务，则当前运行的守护进程将终止，并且不会在重新引导服务器时启动。必须使 DHCP 守护进程在系统引导时自动启动。DHCP 数据表不会受到影响。您可以使用 DHCP 管理程序、dhcpconfig 命令或 SMF 命令来启用和禁用 DHCP 服务。
- **取消配置命令**可关闭守护进程、防止守护进程在系统重新引导时启动，以及用于删除 DHCP 数据表。您可以使用 DHCP 管理程序或 dhcpconfig 命令来取消配置 DHCP 服务。取消配置在第 14 章，配置 DHCP 服务（任务）中有介绍。

注 – 如果一个服务器具有多个网络接口，但是您不希望在所有网络上都提供 DHCP 服务，请参见第 318 页中的 “指定 DHCP 监视的网络接口”。

以下过程可帮助您启动、停止、启用和禁用 DHCP 服务。

▼ 如何启动和停止 DHCP 服务（DHCP 管理程序）

1. **以超级用户的身份登录 DHCP 服务器系统。**
2. **启动 DHCP 管理程序。**

   ```
   # /usr/sadm/admin/bin/dhcpmgr &
   ```

3. **选择以下操作之一：**

   - **从 "Service"（服务）菜单中选择 "Start"（启动）以启动 DHCP 服务。**

- **从 "Service"（服务）菜单中选择 "Stop"（停止）以停止 DHCP 服务。**
  DHCP 守护进程将停止，直到重新启动它或重新引导系统。
- **从 "Service"（服务）菜单中选择 "Restart"（重新启动）以停止并立即重新启动 DHCP 服务。**

▼ 如何启用和禁用 DHCP 服务（DHCP 管理程序）

- **在 DHCP 管理程序中，选择以下操作之一：**
  - **从 "Service"（服务）菜单中选择 "Enable"（启用），以便将 DHCP 守护进程配置为在系统引导时自动启动。**
    DHCP 服务将在启用之后立即启动。
  - **从 "Service"（服务）菜单中选择 "Disable"（禁用）以防止 DHCP 守护进程在系统引导时自动启动。**
    DHCP 服务将在禁用之后立即停止。

▼ 如何启用和禁用 DHCP 服务 (dhcpconfig -S)

1. **登录到 DHCP 服务器系统。**
2. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**
   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。
3. **选择以下操作之一：**
   - **要启用 DHCP 服务，请键入以下命令：**
     ```
     # /usr/sbin/dhcpconfig -S -e
     ```
   - **要禁用 DHCP 服务，请键入以下命令：**
     ```
     # /usr/sbin/dhcpconfig -S -d
     ```

DHCP 服务和服务管理工具

服务管理工具 (Service Management Facility, SMF) 在《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”中有介绍。可以使用 SMF `svcadm` 命令来启用和启动 DHCP 服务器，以及禁用和停止 DHCP 服务器。但是，不能使用 SMF 命令修改可以使用 DHCP 工具设置的 DHCP 服务选项。特别是，不能使用 SMF 工具来设置 `/etc/dhcp/dhcpsvc.conf` 文件中存储的服务选项。

下表显示 DHCP 命令与等效的 SMF 命令之间的映射关系。

表 15–1 用于 DHCP 服务器任务的 SMF 命令

任务	DHCP 命令	SMF 命令
启用 DHCP 服务	`dhcpconfig -S -e`	`svcadm enable svc:/network/dhcp-server`
禁用 DHCP 服务	`dhcpconfig -S -d`	`svcadm disable svc:/network/dhcp-server`
仅针对当前会话启动 DHCP 服务	无	`svcadm enable -t svc:/network/dhcp-server`
停止当前会话的 DHCP 服务	无	`svcadm disable -t svc:/network/dhcp-server`
重新启动 DHCP 服务	`dhcpconfig -S -r`	`svcadm restart svc:/network/dhcp-server`

修改 DHCP 服务选项（任务列表）

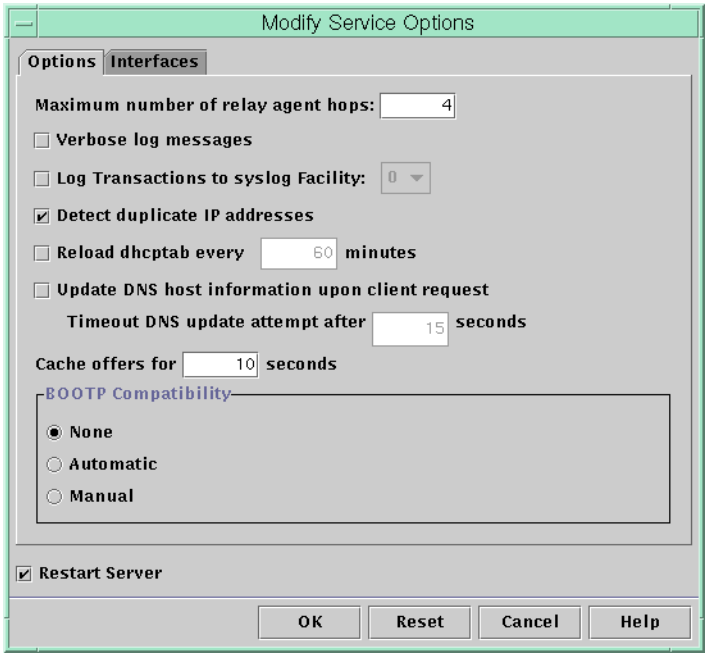
您可以更改 DHCP 服务的某些附加功能的值，使用 DHCP 管理程序进行初始配置时可能未提供这些功能。要更改服务选项，您可以使用 DHCP 管理程序中的 "Modify Service Options"（修改服务选项）对话框。或者，可以使用 `dhcpconfig` 命令指定选项。

以下是描述修改 DHCP 服务选项的任务列表。此任务列表包括指向完成每项任务的过程的链接。

任务	说明	参考
更改日志选项。	启用或禁用日志，并选择一个用于记录 DHCP 事务的 `syslog` 工具。	第 310 页中的“如何生成详细的 DHCP 日志消息（DHCP 管理程序）” 第 311 页中的“如何生成详细的 DHCP 日志消息（命令行）” 第 311 页中的“如何启用和禁用 DHCP 事务日志（DHCP 管理程序）” 第 312 页中的“如何启用和禁用 DHCP 事务日志（命令行）” 第 312 页中的“如何将 DHCP 事务记录到单独的 `syslog` 文件中”
更改 DNS 更新选项。	启用或禁用服务器的功能，以便针对提供主机名的客户机动态添加 DNS 项。确定服务器在尝试更新 DNS 时应花费的最长时间。	第 314 页中的“如何针对 DHCP 客户机启用动态 DNS 更新”
启用或禁用重复 IP 地址检测功能。	启用或禁用 DHCP 服务器的功能，以便在为客户机提供 IP 地址之前确定此 IP 地址尚未使用。	第 316 页中的“如何定制 DHCP 性能选项（DHCP 管理程序）” 第 317 页中的“如何定制 DHCP 性能选项（命令行）”
更改 DHCP 服务器用于读取配置信息的选项。	启用或禁用按指定间隔自动读取 `dhcptab` 的功能，或者更改读取间隔。	第 316 页中的“如何定制 DHCP 性能选项（DHCP 管理程序）” 第 317 页中的“如何定制 DHCP 性能选项（命令行）”
更改中继代理点数/代理数。	增加或减少在 DHCP 守护进程删除请求之前此请求可经过的网络数。	第 316 页中的“如何定制 DHCP 性能选项（DHCP 管理程序）” 第 317 页中的“如何定制 DHCP 性能选项（命令行）”
更改对提供的 IP 地址进行高速缓存的时间长度。	增加或减少 DHCP 服务在将所提供的 IP 地址提供给新客户机之前保留此地址的秒数。	第 316 页中的“如何定制 DHCP 性能选项（DHCP 管理程序）” 第 317 页中的“如何定制 DHCP 性能选项（命令行）”

下图显示了 DHCP 管理程序的 "Modify Service Options"（修改服务选项）对话框。

图 15–3 DHCP 管理程序中的 "Modify Service Options"（修改服务选项）对话框



更改 DHCP 日志选项

DHCP 服务可以将 DHCP 服务消息和 DHCP 事务记录到 `syslog` 中。有关 `syslog` 的更多信息，请参见 `syslogd(1M)` 和 `syslog.conf(4)` 手册页。

记录到 `syslog` 中的 DHCP 服务消息包括：

- 错误消息，这些消息通知您发生了某些错误，这些错误将阻止 DHCP 服务完成由客户机或您发出的请求。
- 警告和通知，它们通知您发生了异常情况，但这些异常情况不会阻止 DHCP 服务完成请求。

您可以增加使用 DHCP 守护进程的详细选项报告的信息量。详细消息输出可以帮助您对 DHCP 问题进行故障排除。请参见第 310 页中的 “如何生成详细的 DHCP 日志消息（DHCP 管理程序）”。

另一种有用的错误诊断技术是事务日志。事务提供了有关 DHCP 服务器或 BOOTP 中继与客户机之间每次交换的信息。DHCP 事务包括以下消息类型：

- ASSIGN－指定 IP 地址
- ACK－服务器确认客户机接受所提供的 IP 地址，并发送配置参数
- EXTEND－租用期延长
- RELEASE－IP 地址释放

- DECLINE－客户机正在拒绝指定地址
- INFORM－客户机正在请求网络配置参数但不请求 IP 地址
- NAK－服务器没有确认客户机发出的使用先前所用的 IP 地址的请求
- ICMP_ECHO－服务器检测到可能的 IP 地址已由其他主机使用

BOOTP 中继事务包括以下消息类型：

- RELAY-CLNT－消息正在从 DHCP 客户机中继到 DHCP 服务器
- RELAY–SRVR－消息正在从 DHCP 服务器中继到 DHCP 客户机

缺省情况下，DHCP 事务日志处于禁用状态。DHCP 事务日志在启用后将缺省使用 `syslog` 中的 `local0` 工具。生成的 DHCP 事务消息的 `syslog` 严重级别为 *notice*。此严重级别将导致在记录其他系统通知的文件中记录 DHCP 事务。不过，由于使用 `local` 工具，因此可以独立于其他通知记录 DHCP 事务消息。要单独记录事务消息，您必须编辑 `syslog.conf` 文件以指定单独的日志文件。有关 `syslog.conf` 文件的更多信息，请参见 `syslog.conf(4)` 手册页。

您可以禁用或启用事务日志，并可指定 `local0` 到 `local7` 之间的其他 `syslog` 工具，如第 311 页中的“如何启用和禁用 DHCP 事务日志（DHCP 管理程序）”中所述。在服务器系统的 `syslog.conf` 文件中，您还可以指示 `syslogd` 在单独的文件中存储 DHCP 事务消息。有关更多信息，请参见第 312 页中的“如何将 DHCP 事务记录到单独的 `syslog` 文件中”。

▼ 如何生成详细的 DHCP 日志消息（DHCP 管理程序）

1. **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的“如何启动和停止 DHCP 管理程序”。

   将打开 "Modify Service Options"（修改服务选项）对话框并显示 "Options"（选项）选项卡。请参见图 15–3。

2. **选择 "Verbose Log Messages"（详细日志消息）。**

3. **选择 "Restart Server"（重新启动服务器）。**

   "Restart Server"（重新启动服务器）选项位于对话框底部附近。

4. **单击 "OK"（确定）。**

   对于此会话以及每个后续会话，守护进程将在详细模式下运行，直到重置此选项。详细模式会降低守护进程的效率，因为显示消息要花费时间。

▼ 如何生成详细的 DHCP 日志消息（命令行）

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **键入以下命令设置详细模式：**

```
# /usr/sbin/dhcpconfig -P VERBOSE=true
```

下次启动 DHCP 服务器时，此服务器将以详细模式运行，直到关闭详细模式。

要关闭详细模式，请键入以下命令：

```
# /usr/sbin/dhcpconfig -P VERBOSE=
```

此命令不为 `VERBOSE` 关键字设置任何值，这样会导致从服务器的配置文件中删除此关键字。

详细模式会降低守护进程的效率，因为显示消息要花费时间。

▼ 如何启用和禁用 DHCP 事务日志（DHCP 管理程序）

此过程可为所有后续 DHCP 服务器会话启用和禁用事务日志。

1 **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **选择 "Log Transactions to Syslog Facility"（将事务处理记录到系统日志设备）。**

要禁用事务日志，请取消选中此选项。

3 **可选选择 0 到 7 之间的本地工具，用于记录 DHCP 事务。**

缺省情况下，在记录系统通知的位置记录 DHCP 事务，具体取决于 `syslogd` 的配置方式。如果要将 DHCP 事务记录到一个单独的没有记录其他系统通知的文件，请参见第 312 页中的 “如何将 DHCP 事务记录到单独的 `syslog` 文件中”。

启用事务日志之后，消息文件便会迅速变得非常庞大。

4 **选择 "Restart Server"（重新启动服务器）。**

5 **单击 "OK"（确定）。**

守护进程将在选定的 `syslog` 工具中为此会话以及每个后续会话记录事务，直到禁用日志为止。

▼ 如何启用和禁用 DHCP 事务日志（命令行）

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **选择以下步骤之一：**

- **要启用 DHCP 事务日志，请键入以下命令：**

  `# /usr/sbin/dhcpconfig -P LOGGING_FACILITY=`*syslog-local-facility***

  syslog-local-facility 是 0 到 7 之间的一个数字。如果省略此选项，则使用 0。

  缺省情况下，在记录系统通知的位置记录 DHCP 事务，具体取决于 `syslogd` 的配置方式。如果要将 DHCP 事务记录到一个单独的没有记录其他系统通知的文件，请参见第 312 页中的 “如何将 DHCP 事务记录到单独的 `syslog` 文件中”。

  启用事务日志之后，消息文件便会迅速变得非常庞大。

- **要禁用 DHCP 事务日志，请键入以下命令：**

  `# /usr/sbin/dhcpconfig -P LOGGING_FACILITY=`

  请注意，不要为此参数提供任何值。

▼ 如何将 DHCP 事务记录到单独的 syslog 文件中

1 **成为 DHCP 服务器系统上的超级用户或承担等效角色。**

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

为 DHCP 管理配置文件指定的角色可能没有足够的权限来执行此任务。角色必须拥有编辑 `syslog` 文件的权限。

2 **在服务器系统上编辑 `/etc/syslog.conf` 文件以添加具有如下格式的一行：**

`local`*n*`.notice` *path-to-logfile*

n 是您为事务日志指定的 syslog 工具编号，*path-to-logfile* 是用于记录事务的文件的完整路径。

例如，您可以添加以下行：

```
local0.notice /var/log/dhcpsrvc
```

有关 syslog.conf 文件的更多信息，请参见 syslog.conf(4) 手册页。

通过 DHCP 服务器启用动态 DNS 更新

DNS 为 Internet 提供了名称到地址以及地址到名称的服务。进行 DNS 映射之后，便可通过系统的主机名或 IP 地址访问此系统。还可以从系统所在域的外部访问此系统。

DHCP 服务可以通过两种方法使用 DNS：

- DHCP 服务器可以查找映射到服务器为客户机指定的 IP 地址的主机名。然后，服务器返回客户机的主机名以及客户机的其他配置信息。
- 如果 DHCP 服务器配置为更新 DNS，则它可以尝试代表客户机进行 DNS 映射。当请求 DHCP 服务时，客户机可以提供自己的主机名。如果配置为进行 DNS 更新，则 DHCP 服务器会尝试使用客户机建议的主机名来更新 DNS。如果 DNS 更新成功，则 DHCP 服务器会将请求的主机名返回到客户机。如果 DNS 更新不成功，则 DHCP 服务器会将其他主机名返回到客户机。

您可以启用 DHCP 服务，以便针对提供自己主机名的 DHCP 客户机更新 DNS 服务。要使 DNS 更新功能能够正常工作，必须正确设置 DNS 服务器、DHCP 服务器以及 DHCP 客户机。此外，域中的其他系统不得使用请求的主机名。

如果以下陈述成立，则 DHCP 服务器的 DNS 更新功能即可正常工作：

- DNS 服务器支持 RFC 2136。
- 无论在 DHCP 服务器系统上还是在 DNS 服务器系统上，DNS 软件均基于 BIND v8.2.2（修补程序级别 5 或更高版本）。
- DNS 服务器配置为从 DHCP 服务器接受动态 DNS 更新。
- DHCP 服务器配置为进行动态 DNS 更新。
- 在 DHCP 服务器上针对 DHCP 客户机网络配置了 DNS 支持。
- DHCP 客户机配置为在其 DHCP 请求消息中提供请求的主机名。
- 请求的主机名对应于 DHCP 拥有的地址。此主机名也可能没有相应的地址。

▼ 如何针对 DHCP 客户机启用动态 DNS 更新

注 – 应注意，动态 DNS 更新存在**安全风险**。

缺省情况下，Oracle Solaris DNS 守护进程 (`in.named`) 不允许动态更新。动态 DNS 更新的授权在 DNS 服务器系统上的 `named.conf` 配置文件中授予。没有提供任何其他安全性。您必须仔细权衡此功能为用户提供的便利性和启用动态 DNS 更新时存在的安全风险。

1. **在 DNS 服务器上，以超级用户的身份编辑 `/etc/named.conf` 文件。**

2. **在 `named.conf` 文件中查找适当的域的 `zone` 部分。**

3. **将 DHCP 服务器的 IP 地址添加到 `allow-update` 关键字中。**

   如果 `allow-update` 关键字不存在，请插入该关键字。

   例如，如果 DHCP 服务器驻留在地址 `10.0.0.1` 和 `10.0.0.2` 上，则应按如下方式修改 `dhcp.domain.com` 区域的 `named.conf` 文件：

   ```
   zone "dhcp.domain.com" in {
               type master;
               file "db.dhcp";
               allow-update { 10.0.0.1; 10.0.0.2; };
   };

   zone "10.IN-ADDR.ARPA" in {
               type master;
               file "db.10";
               allow-update { 10.0.0.1; 10.0.0.2; };
   };
   ```

   请注意，必须针对两个区域启用 `allow-update`，以便允许 DHCP 服务器同时更新 DNS 服务器上的 A 和 PTR 记录。

4. **在 DHCP 服务器上，启动 DHCP 管理程序。**

   ```
   # /usr/sadm/admin/bin/dhcpmgr &
   ```

   有关更多详细信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

5. **从 "Service"（服务）菜单中选择 "Modify"（修改）。**

   将打开 "Modify Service Options"（修改服务选项）对话框。

6. **选择 "Update DNS Host Information Upon Client Request"（根据用户的要求更新 DNS 主机信息）。**

7. **指定在超时之前等待 DNS 服务器发出响应的秒数，然后单击 "OK"（确定）。**

   缺省值 15 秒应已足够。如果有超时问题，则可以随后增大此值。

8 **单击 "Macros"（宏）选项卡，并确保指定了正确的 DNS 域。**

必须将带有正确域名的 `DNSdmain` 选项传递到任何需要动态 DNS 更新支持的客户机。缺省情况下，在服务器宏中指定 `DNSdmain`，此服务器宏用作绑定到每个 IP 地址的配置宏。

9 **设置 DHCP 客户机以便在请求 DHCP 服务时指定其主机名。**

如果您使用 DHCP 客户机，请参见第 384 页中的 “如何使 DHCPv4 客户机请求特定的主机名”。如果客户机不是 DHCP 客户机，请参见客户机文档以了解有关如何指定主机名的信息。

客户机主机名注册

如果让 DHCP 服务器针对放入 DHCP 服务中的 IP 地址生成主机名，则 DHCP 服务器可以在 NIS+、`/etc/inet/hosts` 或 DNS 名称服务中注册这些主机名。不能在 NIS 中进行主机名注册，因为 NIS 没有提供允许更新程序并传播 NIS 映射的协议。

注 – 仅当 DNS 服务器和 DHCP 服务器在同一系统上运行时，DHCP 服务器才能使用生成的主机名来更新 DNS。

如果 DHCP 客户机提供它自己的主机名，并且 DNS 服务器配置为允许从 DHCP 服务器进行动态更新，则 DHCP 服务器可以代表客户机更新 DNS。即使 DNS 服务器和 DHCP 服务器在不同的系统上运行，也可进行动态更新。有关启用此功能的更多信息，请参见第 313 页中的 “通过 DHCP 服务器启用动态 DNS 更新”。

下表概括了使用各种名称服务对 DHCP 客户机系统进行的客户机主机名注册。

表 15–2 名称服务中的客户机主机名注册

	主机名注册者	
名称服务	**DHCP 生成的主机名**	**DHCP 客户机提供的主机名**
NIS	NIS 管理员	NIS 管理员
NIS+	DHCP 工具	DHCP 工具
`/etc/hosts`	DHCP 工具	DHCP 工具
DNS	DHCP 工具（如果 DNS 服务器与 DHCP 服务器在同一系统上运行）	DHCP 服务器（如果配置为进行动态 DNS 更新）
	DNS 管理员（如果 DNS 服务器在其他系统上运行）	DNS 管理员（如果 DHCP 服务器没有配置为进行动态 DNS 更新）

DHCP 客户机可以在 DHCP 请求中请求特定的主机名，前提是它们已配置为可执行此操作，如第 384 页中的 “如何使 DHCPv4 客户机请求特定的主机名”中所述。请参阅其他 DHCP 客户机的供应商文档以确定是否支持此功能。

定制 DHCP 服务器的性能选项

您可以更改影响 DHCP 服务器性能的选项。这些选项在下表中介绍。

表 15–3 影响 DHCP 服务器性能的选项

服务器选项	说明	关键字
BOOTP 中继代理的最大跃点数	如果请求经过的 BOOTP 中继代理数超过给定数量，则此请求会被丢弃。缺省的中继代理的最大跃点数为 4。对于大多数网络而言，四个跃点可能已经足够。如果 DHCP 请求经过多个 BOOTP 中继代理之后才到达 DHCP 服务器，则网络所需的跃点数可能超过四个。	`RELAY_HOPS=`*integer*
检测重复地址	缺省情况下，服务器在将 IP 地址提供给客户机之前会对此地址执行 ping 操作。如果系统没有对 ping 操作做出响应，则表明尚未使用此地址。您可以禁用此功能，以便缩短服务器提供地址所花费的时间。但是，禁用此功能会存在使用重复 IP 地址的风险。	`ICMP_VERIFY=TRUE/FALSE`
按指定间隔自动重新装入 `dhcptab`	可以将服务器设置为按指定的间隔（以分钟为单位）自动读取 `dhcptab`。如果网络配置信息并非频繁更改且没有多台 DHCP 服务器，则无需自动重新装入 `dhcptab`。另外，请注意 DHCP 管理程序使您可以选择在更改数据之后让服务器重新装入 `dhcptab`。	`RESCAN_INTERVAL=`*min*
按指定间隔对所提供的 IP 地址进行高速缓存	服务器将 IP 地址提供给客户机之后，便会对此地址进行高速缓存。对所提供的地址进行高速缓存之后，服务器便不会再提供此地址。您可以更改对所提供的地址进行高速缓存的秒数。缺省值是 10 秒。在速度较慢的网络上，您可能需要延长对所提供的地址进行高速缓存的时间。	`OFFER_CACHE_TIMEOUT=`*sec*

以下过程介绍如何更改这些选项。

▼ 如何定制 DHCP 性能选项（DHCP 管理程序）

1. **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **更改所需的选项。**

有关选项的信息，请参见表 15–3。

3 **选择 "Restart Server"（重新启动服务器）。**

4 **单击 "OK"（确定）。**

▼ 如何定制 DHCP 性能选项（命令行）

如果通过此过程更改选项，则仅当重新启动 DHCP 服务器之后才能使用已更改的选项。

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **修改一个或多个性能选项：**

```
# /usr/sbin/dhcpconfig -P keyword=value,keyword=value...
```

keyword=value 可以是以下任一关键字：

`RELAY_HOPS=`*integer*	指定在守护进程删除 DHCP 或 BOOTP 数据报之前可以执行的中继代理的最大跃点数。
`ICMP_VERIFY=TRUE/FALSE`	启用或禁用重复 IP 地址自动检测功能。建议不要将此关键字设置为 `FALSE`。
`RESCAN_INTERVAL=`*minutes*	指定 DHCP 服务器应当用于安排自动重新读取 `dhcptab` 信息的间隔（以分钟为单位）。
`OFFER_CACHE_TIMEOUT=`*seconds*	指定 DHCP 服务器应当对所提供的地址进行高速缓存的秒数，这是为了搜索 DHCP 客户机而延长的时间。缺省设置为 10 秒。

示例 15–1 设置 DHCP 性能选项

以下是如何指定所有命令选项的示例。

```
# dhcpconfig -P RELAY_HOPS=2,ICMP_VERIFY=TRUE,\
RESCAN_INTERVAL=30,OFFER_CACHE_TIMEOUT=20
```

添加、修改和删除 DHCP 网络（任务列表）

配置 DHCP 服务器时，您还至少必须配置一个网络以使用 DHCP 服务。您可以随时添加更多的网络。

以下是描述使用初始化配置后的 DHCP 网络时可以执行的额外任务的任务列表。此任务列表包括指向用于执行这些任务的过程的链接。

任务	说明	参考
在服务器网络接口上启用或禁用 DHCP 服务	缺省行为是监视所有网络接口上的 DHCP 请求。如果不希望所有接口都接受 DHCP 请求，可以从受监视接口的列表中删除一个接口。	第 319 页中的“如何指定 DHCP 监视的网络接口（DHCP 管理程序）”
将新网络添加到 DHCP 服务中。	将网络纳入 DHCP 管理，目的是管理此网络上的 IP 地址。	第 321 页中的“如何添加 DHCP 网络（DHCP 管理程序）” 第 322 页中的“如何添加 DHCP 网络 (dhcpconfig)”
更改 DHCP 管理的网络的参数。	修改传递到特定网络上的客户机的信息。	第 323 页中的“如何修改 DHCP 网络配置（DHCP 管理程序）” 第 324 页中的“如何修改 DHCP 网络配置 (dhtadm)”
从 DHCP 服务中删除网络。	删除网络以使此网络上的 IP 地址不再由 DHCP 管理。	第 325 页中的“如何删除 DHCP 网络（DHCP 管理程序）” 第 326 页中的“如何删除 DHCP 网络 (pntadm)”

指定 DHCP 监视的网络接口

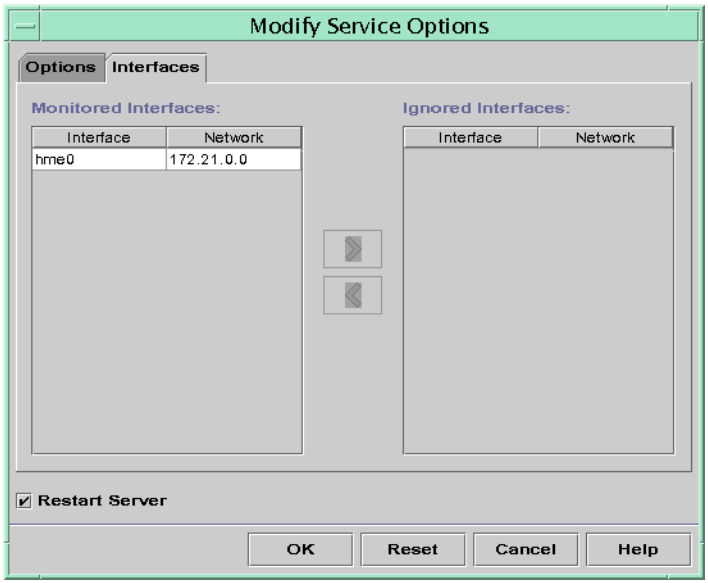
缺省情况下，`dhcpconfig` 和 DHCP 管理程序的 "Configuration Wizard"（配置向导）将 DHCP 服务器配置为监视所有服务器系统的网络接口。如果您将新网络接口添加到服务器系统，则引导系统时 DHCP 服务器会自动监视这一新的接口。然后，您可以通过此网络接口添加任何要监视的网络。

但是，您也可以指定应监视的网络接口以及应忽略的接口。如果您不希望在某个网络上提供 DHCP 服务，则可能需要忽略用于该网络的接口。

如果您指定应忽略任意接口，并在之后又安装了一个新接口，则 DHCP 服务器会忽略这一新接口。您必须将这一新接口添加到服务器的受监视接口列表中。您可以使用 DHCP 管理程序或 `dhcpconfig` 实用程序来指定接口。

本节介绍可用于指定 DHCP 应监视或忽略哪些网络接口的过程。此 DHCP 管理程序过程将使用 DHCP 管理程序的 "Modify Service Options"（修改服务选项）对话框的 "Interfaces"（接口）选项卡（下图中所示）。

图 15–4 DHCP 管理程序中 "Modify Service Options"（修改服务选项）对话框的 "Interfaces"（接口）选项卡



▼ 如何指定 DHCP 监视的网络接口（DHCP 管理程序）

1. **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。**

   将显示 "Modify Service Options"（修改服务选项）对话框。

   有关 DHCP 管理程序的信息，请参见第 302 页中的“如何启动和停止 DHCP 管理程序”。

2. **选择 "Interfaces"（接口）选项卡。**

3. **选择适当的网络接口。**

4. **单击箭头按钮以便将此接口移动到适当的列表中。**

   例如，要忽略某个接口，请在 "Monitored Interfaces"（监视接口）列表中选择此接口，然后单击右箭头按钮。这样，此接口便会显示在 "Ignored Interfaces"（忽略接口）列表中。

5. **选择 "Restart Server"（重新启动服务器），然后单击 "OK"（确定）。**

   每次重新引导之后都会保留您所做的更改。

▼ 如何指定 DHCP 监视的网络接口 (dhcpconfig)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **在 DHCP 服务器系统上键入以下命令：**

```
# /usr/sbin/dhcpconfig -P INTERFACES=int,int,...
```

int, int,... 是要监视的接口的列表。接口名称必须用逗号分隔。

例如，可以使用以下命令仅监视 `ge0` 和 `ge1`：

```
#/usr/sbin/dhcpconfig -P INTERFACES=ge0,ge1
```

在 `dhcpconfig` 命令行中，应省略您要忽略的接口。

每次重新引导之后都会保留使用此命令所做的更改。

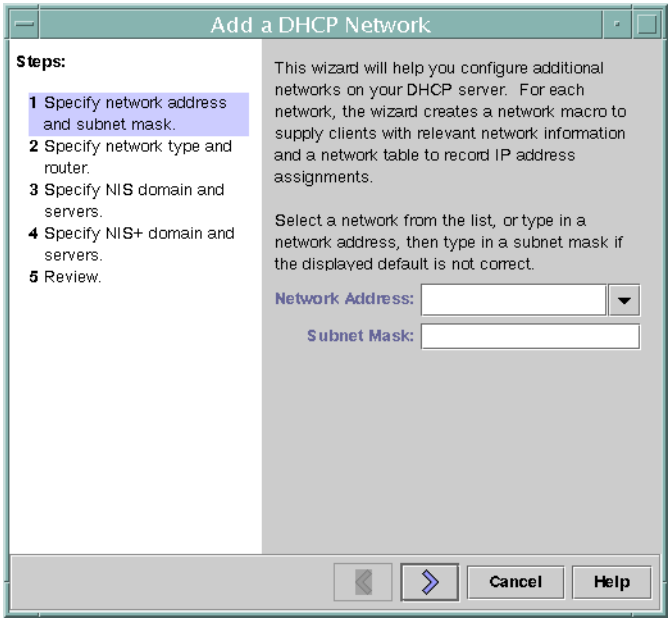
添加 DHCP 网络

在使用 DHCP 管理程序配置服务器的同时，还会配置第一个网络。第一个网络通常为服务器系统主接口上的本地网络。如果您要配置其他网络，请使用 DHCP 管理程序中的 "DHCP Network Wizard"（DHCP 网络向导）。

如果您使用 `dhcpconfig -D` 命令来配置服务器，则必须分别配置所有要使用 DHCP 服务的网络。有关更多信息，请参见第 322 页中的 “如何添加 DHCP 网络 (dhcpconfig)”。

下图显示了 DHCP 管理程序中 "DHCP Network Wizard"（DHCP 网络向导）的初始对话框。

图 15–5 DHCP 管理程序的 "Network Wizard"（网络向导）



当您配置新网络时，DHCP 管理程序会创建以下组件：

- 数据存储中的网络表。新网络显示在 DHCP 管理程序的 "Addresses"（地址）选项卡内的网络列表中。
- 包含驻留在此网络上的客户机所需的信息的网络宏。网络宏的名称与网络的 IP 地址相匹配。网络宏将添加到数据存储内的 `dhcptab` 表中。

▼ 如何添加 DHCP 网络（DHCP 管理程序）

1 **在 DHCP 管理程序中，单击 "Addresses"（地址）选项卡。**

将列出已针对 DHCP 服务配置的所有网络。

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **从 "Edit"（编辑）菜单中选择 "Network Wizard"（网络向导）。**

3 **选择选项，或者键入所需的信息。请使用在规划阶段所作的决定来确定要指定的信息。**

规划在第 287 页中的 “规划远程网络的 DHCP 配置”中介绍。

如果您不了解此向导，请在向导窗口中单击 "Help"（帮助）。Web 浏览器便会显示 "DHCP Network Wizard"（DHCP 网络向导）的帮助信息。

4 **完成指定所需的信息之后，单击 "Finish"（完成）以完成网络配置。**

"Network Wizard"（网络向导）将创建空的网络表，此表在窗口的左窗格中列出。

"Network Wizard"（网络向导）还将创建一个名称与网络的 IP 地址相匹配的网络宏。

5 **可选选择 "Macros"（宏）选项卡并选择网络宏以查看宏的内容。**

您可以确认在此向导中提供的信息是否已作为网络宏中选项的值插入。

另请参见 您必须为网络添加地址，然后此网络的 IP 地址才可由 DHCP 管理。有关更多信息，请参见第 333 页中的 “将 IP 地址添加到 DHCP 服务”。

即使将网络表保留为空，DHCP 服务器仍可以为客户机提供配置信息。有关更多信息，请参见第 364 页中的 “设置 DHCP 客户机为仅接收信息（任务列表）”。

▼ 如何添加 DHCP 网络 (dhcpconfig)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **在 DHCP 服务器系统上键入以下命令：**

```
# /usr/sbin/dhcpconfig -N network-address
```

network-address 是您要添加到 DHCP 服务中的网络 IP 地址。有关可以与 `-N` 选项一起使用的子选项，请参见 `dhcpconfig(1M)` 手册页。

如果没有使用子选项，则 `dhcpconfig` 使用网络文件来获取有关网络的信息。

另请参见 您必须为网络添加地址，然后此网络的 IP 地址才可由 DHCP 管理。有关更多信息，请参见第 333 页中的 “将 IP 地址添加到 DHCP 服务”。

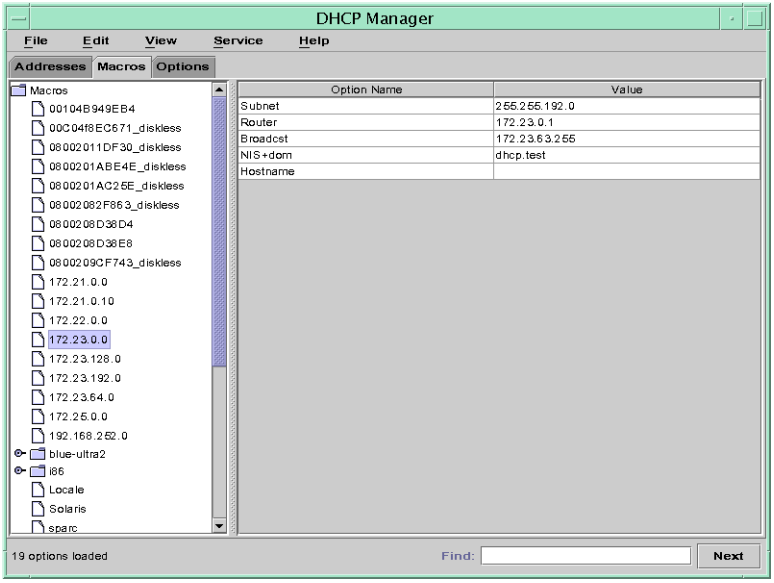
即使将网络表保留为空，DHCP 服务器仍可以为客户机提供配置信息。有关更多信息，请参见第 364 页中的 “设置 DHCP 客户机为仅接收信息（任务列表）”。

修改 DHCP 网络配置

将网络添加到 DHCP 服务之后，便可修改您最初提供的配置信息。配置信息存储在用于将信息传递到网络上的客户机的网络宏中。您必须修改网络宏才能更改网络配置。

下图显示了 DHCP 管理程序的 "Macros"（宏）选项卡。

图 15–6 DHCP 管理程序的 "Macros"（宏）选项卡



▼ 如何修改 DHCP 网络配置（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**
   将在左窗格中列出针对此 DHCP 服务器定义的所有宏。

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择名称与要更改的网络配置相匹配的网络宏。**
   网络宏名称即为网络 IP 地址。

3. **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**
   "Macro Properties"（宏属性）对话框将显示宏中的选项表。

4 **选择要修改的选项。**

选项名称和值将显示在此对话框顶部附近的文本字段中。

5 **可选修改选项名称，或者选择 "Select"（选择）按钮以显示选项名称列表。**

"Select Option"（选择选项）对话框将显示所有 DHCP 标准选项的列表以及每个选项的简短说明。

6 **可选在 "Select Option"（选择选项）对话框中选择选项名称，然后单击 "OK"（确定）。**

新的选项名称将显示在 "Option Name"（选项名称）字段中。

7 **为选项键入新值，然后单击 "Modify"（修改）。**

8 **可选您还可以通过在此对话框中选择 "Select"（选择），将选项添加到网络宏中。**

有关修改宏的更多常规信息，请参见第 346 页中的 “修改 DHCP 宏”。

9 **选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器），然后单击 "OK"（确定）。**

此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

▼ 如何修改 DHCP 网络配置 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **确定哪个宏包含网络的所有客户机的信息。**

网络宏的名称与网络 IP 地址相匹配。

如果您不知道哪个宏包含此信息，则可以使用命令 `dhtadm -P` 显示 `dhcptab` 表来列出所有宏。

3 **键入如下格式的命令以更改要更改的选项的值：**

```
# dhtadm -M -m macro-name -e 'symbol=value' -g
```

有关 `dhtadm` 命令行选项的更多信息，请参见 `dhtadm(1M)` 手册页。

示例 15–2 使用 dhtadm 命令修改 DHCP 宏

例如，要将 10.25.62.0 宏的租用时间更改为 57600 秒，将 NIS 域更改为 sem.example.com，请键入以下命令：

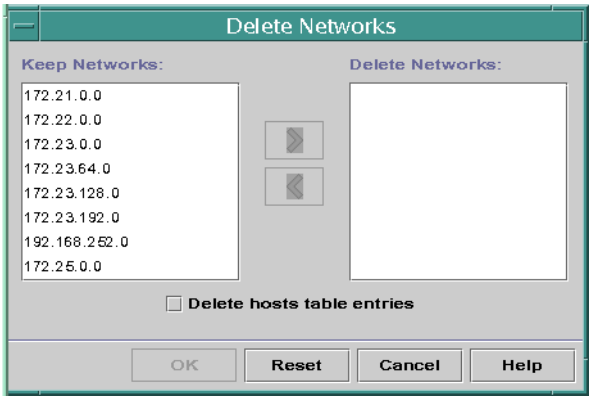
```
# dhtadm -M -m 10.25.62.0 -e 'LeaseTim=57600' -g

# dhtadm -M -m 10.25.62.0 -e 'NISdmain=sem.example.com' -g
```

-g 选项会使 DHCP 守护进程重新读取 dhcptab 表，并使更改生效。

删除 DHCP 网络

使用 DHCP 管理程序，一次可以删除多个网络。您还可以选择自动删除与这些网络上 DHCP 管理的 IP 地址关联的主机表项。下图显示了 DHCP 管理程序的 "Delete Networks"（删除网络）对话框。

图 15–7 DHCP 管理程序中的 "Delete Networks"（删除网络）对话框



pntadm 命令要求您先删除网络中的每个 IP 地址项，然后再删除网络。一次只能删除一个网络。

▼ 如何删除 DHCP 网络（DHCP 管理程序）

1 **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **从 "Edit"（编辑）菜单中选择 "Delete Networks"（删除网络）。**

将打开 "Delete Networks"（删除网络）对话框。

3 **在 "Keep Networks"（保留的网络）列表中，选择要删除的网络。**

在按住 Ctrl 键的同时用鼠标单击以选择多个网络。在按住 Shift 键的同时进行单击以选择一系列网络。

4 **单击右箭头按钮将选定的网络移动到 "Delete Networks"（删除的网络）列表中。**

5 **如果您要删除此网络 DHCP 地址的主机表项，请选择 "Delete Host Table Entries"（删除主机表项）。**

请注意，删除主机表项不会删除 DNS 服务器上这些地址的主机注册，而只是在本地名称服务中删除这些项。

6 **单击 "OK"（确定）。**

▼ 如何删除 DHCP 网络 (pntadm)

请注意，此过程将先从 DHCP 网络表中删除网络的 IP 地址，然后再删除网络。将删除地址以确保从 `hosts` 文件或数据库中删除主机名。

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **键入如下格式的命令以从名称服务中删除 IP 地址及其主机名：**

`# pntadm -D -y` *IP-address*

例如，要删除 IP 地址 `10.25.52.1`，请键入以下命令：

```
# pntadm -D -y 10.25.52.1
```

`-y` 选项指定删除主机名。

3 **对于网络中的每个地址重复执行 `pntadm -D -y` 命令。**

如果您要删除多个地址，则可能需要创建脚本以运行 `pntadm` 命令。

4 **删除所有地址之后，键入以下命令以从 DHCP 服务中删除网络。**

`# pntadm -R` *network-IP-address*

例如，要删除网络 `10.25.52.0`，请键入以下命令：

```
# pntadm -R 10.25.52.0
```

有关使用 pntadm 实用程序的更多信息，请参见 pntadm(1M) 手册页。

通过 DHCP 服务支持 BOOTP 客户机（任务列表）

要在 DHCP 服务器上支持 BOOTP 客户机，您必须将 DHCP 服务器设置为与 BOOTP 兼容。如果您要指定哪些 BOOTP 客户机可以使用 DHCP，则可以在 DHCP 服务器的网络表中注册 BOOTP 客户机。或者，您可以保留多个 IP 地址以自动分配给 BOOTP 客户机。

注 – 无论是否为 BOOTP 地址明确指定永久性租用，都会永久指定此 BOOTP 地址。

以下任务列表描述支持 BOOTP 客户机可能需要执行的任务。此任务列表包含指向用于执行这些任务的过程的链接。

任务	说明	参考
设置自动 BOOTP 支持。	在 DHCP 管理的网络上，或者在通过中继代理连接到 DHCP 管理的网络的网络上，为任何 BOOTP 客户机提供 IP 地址。 您必须保留一个地址池来专供 BOOTP 客户机使用。在服务器必须支持大量 BOOTP 客户机的情况下，此选项可能更为有用。	第 327 页中的“如何设置对任意 BOOTP 客户机的支持（DHCP 管理程序）”
设置手动 BOOTP 支持。	仅为那些已手动注册 DHCP 服务的 BOOTP 客户机提供 IP 地址。 此选项要求您将客户机的 ID 绑定到已为 BOOTP 客户机标记的特定 IP 地址。在具有少量 BOOTP 客户，或者您要限制可以使用 DHCP 服务器的 BOOTP 客户机的情况下，此选项非常有用。	第 328 页中的“如何设置对已注册的 BOOTP 客户机的支持（DHCP 管理程序）”

▼ 如何设置对任意 BOOTP 客户机的支持（DHCP 管理程序）

1. **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。**

   将打开 "Modify Service Options"（修改服务选项）对话框。

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **在此对话框的 "BOOTP Compatibility"（BOOTP 兼容性）部分中，选择 "Automatic"（自动）。**
3. **选择 "Restart Server"（重新启动服务器），然后单击 "OK"（确定）。**
4. **选择 "Addresses"（地址）选项卡。**
5. **选择要为 BOOTP 客户机保留的地址。**

   通过单击第一个地址，按住 Shift 键，然后单击最后一个地址，选择一系列地址。通过在按住 Ctrl 键的同时单击每个地址，选择多个不连续的地址。
6. **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**

   将打开 "Modify Multiple Addresses"（修改多个地址）对话框。
7. **在 "BOOTP" 部分中，选择 "Assign All Addresses Only to BOOTP Clients"（仅为 BOOTP 客户机指定所有地址）。**

   所有其他选项应设置为 "Keep Current Settings"（保持当前设置）。
8. **单击 "OK"（确定）。**

   现在，任意 BOOTP 客户机均可从此 DHCP 服务器中获取地址。

▼ 如何设置对已注册的 BOOTP 客户机的支持（DHCP 管理程序）

1. **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。**

   将打开 "Modify Service Options"（修改服务选项）对话框。

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。
2. **在此对话框的 "BOOTP Compatibility"（BOOTP 兼容性）部分中，选择 "Manual"（手动）。**
3. **选择 "Restart Server"（重新启动服务器），然后单击 "OK"（确定）。**
4. **选择 "Addresses"（地址）选项卡。**
5. **选择要指定给特定 BOOTP 客户机的地址。**

6 **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**

将打开 "Address Properties"（地址属性）对话框。

7 **在 "Address Properties"（地址属性）对话框中，选择 "Lease"（租用）选项卡。**

8 **在 "Client ID"（客户机 ID）字段中，键入客户机的标识符。**

对于以太网上的 BOOTP Oracle Solaris 客户机，客户机 ID 是从客户机的十六进制以太网地址派生的字符串。客户机 ID 包括一个指示以太网地址解析协议 (Address Resolution Protocol, ARP) 类型的前缀 (01)。例如，以太网地址为 `8:0:20:94:12:1e` 的 BOOTP 客户机可以使用客户机 ID `0108002094121E`。

提示 – 以 Oracle Solaris 客户机系统上超级用户的身份，键入以下命令来获取接口的以太网地址：

```
# ifconfig -a
```

---

9 **选择 "Reserved"（保留）以保留此客户机的 IP 地址。**

10 **选择 "Assign Only to BOOTP Clients"（仅指定给 BOOTP 客户机），然后单击 "OK"（确定）。**

在 "Addresses"（地址）选项卡中，BOOTP 将在 "Status"（状态）字段中显示，您指定的客户机 ID 将在 "Client ID"（客户机 ID）字段中列出。

在 DHCP 服务中处理 IP 地址（任务列表）

您可以使用 DHCP 管理程序或 `pntadm` 命令在 DHCP 服务中添加 IP 地址、修改地址属性以及删除地址。在处理 IP 地址之前，应参阅表 15–4 熟悉 IP 地址属性。此表为用户提供了 DHCP 管理程序和 `pntadm` 的信息。

注 – 表 15–4 包括添加和修改 IP 地址时使用 `pntadm` 指定 IP 地址属性的示例。有关 `pntadm` 的更多信息，另请参阅 `pntadm(1M)` 手册页。

以下任务列表列出了添加、修改或删除 IP 地址时必须执行的任务。此任务列表还包含指向用于执行这些任务的过程的链接。

任务	说明	参考
将单个或多个 IP 地址添加到 DHCP 服务中。	使用 DHCP 管理程序，在已由 DHCP 服务管理的网络上添加 IP 地址。	第 334 页中的“如何添加单个 IP 地址（DHCP 管理程序）” 第 334 页中的“如何复制现有 IP 地址（DHCP 管理程序）” 第 335 页中的“如何添加多个 IP 地址（DHCP 管理程序）” 第 335 页中的“如何添加 IP 地址 (pntadm)”
更改 IP 地址的属性。	更改表 15–4 中所述的任意 IP 地址属性。	第 337 页中的“如何修改 IP 地址属性（DHCP 管理程序）” 第 338 页中的“如何修改 IP 地址属性 (pntadm)”
从 DHCP 服务中删除 IP 地址。	防止使用由 DHCP 指定的 IP 地址。	第 339 页中的“如何将 IP 地址标记为不可用（DHCP 管理程序）” 第 339 页中的“如何将 IP 地址标记为不可用 (pntadm)” 第 340 页中的“如何从 DHCP 服务中删除 IP 地址（DHCP 管理程序）” 第 341 页中的“如何从 DHCP 服务中删除 IP 地址 (pntadm)”
为 DHCP 客户机指定相同的 IP 地址。	将客户机设置为在每次请求其配置时都收到相同的 IP 地址。	第 342 页中的“如何为 DHCP 客户机指定相同的 IP 地址（DHCP 管理程序）” 第 343 页中的“如何为 DHCP 客户机指定相同的 IP 地址 (pntadm)”

下表列出并介绍了 IP 地址的属性。

表 15–4 IP 地址属性

属性	说明	如何在 pntadm 命令中指定
网络地址	包含您要处理的 IP 地址的网络的地址。 网络地址显示在 DHCP 管理程序的 "Addresses"（地址）选项卡内的 "Networks"（网络）列表中。	网络地址必须是用于创建、修改或删除 IP 地址的 pntadm 命令行中的最后一个参数。 例如，要将 IP 地址添加到网络 10.21.0.0 中，请键入： **pntadm -A** *ip-address options* **10.21.0.0**

表 15–4 IP 地址属性 （续）

属性	说明	如何在 pntadm 命令中指定
IP 地址	您要处理（创建、修改或删除）的地址。 IP 地址显示在 DHCP 管理程序的 "Addresses"（地址）选项卡的第一列中。	IP 地址必须与 pntadm 命令的 -A、-M 和 -D 选项一起出现。 例如，要修改 IP 地址 10.21.5.12，请键入： **pntadm -M 10.21.5.12** *options* **10.21.0.0**
客户机名称	映射到主机表中 IP 地址的主机名。此名称可以由 DHCP 管理程序在创建地址时自动生成。如果您创建单个地址，则可以提供名称。	使用 -h 选项指定客户机名称。 例如，要为 10.21.5.12 指定客户机名称 carrot12，请键入： **pntadm -M 10.21.5.12 -h carrot12 10.21.0.0**
归服务器所有	管理 IP 地址并对 DHCP 客户机 IP 地址分配的请求做出响应的 DHCP 服务器。	使用 -s 选项指定所属服务器名称。 例如，要指定服务器 blue2 拥有 10.21.5.12，请键入： **pntadm -M 10.21.5.12 -s blue2 10.21.0.0**
配置宏	DHCP 服务器用于从 dhcptab 表中获取网络配置选项的宏。当您配置服务器和添加网络时，会自动创建多个宏。有关宏的更多信息，请参见第 273 页中的 “关于 DHCP 宏”。创建地址时，还会创建一个服务器宏。服务器宏作为配置宏指定给每个地址。	使用 -m 选项指定宏名称。 例如，要将服务器宏 blue2 指定给地址 10.21.5.12，请键入： **pntadm -M 10.21.5.12 -m blue2 10.21.0.0**
客户机 ID	在 DHCP 服务中具有唯一性的文本字符串。 如果列出的客户机 ID 为 00，则表示没有为任何客户机分配地址。如果您在修改 IP 地址属性时指定了客户机 ID，则此地址以独占方式绑定到此客户机。 客户机 ID 由 DHCP 客户机的供应商确定。如果客户机不是 DHCP 客户机，请参阅客户机文档以了解更多信息。	使用 -i 选项指定客户机 ID。 例如，要将客户机 ID 08002094121E 指定给地址 10.21.5.12，请键入： **pntadm -M 10.21.5.12 -i 0108002094121E 10.21.0.0**

表 15–4 IP 地址属性 （续）

属性	说明	如何在 pntadm 命令中指定
	对于 DHCP 客户机，客户机 ID 由客户机的十六进制硬件地址派生而来。客户机 ID 包括一个表示网络类型的 ARP 代码前缀，例如 01 表示以太网。ARP 代码由 Internet 编号分配机构 (Internet Assigned Numbers Authority, IANA) 在 http://www.iana.com/numbers.html 上编号指定标准的 "ARP Parameters"（ARP 参数）部分指定。 例如，十六进制以太网地址为 8:0:20:94:12:1e 的 Oracle Solaris 客户机所使用的客户机 ID 为 0108002094121E。如果客户机当前使用一个地址，则客户机 ID 将在 DHCP 管理程序和 pntadm 中列出。 **提示**：在 Oracle Solaris 客户机系统上以超级用户身份，键入以下命令来获取接口的以太网地址：ifconfig -a。	
保留	该设置指定地址专门为客户机 ID 指示的客户机保留并且 DHCP 服务器不能回收此地址。如果您选择此选项，请手动为客户机指定地址。	指定手动或使用 -f 选项保留地址。 例如，要指定为客户机保留 IP 地址 10.21.5.12，请键入： **pntadm -M 10.21.5.12 -f MANUAL 10.21.0.0**
租用类型或策略	该设置确定 DHCP 如何管理客户机对 IP 地址的使用。租用既可以是动态的，也可以是永久性的。有关完整说明，请参见第 285 页中的 “动态和永久租用类型”。	指定使用 -f 选项永久指定地址。缺省情况下，将动态租用地址。 例如，要指定永久租用 IP 地址 10.21.5.12，请键入： **pntadm -M 10.21.5.12 -f PERMANENT 10.21.0.0**
租用失效日期	租用失效的日期，仅当指定了动态租用时才适用。日期以 *mm/dd/yyyy* 格式指定。	使用 -e 选项指定失效日期。 例如，要将失效日期指定为 2006 年 1 月 1 日，请键入： **pntadm -M 10.21.5.12 -e 01/01/2006 10.21.0.0**
BOOTP 设置	该设置将地址标记为保留供 BOOTP 客户机使用。有关支持 BOOTP 客户机的更多信息，请参见第 327 页中的 “通过 DHCP 服务支持 BOOTP 客户机（任务列表）”。	使用 -f 选项为 BOOTP 客户机保留地址。 例如，要为 BOOTP 客户机保留 IP 地址 10.21.5.12，请键入： **pntadm -M 10.21.5.12 -f BOOTP 10.21.0.0**
不可用的设置	该设置将地址标记为禁止向任何客户机指定此地址。	使用 -f 选项将地址标记为不可用。 例如，要将 IP 地址 10.21.5.12 标记为不可用，请键入： **pntadm -M 10.21.5.12 -f UNUSABLE 10.21.0.0**

将 IP 地址添加到 DHCP 服务

在添加 IP 地址之前，您必须将拥有这些地址的网络添加到 DHCP 服务中。有关添加网络的信息，请参见第 320 页中的 “添加 DHCP 网络”。

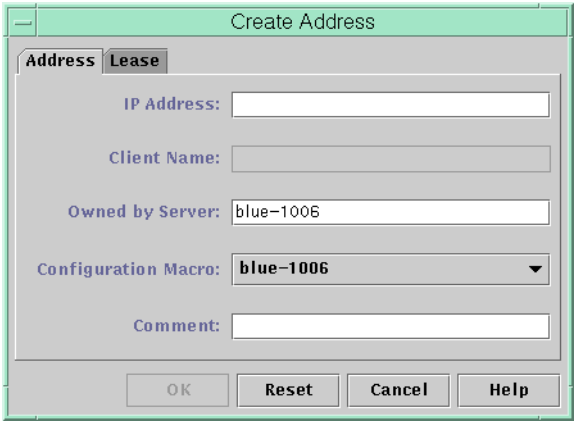
您可以使用 DHCP 管理程序或 `pntadm` 命令来添加地址。

在已由 DHCP 服务管理的网络上，您可以使用 DHCP 管理程序通过几种方法来添加地址：

- **添加单个 IP 地址** – 将一个新的 IP 地址纳入 DHCP 管理。
- **复制现有 IP 地址** – 复制由 DHCP 管理的现有 IP 地址的属性，并提供新的 IP 地址和客户机名称。
- **添加一系列 IP 地址** – 使用 "Address Wizard"（地址向导）将一系列 IP 地址纳入 DHCP 管理。

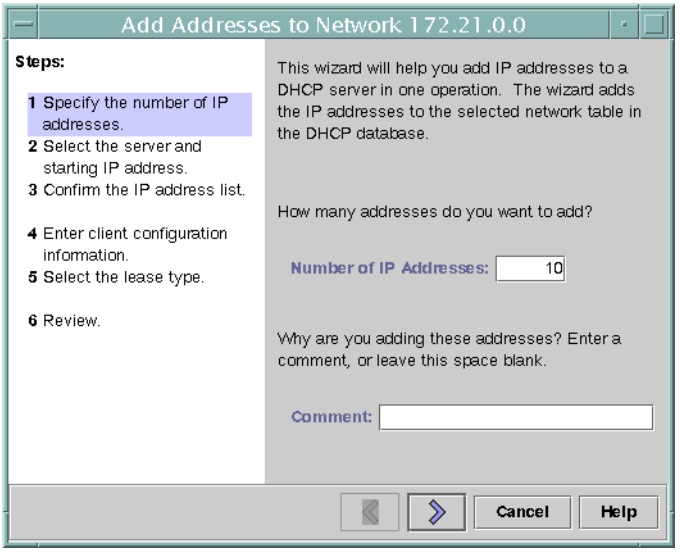
下图显示了 "Create Address"（创建地址）对话框。"Duplicate Address"（复制地址）对话框等同于 "Create Address"（创建地址）对话框，只是文本字段显示现有地址的值。

图 15–8 DHCP 管理程序中的 "Create Address"（创建地址）对话框



下图显示了用于添加一系列 IP 地址的 "Add Addresses to Network"（将地址添加到网络）向导的第一个对话框。

图 15–9 DHCP 管理程序中的 "Add Addresses to Network"（将地址添加到网络）向导



▼ 如何添加单个 IP 地址（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。
2. **选择要添加新 IP 地址的网络。**
3. **从 "Edit"（编辑）菜单中选择 "Create"（创建）。**
   将打开 "Create Address"（创建地址）对话框。
4. **在 "Address"（地址）和 "Lease"（租用）选项卡上，为地址设置选择或键入值。**
   选择 "Help"（帮助）按钮以打开 Web 浏览器，来显示此对话框的帮助信息。有关这些设置的详细信息，另请参见表 15–4。
5. **单击 "OK"（确定）。**

▼ 如何复制现有 IP 地址（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **选择新 IP 地址所在的网络。**

3 **选择具有要复制的属性的地址。**

4 **从 "Edit"（编辑）菜单中选择 "Duplicate"（复制）。**

5 **在 "IP Address"（IP 地址）字段中指定新的 IP 地址。**

6 **可选为此地址指定新的客户机名称。**
您所使用的名称不能与要复制的地址所使用的名称相同。

7 **可选如有必要，修改其他选项值。**
大多数其他选项值应保持不变。

8 **单击 "OK"（确定）。**

▼ 如何添加多个 IP 地址（DHCP 管理程序）

1 **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**
有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **选择要添加多个新 IP 地址的网络。**

3 **从 "Edit"（编辑）菜单中选择 "Address Wizard"（地址向导）。**
"Add Addresses to Network"（将地址添加到网络）对话框将提示您为 IP 地址属性提供值。有关这些属性的更多信息，请参见表 15–4，或者在此对话框中选择 "Help"（帮助）按钮。第 284 页中的 “为 IP 地址管理做出决定（任务列表）”介绍了更详细的信息。

4 **完成每个屏幕上的操作时单击右箭头按钮，并在最后一个屏幕上单击 "Finish"（完成）。**
"Addresses"（地址）选项卡将使用新地址进行更新。

▼ 如何添加 IP 地址 (pntadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**
有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来添加 IP 地址：**

```
# pntadm -A ip-address options network-address
```

有关可与 `pntadm -A` 一起使用的选项的列表，请参阅 `pntadm(1M)` 手册页。此外，表 15–4 还显示了指定这些选项的一些 `pntadm` 命令样例。

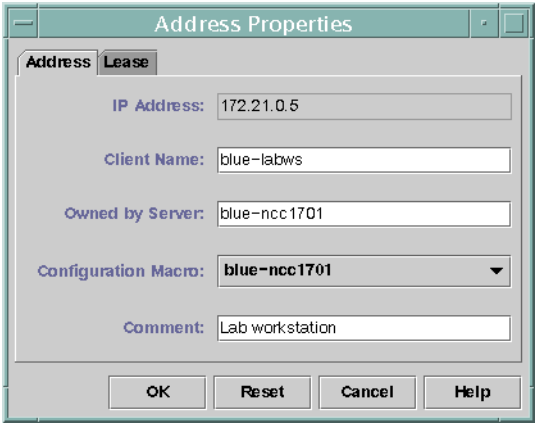
注 – 您可以编写脚本以使用 `pntadm` 添加多个地址。有关示例，请参见示例 18–1。

在 DHCP 服务中修改 IP 地址

您可以使用 DHCP 管理程序或 `pntadm -M` 命令来修改表 15–4 中所述的任何地址属性。有关 `pntadm -M` 的更多信息，请参见 `pntadm(1M)` 手册页。

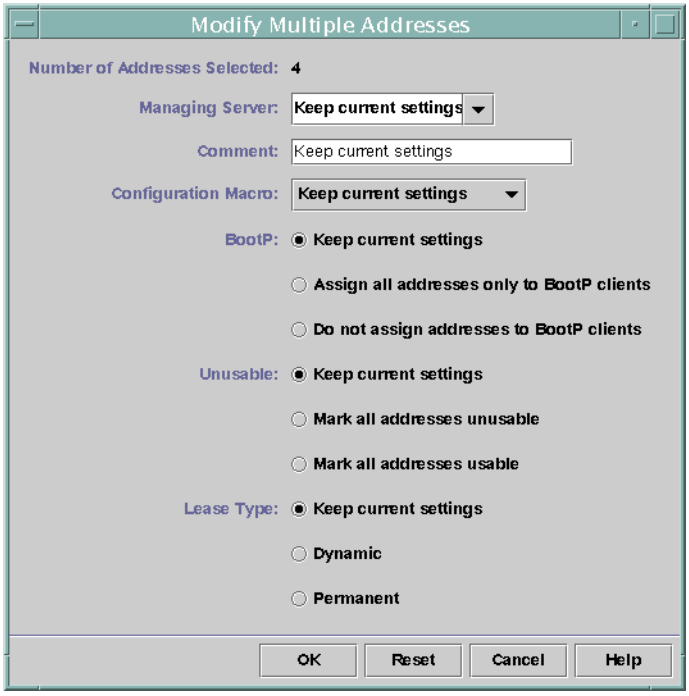
下图显示了用于修改 IP 地址属性的 "Address Properties"（地址属性）对话框。

图 15–10 DHCP 管理程序中的 "Address Properties"（地址属性）对话框



下图显示了用于修改多个 IP 地址的 "Modify Multiple Addresses"（修改多个地址）对话框。

图 15–11 DHCP 管理程序中的 "Modify Multiple Addresses"（修改多个地址）对话框



▼ 如何修改 IP 地址属性（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择 IP 地址所在的网络。**

3. **选择一个或多个要修改的 IP 地址。**

   如果您要修改多个地址，请在按住 Ctrl 键的同时用鼠标单击以选择多个地址。您还可以在按住 Shift 键的同时进行单击以选择一系列地址。

4. **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**

   将打开 "Address Properties"（地址属性）对话框或 "Modify Multiple Address"（修改多个地址）对话框。

5. **更改适当的属性。**

   单击 "Help"（帮助）按钮，或参阅表 15–4获取有关这些属性的信息。

6 **单击 "OK"（确定）。**

▼ 如何修改 IP 地址属性 (pntadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来修改 IP 地址属性：**

```
# pntadm -M ip-address options network-address
```

许多选项都可以与 `pntadm` 命令一起使用，这些选项在 `pntadm(1M)` 手册页中进行了介绍。

表 15–4 显示了指定这些选项的一些 `pntadm` 命令样例。

从 DHCP 服务中删除 IP 地址

有时，您可能希望 DHCP 服务停止管理某个特定的 IP 地址或地址组。用于从 DHCP 中删除地址的方法取决于您希望更改是临时更改还是永久性更改。

- 要暂时阻止使用地址，可以在 "Address Properties"（地址属性）对话框中将地址标记为不可用，如第 338 页中的 “通过 DHCP 服务将 IP 地址标记为不可用”中所述。
- 要永久阻止 DHCP 客户机使用地址，请从 DHCP 网络表中删除地址，如第 340 页中的 “从 DHCP 服务中删除 IP 地址”中所述。

通过 DHCP 服务将 IP 地址标记为不可用

您可以使用带有 `-f UNUSABLE` 选项的 `pntadm -M` 命令将地址标记为不可用。

在 DHCP 管理程序中，可以使用图 15–10 中所示的 "Address Properties"（地址属性）对话框标记单个地址。可以使用图 15–11 中所示的 "Modify Multiple Addresses"（修改多个地址）对话框标记多个地址，如以下过程中所述。

▼ 如何将 IP 地址标记为不可用（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的“如何启动和停止 DHCP 管理程序”。

2. **选择 IP 地址所在的网络。**

3. **选择一个或多个要标记为不可用的 IP 地址。**

   如果您要将多个地址标记为不可用，请在按住 Ctrl 键的同时用鼠标单击以选择多个地址。您还可以在按住 Shift 键的同时进行单击以选择一系列地址。

4. **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**

   将打开 "Address Properties"（地址属性）对话框或 "Modify Multiple Address"（修改多个地址）对话框。

5. **如果要修改一个地址，请选择 "Lease"（租用）选项卡。**

6. **选择 "Address is Unusable"（地址不可用）。**

   如果要编辑多个地址，请选择 "Mark All Addresses Unusable"（将所有地址标记为不可用）。

7. **单击 "OK"（确定）。**

▼ 如何将 IP 地址标记为不可用 (pntadm)

1. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的“设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **通过键入如下格式的命令将 IP 地址标记为不可用：**

   `# pntadm -M` *ip-address* **`-f UNUSABLE`** *network-address*

   例如，要将地址 `10.64.3.3` 标记为不可用，请键入：

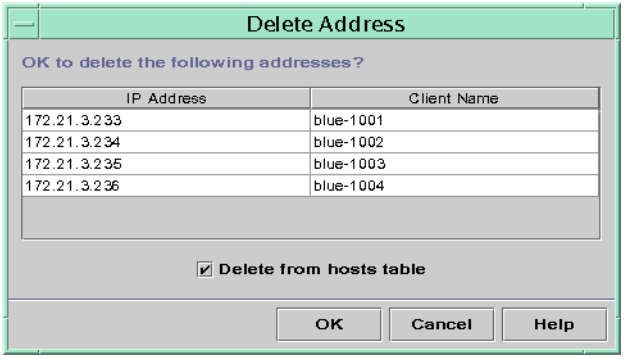
   ```
   pntadm -M 10.64.3.3 -f UNUSABLE 10.64.3.0
   ```

从 DHCP 服务中删除 IP 地址

如果您不再希望 IP 地址由 DHCP 进行管理，请从 DHCP 网络表中删除这些地址。您可以使用 `pntadm -D` 命令或 DHCP 管理程序的 "Delete Address"（删除地址）对话框。

下图显示了 "Delete Address"（删除地址）对话框。

图 15–12 DHCP 管理程序中的 "Delete Address"（删除地址）对话框



▼ 如何从 DHCP 服务中删除 IP 地址（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择 IP 地址所在的网络。**

3. **选择一个或多个要删除的 IP 地址。**

   如果您要删除多个地址，请在按住 Ctrl 键的同时用鼠标单击以选择多个地址。您还可以在按住 Shift 键的同时进行单击以选择一系列地址。

4. **从 "Edit"（编辑）菜单中选择 "Delete"（删除）。**

   "Delete Address"（删除地址）对话框将列出所选的地址，以便您可以确认删除操作。

5. **如果您要从主机表中删除主机名，请选择 "Delete From Hosts Table"（从主机表中删除）。**

   如果主机名由 DHCP 管理程序生成，您可能需要从主机表中删除主机名。

6 **单击 "OK"（确定）。**

▼ 如何从 DHCP 服务中删除 IP 地址 (pntadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来删除 IP 地址：**

```
# pntadm -D ip-address options network-address
```

如果包括 `-y` 选项，则会从维护主机名的名称服务中删除主机名。

例如，要从网络 `10.64.3.0` 中删除地址 `10.64.3.3`，并且删除相应的主机名，请键入：

```
pntadm -D 10.64.3.3 -y 10.64.3.0
```

为 DHCP 客户机指定保留的 IP 地址

DHCP 服务尝试为先前通过 DHCP 获取地址的客户机提供相同的 IP 地址。但是，有时地址已重新指定给其他客户机。

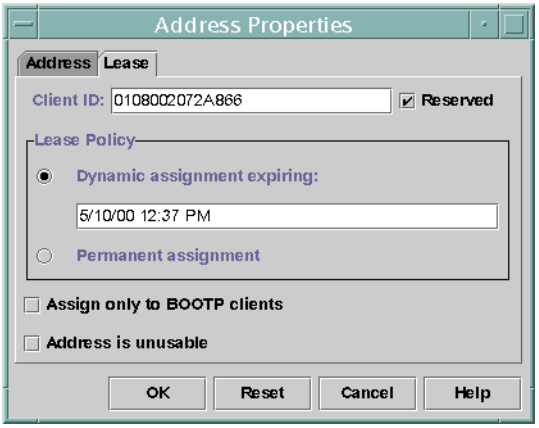
对于网络至关重要的路由器、NIS 或 NIS+ 服务器、DNS 服务器以及其他主机都不应作为 DHCP 客户机。为网络提供服务的主机不应依赖于网络来获取其 IP 地址。打印服务器或文件服务器之类的客户机也应当具有固定的 IP 地址。这些客户机可以从 DHCP 服务器接收其网络配置，并且还指定有固定的 IP 地址。

您可以将 DHCP 服务器设置为在客户机每次请求配置时都为此客户机提供相同的 IP 地址。通过根据您希望客户机使用的地址手动指定客户机的 ID，您可以为客户机保留 IP 地址。您可以将保留的地址设置为使用动态租用或永久性租用。如果客户机的地址使用动态租用，则可以轻松跟踪地址的使用。无盘客户机便是一个应当使用保留的地址以及动态租用的客户机示例。如果客户机的地址使用永久性租用，则不能跟踪地址的使用。客户机在获取永久性租用之后便不会再与服务器联系。客户机仅通过释放 IP 地址并重新启动 DHCP 租用协商，便可获取更新的配置信息。

您可以使用 `pntadm -M` 命令或 DHCP 管理程序的 "Address Properties"（地址属性）对话框来设置租用属性。

下图显示了 "Address Properties"（地址属性）对话框中用于修改租用的 "Lease"（租用）选项卡。

图 15–13 DHCP 管理程序中 "Address Properties"（地址属性）的 "Lease"（租用）选项卡



▼ 如何为 DHCP 客户机指定相同的 IP 地址（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Addresses"（地址）选项卡。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择适当的网络。**

3. **双击您希望客户机使用的 IP 地址。**
   将打开 "Address Properties"（地址属性）窗口。

4. **选择 "Lease"（租用）选项卡。**

5. **在 "Client ID"（客户机 ID）字段中，键入客户机 ID。**
   客户机 ID 由客户机的硬件地址派生而来。有关更多信息，请参见表 15–4 中的 "Client ID"（客户机 ID）项。

6. **选择 "Reserved"（保留的）选项以防止服务器回收 IP 地址。**

7. **在窗口的 "Lease Policy"（租用策略）区域中，选择 "Dynamic"（到期动态分配）或 "Permanent"（永久分配）。**
   如果您希望客户机通过协商来更新租用，请选择 "Dynamic"（到期动态分配），从而可以跟踪使用地址的时间。由于您选择了 "Reserved"（保留的），因此即使指定了动态租用，也无法回收地址。您不需要为此租用指定失效日期。DHCP 服务器将使用租用时间来计算失效日期。

如果您选择 "Permanent"（永久分配），则无法跟踪 IP 地址的使用，除非启用事务日志。

8 **单击 "OK"（确定）。**

▼ 如何为 DHCP 客户机指定相同的 IP 地址 (pntadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来设置租用标志：**

```
# pntadm -M ip-address -i client-id -f MANUAL+BOOTP network-address
```

例如，要使 MAC 地址为 `08:00:20:94:12:1E` 的 DHCP 客户机始终收到 IP 地址 `10.21.5.12`，请键入：

```
pntadm -M 10.21.5.12 -i 0108002094121E -f MANUAL+BOOTP 10.21.0.0
```

提示 – 有关如何确定客户机标识符的更多信息，请参阅表 15–4 中的 "Client ID"（客户机 ID）项。

使用 DHCP 宏（任务列表）

DHCP **宏**是 DHCP 选项的容器。DHCP 服务使用宏来收集应传递到客户机的选项。当您配置服务器时，DHCP 管理程序和 `dhcpconfig` 实用程序便会自动创建多个宏。有关宏的背景信息，请参见第 273 页中的 “关于 DHCP 宏”。有关缺省情况下创建的宏的信息，请参见第 14 章，配置 DHCP 服务（任务）。

您可能会发现：当网络发生更改时，需要对传递到客户机的配置信息进行更改。要更改配置信息，需要使用 DHCP 宏。您可以查看、创建、修改、复制以及删除 DHCP 宏。

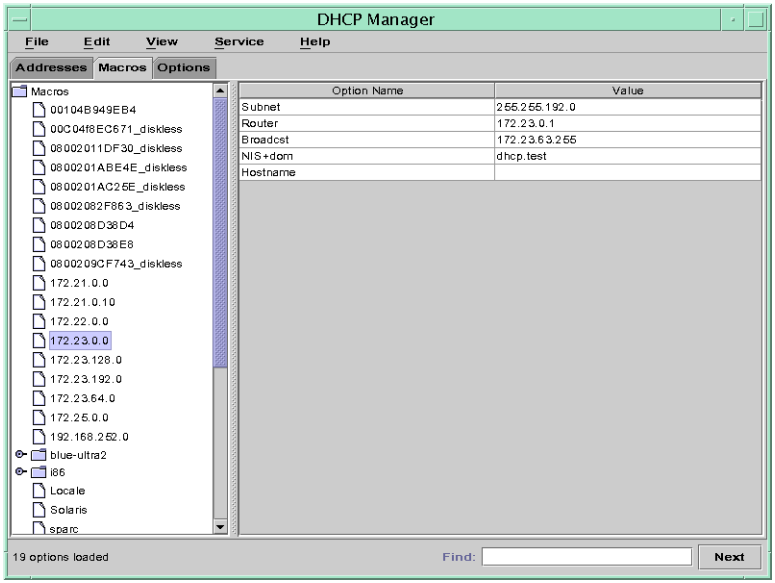
使用宏时，您必须了解 DHCP 标准选项，这些选项在 `dhcp_inittab(4)` 手册页中进行了介绍。

以下任务列表列出了可帮助您查看、创建、修改和删除 DHCP 宏的任务。此列表还包括指向详细说明如何完成每项任务的章节的链接。

任务	说明	参考
查看 DHCP 宏。	显示在 DHCP 服务器上定义的所有宏的列表。	第 345 页中的 “如何查看在 DHCP 服务器上定义的宏（DHCP 管理程序）” 第 346 页中的 “如何查看在 DHCP 服务器上定义的宏 (dhtadm)”
创建 DHCP 宏。	创建新的宏以支持 DHCP 客户机。	第 351 页中的 “如何创建 DHCP 宏（DHCP 管理程序）” 第 352 页中的 “如何创建 DHCP 宏 (dhtadm)”
修改通过宏传递到 DHCP 客户机的值。	通过修改现有选项，将选项添加到宏或从宏中删除选项来更改宏。	第 347 页中的 “如何在 DHCP 宏中更改选项的值（DHCP 管理程序）” 第 348 页中的 “如何在 DHCP 宏中修改选项的值 (dhtadm)” 第 348 页中的 “如何将选项添加到 DHCP 宏（DHCP 管理程序）” 第 349 页中的 “如何将选项添加到 DHCP 宏 (dhtadm)” 第 349 页中的 “如何从 DHCP 宏中删除选项（DHCP 管理程序）” 第 350 页中的 “如何从 DHCP 宏中删除选项 (dhtadm)”
删除 DHCP 宏。	删除不再使用的 DHCP 宏。	第 353 页中的 “如何删除 DHCP 宏（DHCP 管理程序）” 第 353 页中的 “如何删除 DHCP 宏 (dhtadm)”

下图显示了 DHCP 管理程序窗口中的 "Macros"（宏）选项卡。

图 15–14 DHCP 管理程序的 "Macros"（宏）选项卡



▼ 如何查看在 DHCP 服务器上定义的宏（DHCP 管理程序）

1 **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

在此窗口左侧的 "Macro"（宏）区域中，按字母顺序显示了在 DHCP 服务器上定义的所有宏。前面标有文件夹图标的宏包括对其他宏的引用，而前面标有文档图标的宏则不引用其他宏。

2 **要打开宏文件夹，请单击文件夹图标左侧的句柄图标。**

将列出选定宏中包含的宏。

3 **要查看宏的内容，请单击宏名称。**

将显示选项以及为其指定的值。

▼ 如何查看在 DHCP 服务器上定义的宏 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入以下命令来显示宏：**

```
# dhtadm -P
```

此命令以标准输出的形式列显已格式化的 `dhcptab` 表内容，其中包括在 DHCP 服务器上定义的所有宏和符号。

修改 DHCP 宏

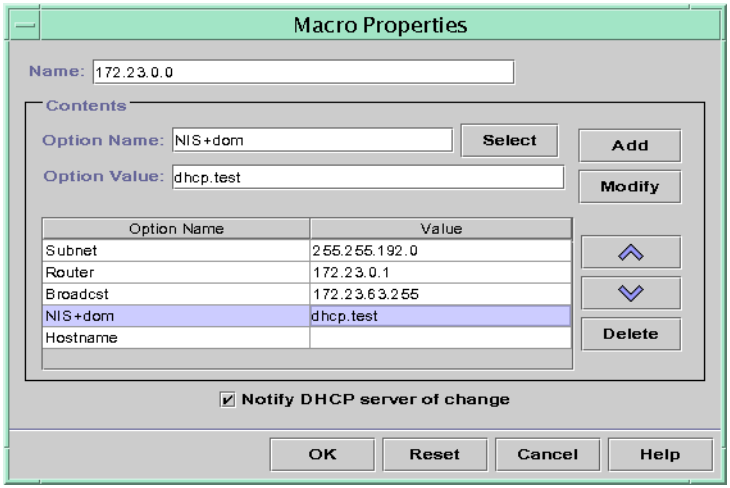
当网络在某些方面发生更改并且一台或多台 DHCP 客户机需要了解该更改时，您可能需要修改宏。例如，您可能会添加路由器或 NIS 服务器、创建新的子网或者更改租用策略。

在修改宏之前，请确定要更改、添加或删除的 DHCP 选项的名称。标准 DHCP 选项在 DHCP 管理程序帮助和 `dhcp_inittab(4)` 手册页中列出。

可以使用 `dhtadm -M -m` 命令或 DHCP 管理程序来修改宏。有关 `dhtadm` 的更多信息，请参见 `dhtadm(1M)` 手册页。

下图显示了 DHCP 管理程序的 "Macro Properties"（宏属性）对话框。

图 15–15 DHCP 管理程序中的 "Macro Properties"（宏属性）对话框



▼ 如何在 DHCP 宏中更改选项的值（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择要更改的宏。**

3. **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**
   将打开 "Macro Properties"（宏属性）对话框。

4. **在 "Options"（选项）表中，选择要更改的选项。**
   选项的名称和值将分别显示在 "Option Name"（选项名称）和 "Option Value"（选项值）字段中。

5. **在 "Option Value"（选项值）字段中，针对选项选择旧值并键入新值。**

6. **单击 "Modify"（修改）。**
   新值将显示在选项表中。

7. **选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**
   此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

8. **单击 "OK"（确定）。**

▼ 如何在 DHCP 宏中修改选项的值 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来更改选项值：**

```
# dhtadm -M -m macroname -e 'option=value:option=value' -g
```

例如，要在宏 `bluenote` 内更改租用时间和 "Universal Time Offset"（通用时间偏移），请键入：

```
# dhtadm -M -m bluenote -e 'LeaseTim=43200:UTCOffst=28800' -g
```

▼ 如何将选项添加到 DHCP 宏（DHCP 管理程序）

1 **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **选择要更改的宏。**

3 **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**

将打开 "Macro Properties"（宏属性）对话框。

4 **在 "Option Name"（选项名称）字段中，使用以下方法之一指定选项的名称：**

- **单击 "Option Name"（选项名称）字段旁边的 "Select"（选择）按钮来选择要添加到宏中的选项。**

  "Select Option"（选择选项）对话框将显示按字母顺序排列标准类别选项名称及其说明的列表。如果您要添加的选项不在标准类别中，请使用 "Category"（类别）列表来选择一个类别。

  有关宏类别的更多信息，请参见第 273 页中的 “关于 DHCP 宏”。

- **如果您要使新的宏包含对现有宏的引用，请键入 `Include`。**

5 **在 "Option Value"（选项值）字段中键入选项的值。**

如果您键入了 **Include** 作为选项名称，则必须在 "Option Value"（选项值）字段中指定现有宏的名称。

6 **单击 "Add"（添加）。**

选项将添加到此宏的选项列表的底部。要更改选项在宏中的位置，请选择此选项并单击箭头按钮，以便在列表中上下移动它。

7 **选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**

此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

8 **单击 "OK"（确定）。**

▼ 如何将选项添加到 DHCP 宏 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令将选项添加到宏：**

```
# dhtadm -M -m macroname -e 'option=value' -g
```

例如，要在宏 `bluenote` 内添加协商租用的功能，请键入以下命令：

```
# dhtadm -M -m bluenote -e 'LeaseNeg=_NULL_VALUE' -g
```

请注意，如果某选项不需要值，则必须使用 `_NULL_VALUE` 作为此选项的值。

▼ 如何从 DHCP 宏中删除选项（DHCP 管理程序）

1 **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **选择要更改的宏。**

3 **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**
将打开 "Macro Properties"（宏属性）对话框。

4 **选择要从宏中删除的选项。**

5 **单击 "Delete"（删除）。**
选项将从此宏的选项列表中删除。

6 **选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**
此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

7 **单击 "OK"（确定）。**

▼ 如何从 DHCP 宏中删除选项 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**
有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令从宏中删除选项：**

```
# dhtadm -M -m macroname -e 'option=' -g
```

例如，要从宏 `bluenote` 内删除协商租用的功能，请键入以下命令：

```
# dhtadm -M -m bluenote -e 'LeaseNeg=' -g
```

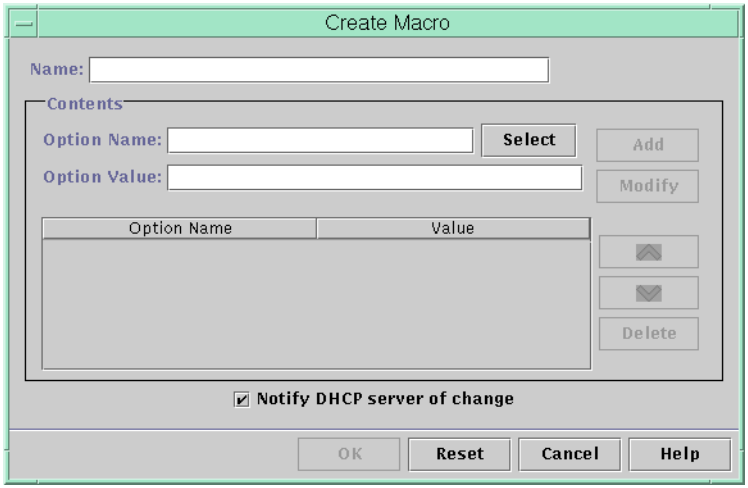
如果指定了不带值的选项，则会从宏中删除此选项。

创建 DHCP 宏

要支持具有特定需求的客户机，您可能需要向 DHCP 服务中添加新宏。您可以使用 `dhtadm -A -m` 命令或 DHCP 管理程序的 "Create Macro"（创建宏）对话框来添加宏。有关 `dhtadm` 命令的更多信息，请参见 `dhtadm(1M)` 手册页。

下图显示了 DHCP 管理程序的 "Create Macro"（创建宏）对话框。

图 15–16 DHCP 管理程序中的 "Create Macro"（创建宏）对话框



▼ 如何创建 DHCP 宏（DHCP 管理程序）

1 **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **从 "Edit"（编辑）菜单中选择 "Create"（创建）。**

将打开 "Create Macro"（创建宏）对话框。

3 **为宏键入唯一名称。**

名称最多可包含 128 个字母数字字符。如果您使用的名称与供应商类标识符、网络地址或客户机 ID 相匹配，则会针对适当的客户机自动处理宏。如果您使用其他名称，则不会自动处理宏。宏必须指定给特定 IP 地址或者包含在另一个自动处理的宏中。有关更多详细信息，请参见第 273 页中的 “DHCP 服务器的宏处理”。

4 **单击 "Option Name"（选项名称）字段旁边的 "Select"（选择）按钮。**

"Select Option"（选择选项）对话框将显示按字母顺序排列标准类别选项名称及其说明的列表。如果您要添加的选项不在标准类别中，请使用 "Category"（类别）列表。从 "Category"（类别）列表中选择所需的类别。有关选项类别的更多信息，请参见第 273 页中的 “关于 DHCP 选项”。

5 **选择要添加到宏中的选项，然后单击 "OK"（确定）。**

"Macro Properties"（宏属性）对话框将在 "Option Name"（选项名称）字段中显示选定的选项。

6 **在 "Option Value"（选项值）字段中键入选项的值，然后单击 "Add"（添加）。**

选项将添加到此宏的选项列表的底部。要更改选项在宏中的位置，请选择此选项并单击箭头按钮，以便在列表中上下移动它。

7 **对于每个要添加到宏中的选项，重复执行步骤 5 和步骤 6。**

8 **完成添加选项时，请选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**

此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

9 **单击 "OK"（确定）。**

▼ 如何创建 DHCP 宏 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来创建宏：**

```
# dhtadm -A -m macroname -d ':option=value:option=value:option=value:' -g
```

可以在 *d* 的参数中包括任意数量的 `-option=value` 对。参数必须以冒号开头和结束，并且在各个 *option=value* 对之间也加有冒号。完整字符串必须括在引号中。

例如，要创建宏 `bluenote`，请键入以下命令：

```
# dhtadm -A -m bluenote -d ':Router=10.63.6.121\
:LeaseNeg=_NULL_VALUE:DNSserv=10.63.28.12:' -g
```

请注意，如果某选项不需要值，则必须使用 `_NULL_VALUE` 作为此选项的值。

删除 DHCP 宏

您可能需要从 DHCP 服务中删除宏。例如，如果您从 DHCP 服务中删除网络，则还可以删除关联的网络宏。

您可以使用 `dhtadm -D -m` 命令或 DHCP 管理程序来删除宏。

▼ 如何删除 DHCP 宏（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Macros"（宏）选项卡。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择要删除的宏。**

   "Delete Macro"（删除宏）对话框将提示您确认要删除指定的宏。

3. **选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**

   此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

4. **单击 "OK"（确定）。**

▼ 如何删除 DHCP 宏 (dhtadm)

1. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **通过键入如下格式的命令来删除宏：**

   ```
   # dhtadm -D -m macroname -g
   ```

   例如，要删除宏 `bluenote`，请键入以下命令：

   ```
   # dhtadm -D -m bluenote -g
   ```

使用 DHCP 选项（任务列表）

选项是 DHCP 服务器可传递到客户机的网络配置参数的关键字。在 DHCP 服务中，您无法创建、删除或修改标准 DHCP 选项。标准选项由 DHCP 协议定义，因此不能对这些选项进行更改。您只能针对为站点创建的选项执行任务。因此，当您首次设置 DHCP 服务时，DHCP 管理程序中的 "Options"（选项）选项卡为空，直到针对站点创建选项。

如果您在 DHCP 服务器上创建选项，则还必须在 DHCP 客户机上添加有关这些选项的信息。对于 DHCP 客户机，您必须编辑 `/etc/dhcp/inittab` 文件以针对新选项添加项。有关此文件的更多信息，请参见 `dhcp_inittab(4)` 手册页。

如果您的 DHCP 客户机不是 Oracle Solaris 客户机，请参阅这些客户机的文档以获取有关添加选项或符号的信息。有关 DHCP 中选项的更多信息，请参见第 273 页中的 “关于 DHCP 选项”。

您可以使用 DHCP 管理程序或 `dhtadm` 命令来创建、修改或删除选项。

提示 – 在 DHCP 介绍中，选项被称为**符号**。`dhtadm` 命令及其相关的手册页也将选项称为符号。

以下任务列表列出了创建、修改和删除 DHCP 选项所需执行的任务。此任务列表包含指向这些任务的过程的链接。

任务	说明	参考
创建 DHCP 选项。	添加新选项以获取标准 DHCP 选项不包含的信息。	第 357 页中的 “如何创建 DHCP 选项（DHCP 管理程序）” 第 358 页中的 “如何创建 DHCP 选项 (dhtadm)” 第 362 页中的 “修改 DHCP 客户机的选项信息”
修改 DHCP 选项。	更改已创建的 DHCP 选项的属性。	第 359 页中的 “如何修改 DHCP 选项属性（DHCP 管理程序）” 第 360 页中的 “如何修改 DHCP 选项属性 (dhtadm)”
删除 DHCP 选项。	删除已创建的 DHCP 选项。	第 361 页中的 “如何删除 DHCP 选项（DHCP 管理程序）” 第 361 页中的 “如何删除 DHCP 选项 (dhtadm)”

在创建 DHCP 选项之前，您应该熟悉下表中列出的选项属性。

表 15–5 DHCP 选项属性

选项属性	说明
Category（类别）	选项的 *category*（类别）必须属于以下各项之一： ■ Vendor（供应商）－特定于客户机的供应商平台（硬件或软件）的选项。 ■ Site（站点）－特定于站点的选项。 ■ Extend（扩展）－已添加到 DHCP 协议中，但尚未在 DHCP 中作为标准选项实现的较新选项。
Code（代码）	*code*（代码）是指定给选项的唯一数字。选项类别中的其他任何选项都不能使用相同的代码。代码必须适用于选项类别： ■ Vendor（供应商）－对于每个供应商类，代码值为 1–254 ■ Site（站点）－代码值为 128–254 ■ Extend（扩展）－代码值为 77–127
Data type（数据类型）	*data type*（数据类型）指定哪种数据类型可以作为选项值进行指定。以下列表中介绍了有效的数据类型。 ■ ASCII－文本字符串值。 ■ BOOLEAN－任何值都不与布尔数据类型关联。选项存在表示条件为真，而选项不存在则表示条件为假。例如，`Hostname` 选项为布尔选项。如果宏中存在 `Hostname`，则会使 DHCP 服务器查找与指定的地址关联的主机名。 ■ IP－一个或多个采用点分十进制格式 (*xxx.xxx.xxx.xxx*) 的 IP 地址。 ■ OCTET－未解释的二进制数据 ASCII 表示形式。例如，客户机 ID 使用八位字节数据类型。有效字符包括 0–9、A–F 和 a–f。要表示一个 8 位数，需要使用两个 ASCII 字符。 ■ UNUMBER8、UNUMBER16、UNUMBER32、UNUMBER64、SNUMBER8、SNUMBER16、SNUMBER32 或 SNUMBER64－数值。初始的 U 或 S 表示数字是无符号数还是带符号数。末尾数字指明数字中的二进制位数。
Granularity（粒度）	*granularity*（粒度）指定需要使用某一数据类型的多少“实例”来表示一个完整的选项值。例如，如果数据类型为 IP 并且粒度为 2，则表示选项值必须包含两个 IP 地址。
Maximum（最大值）	可以为选项指定的最大值数目。例如，假设最大值为 2，粒度为 2，并且数据类型为 IP。在这种情况下，选项值最多可以包含两对 IP 地址。

表 15–5 DHCP 选项属性 （续）

选项属性	说明
Vendor client classes（供应商客户机类）	仅当选项类别为 "Vendor"（供应商）时，此选项才可用。供应商客户机类标识与 "Vendor"（供应商）选项关联的客户机类。类为表示客户机类型或操作系统的 ASCII 字符串。例如，某些型号的 Sun 工作站的类字符串为 `SUNW.Sun-Blade-100`。使用此选项类型，可以定义传递到属于同一类的所有客户机（而且**只有**此类的客户机）的配置参数。 您可以指定多个客户机类。只有那些客户机类值与所指定的类相匹配的 DHCP 客户机才能收到此类范围内的选项。 客户机类由 DHCP 客户机的供应商确定。对于不属于 Oracle Solaris 客户机的 DHCP 客户机，请参阅 DHCP 客户机的供应商文档以了解客户机类。 对于 Oracle Solaris 客户机，可以在客户机上键入 `prtconf -b` 命令来获取供应商客户机类。要指定供应商客户机类，请使用句点替换由 `uname` 命令返回的字符串中的所有逗号。例如，如果 `prtconf -b` 命令返回了字符串 `SUNW,Sun-Blade-100`，则应将供应商客户机类指定为 `SUNW.Sun-Blade-100`。

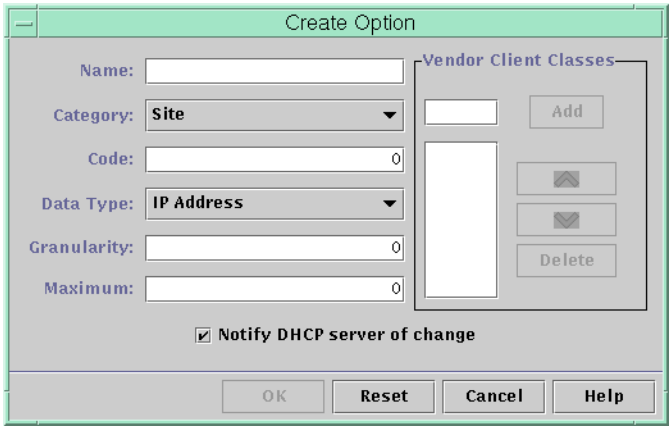
创建 DHCP 选项

如果当前在 DHCP 协议中没有可用于传递客户机信息的选项，则可以创建 DHCP 选项。在创建自己的选项之前，请参见 `dhcp_inittab(4)` 手册页，以获取在 DHCP 中定义的所有选项的列表。

您可以使用 `dhtadm -A -s` 命令或 DHCP 管理程序的 "Create Option"（创建选项）对话框来创建新选项。

下图显示了 DHCP 管理程序的 "Create Option"（创建选项）对话框。

图 15–17 DHCP 管理程序中的 "Create Option"（创建选项）对话框



▼ 如何创建 DHCP 选项（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Options"（选项）选项卡。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **从 "Edit"（编辑）菜单中选择 "Create"（创建）。**
   将打开 "Create Options"（创建选项）对话框。

3. **为新选项键入一个描述性的短名称。**
   名称最多可包含 128 个字母数字字符和空格。

4. **针对此对话框中的每个设置键入或选择值。**
   有关每个设置的信息，请参阅表 15–5 或查看 DHCP 管理程序帮助。

5. **如果完成创建选项，请选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**
   此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

6. **单击 "OK"（确定）。**
   现在，您可以将选项添加到宏，并为要传递到客户机的选项指定值。

▼ 如何创建 DHCP 选项 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入采用如下格式的命令来创建 DHCP 选项：**

```
# dhtadm -A -s option-name -d 'category,code,data-type,granularity,maximum' -g
```

option-name 是一个最多包含 128 个字符的字母数字字符串。

category 是以下各项之一：`Site`、`Extend` 或 `Vendor=`*list-of-classes*。*list-of-classes* 是要将选项应用到的供应商客户机类的空格分隔列表。有关如何确定供应商客户机类的信息，请参见表 15–5。

code 是一个适用于选项类别的数值，如表 15–5 中所述。

data-type 由一个关键字指定，表示与选项一起传递的数据的类型，如表 15–5 中所述。

granularity 指定为非负数，如表 15–5 中所述。

maximum 是一个非负数，如表 15–5 中所述。

示例 15–3 使用 dhtadm 创建 DHCP 选项

以下命令将创建名为 `NewOpt` 的选项，这是一个 "Site"（站点）类别选项。选项的代码为 130。选项的值可以设置为一个 8 位的无符号整数。

```
# dhtadm -A -s NewOpt -d 'Site,130,UNUMBER8,1,1' -g
```

以下命令将创建名为 `NewServ` 的选项，这是一个 "Vendor"（供应商）类别选项，应用于计算机类型为 `SUNW,Sun-Blade-100` 或 `SUNW,Sun-Blade-1000` 的客户机。选项的代码为 200。选项的值可以设置为一个 IP 地址。

```
# dhtadm -A -s NewServ -d 'Vendor=SUNW.Sun-Blade-100 \ SUNW.Sun-Blade-1000,200,IP,1,1' -g
```

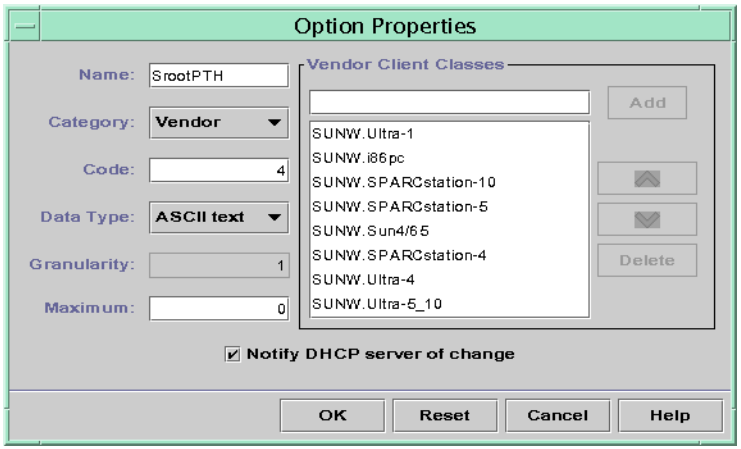
修改 DHCP 选项

如果您已为 DHCP 服务创建了选项，则可以更改这些选项的属性。您可以使用 `dhtadm -M -s` 命令或 DHCP 管理程序的 "Option Properties"（选项特性）对话框来修改选项。

请注意，您应当修改 DHCP 客户机的选项信息，以反映对 DHCP 服务所做的修改。请参见第 362 页中的 “修改 DHCP 客户机的选项信息”。

下图显示了 DHCP 管理程序的 "Option Properties"（选项特性）对话框。

图 15–18 DHCP 管理程序中的 "Option Properties"（选项特性）对话框



▼ 如何修改 DHCP 选项属性（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Options"（选项）选项卡。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择要修改的选项。**

3. **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**
   将打开 "Option Properties"（选项特性）对话框。

4. **根据需要编辑属性。**
   有关属性的信息，请参见表 15–5 或查看 DHCP 管理程序帮助。

5. **完成对选项执行的操作时，请选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**
   将对 `dhcptab` 表进行更改。将发送信号通知 DHCP 服务器重新读取 `dhcptab` 表以使更改生效。

6 **单击 "OK"（确定）。**

▼ 如何修改 DHCP 选项属性 (dhtadm)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入采用如下格式的命令来修改选项：**

```
# dhtadm -M -s option-name -d 'category,code,data-type,granularity,maximum' -g
```

option-name 指定要更改的选项的名称。

category 可以为 `Site`、`Extend` 或 `Vendor=`*list-of-classes*。*list-of-classes* 是要将选项应用到的供应商客户机类的空格分隔列表。例如，`SUNW.Sun-Blade-100 SUNW.Ultra-80 SUNWi86pc`。

code 指定适用于选项类别的数值，如表 15–5 中所述。

data-type 指定一个关键字来表示与选项一起传递的数据的类型，如表 15–5 中所述。

granularity 是一个非负数，如表 15–5 中所述。

maximum 是一个非负数，如表 15–5 中所述。

请注意，您必须使用 `-d` 开关指定所有的 DHCP 选项属性，而不是仅指定要更改的属性。

示例 15–4 使用 `dhtadm` 修改 DHCP 选项

以下命令将修改名为 `NewOpt` 的选项。此选项为 "Site"（站点）类别选项。选项的代码为 135。选项的值可以设置为一个 8 位的无符号整数。

```
# dhtadm -M -s NewOpt -d 'Site,135,UNUMBER8,1,1'
```

以下命令将修改名为 `NewServ` 的选项，这是一个 "Vendor"（供应商）类别选项。现在，此选项应用于计算机类型为 `SUNW,Sun-Blade-100` 或 `SUNW,i86pc` 的客户机。选项的代码为 200。选项的值可以设置为一个 IP 地址。

```
# dhtadm -M -s NewServ -d 'Vendor=SUNW.Sun-Blade-100 \ SUNW.i86pc,200,IP,1,1' -g
```

删除 DHCP 选项

您无法删除标准 DHCP 选项。但是，如果您已为 DHCP 服务定义了选项，则可以使用 DHCP 管理程序或 `dhtadm` 命令来删除这些选项。

▼ 如何删除 DHCP 选项（DHCP 管理程序）

1. **在 DHCP 管理程序中，选择 "Options"（选项）选项卡。**

   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **选择要删除的选项。**

3. **从 "Edit"（编辑）菜单中选择 "Delete"（删除）。**

   将打开 "Delete Option"（删除选项）对话框。

4. **如果完成删除选项，请选择 "Notify DHCP Server of Change"（将更改通知 DHCP 服务器）。**

   此选择将告知 DHCP 服务器重新读取 `dhcptab` 表，以使更改在单击 "OK"（确定）之后立即生效。

5. **单击 "OK"（确定）。**

▼ 如何删除 DHCP 选项 (dhtadm)

1. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **通过键入采用如下格式的命令来删除 DHCP 选项：**

   ```
   # dhtadm -D -s option-name -g
   ```

修改 DHCP 客户机的选项信息

如果您向 DHCP 服务器添加了新的 DHCP 选项，则必须向每个 DHCP 客户机的选项信息中添加一个补充项。如果客户机不是 DHCP 客户机，请参阅该客户机的文档以了解有关如何添加选项或符号的信息。

在 DHCP 客户机上，您必须编辑 `/etc/dhcp/inittab` 文件，并针对每个要添加到 DHCP 服务器的选项添加一项。如果您随后在此服务器上修改选项，则还必须在客户机的 `/etc/dhcp/inittab` 文件中修改此项。

有关 `/etc/dhcp/inittab` 文件语法的详细信息，请参阅 `dhcp_inittab(4)` 手册页。

注 – 如果您向先前的 Oracle Solaris 发行版的 `dhcptags` 文件中添加了 DHCP 选项，则必须将这些选项添加到 `/etc/dhcp/inittab` 文件中。有关更多信息，请参见第 419 页中的 “DHCP 选项信息”。

支持使用 DHCP 服务安装 Oracle Solaris 网络

您可以使用 DHCP 在网络中的特定客户机系统上安装 Oracle Solaris。只有满足运行 Oracle Solaris 的硬件要求的基于 `sun4u` 的系统和 x86 系统才能使用此功能。有关使用 DHCP 在客户机系统引导时针对网络对其进行自动配置的信息，请参见《Oracle Solaris 10 1/13 安装指南：基于网络的安装》中的第 2 章 “预配置系统配置信息（任务）”。

DHCP 还支持通过广域网 (Wide Area Network, WAN) 使用 HTTP 从服务器远程引导和安装的 Oracle Solaris 客户机系统。这种远程引导和安装的方法称为 *WAN Boot* **安装**方法。使用 WAN Boot，可以通过大型的公共网络（其网络基础结构可能不受信任）将 Oracle Solaris 安装在基于 SPARC 的系统上。您可以使用带有安全功能的 WAN Boot 来保护数据的保密性和安装映像的完整性。

在使用 DHCP 并借助 WAN Boot 来远程引导和安装客户机系统之前，必须将 DHCP 服务器配置为向客户机提供以下信息：

- 代理服务器的 IP 地址
- `wanboot-cgi` 程序的位置

有关配置 DHCP 服务器以提供此信息的详细信息，请参见《Oracle Solaris 10 1/13 安装指南：基于网络的安装》中的第 2 章 “预配置系统配置信息（任务）”。有关使用 DHCP 服务器跨 WAN 引导和安装客户机系统的信息，请参见《Oracle Solaris 10 1/13 安装指南：基于网络的安装》中的第 10 章 “WAN boot（概述）”。

有关支持无盘客户机的信息，请参见第 363 页中的 “支持远程引导和无盘引导客户机（任务列表）”。

支持远程引导和无盘引导客户机（任务列表）

DHCP 服务可以支持从其他计算机（OS 服务器）远程挂载操作系统文件的 Oracle Solaris 客户机系统。此类客户机通常称为**无盘客户机**。可以将无盘客户机视为持久性远程引导客户机。每次引导无盘客户机时，此客户机必须获取承载客户机操作系统文件的服务器的名称和 IP 地址。然后，无盘客户机便可通过这些文件进行远程引导。

每台无盘客户机都在 OS 服务器上拥有自己的根分区，其共享名称为客户机主机名。DHCP 服务器必须始终将相同的 IP 地址返回到无盘客户机。必须将此地址映射到名称服务（例如 DNS）中的同一主机名。当无盘客户机收到相同的 IP 地址时，此客户机便会使用相同的主机名，并可访问 OS 服务器上自己的根分区。

除了提供 IP 地址和主机名之外，DHCP 服务器还可以提供无盘客户机操作系统文件的位置。但是，您必须创建选项和宏才能在 DHCP 消息包中传递信息。

以下任务列表列出了支持无盘客户机或任何其他持久性远程引导客户机所需的任务。此任务列表还提供了指向帮助您执行这些任务的过程的链接。

任务	说明	参考
在 Oracle Solaris 服务器上设置 OS 服务。	使用 `smosservice` 命令为客户机创建操作系统文件。	《Oracle Solaris 管理：基本管理》中的第 7 章 “管理无盘客户机（任务）” 另请参见 `smosservice(1M)` 手册页。
设置 DHCP 服务以支持网络引导客户机。	使用 DHCP 管理程序或 `dhtadm` 命令创建新的 "Vendor"（供应商）选项和宏，DHCP 服务器可以使用这些选项和宏将引导信息传递到客户机。 如果您已为网络安装客户机创建了选项，则只需为无盘客户机的供应商客户机类型创建宏。	《Oracle Solaris 10 1/13 安装指南：基于网络的安装》中的第 2 章 “预配置系统配置信息（任务）”
为无盘客户机指定保留的 IP 地址。	针对无盘客户机，使用 DHCP 管理程序将地址标记为保留，或者使用 `pntadm` 命令将地址标记为 `MANUAL`。	第 341 页中的 “为 DHCP 客户机指定保留的 IP 地址”
设置无盘客户机以使用 OS 服务。	使用 `smdiskless` 命令在 OS 服务器上为每台客户机添加操作系统支持。为每台客户机指定保留的 IP 地址。	《Oracle Solaris 管理：基本管理》中的第 7 章 “管理无盘客户机（任务）” 另请参见 `smdiskless(1M)` 手册页。

设置 DHCP 客户机为仅接收信息（ 任务列表 ）

在某些网络中，您可能希望 DHCP 服务仅为客户机提供配置信息。需要信息而不是租用期信息的客户机系统可以使用 DHCP 客户机发出 INFORM 消息。INFORM 消息要求 DHCP 服务器将适当的配置信息发送到客户机。

您可以将 DHCP 服务器设置为支持仅需要信息的客户机。您需要创建与承载客户机的网络相对应的空网络表。此表必须存在，这样才能使 DHCP 服务器可以对此网络中的客户机做出响应。

以下任务列表列出了支持仅信息客户机所需的任务。此任务列表还包括指向帮助您执行这些任务的过程的链接。

任务	说明	参考
创建空的网络表。	使用 DHCP 管理程序或 pntadm 命令为仅需要信息的客户机网络创建网络表。	第 320 页中的 “添加 DHCP 网络”
创建宏以包含客户机所需的信息。	使用 DHCP 管理程序或 dhtadm 命令创建宏，以便将所需的信息传递到客户机。	第 350 页中的 “创建 DHCP 宏”
使 DHCP 客户机发出 INFORM 消息。	使用 ifconfig *int* dhcp inform 命令使 DHCP 客户机发出 INFORM 消息。	第 377 页中的 “DHCP 客户机启动” 第 381 页中的 “用于 DHCP 客户机的 ifconfig 命令选项” ifconfig(1M) 手册页

转换为新的 DHCP 数据存储

DHCP 提供了一个实用程序，可以将 DHCP 配置数据从一种数据存储转换为另一种数据存储。可能存在几种要转换为新的数据存储的原因。例如，您可能具有多台 DHCP 客户机，从而需要 DHCP 服务具备更高的性能或容量。您还可能希望在多台服务器间共享 DHCP 服务器功能。有关每种数据存储类型的相对优缺点的比较，请参见 第 282 页中的 “选择 DHCP 数据存储”。

注 – 如果您要从 Solaris 8 7/01 发行版之前的 Oracle Solaris 发行版进行升级，则应当阅读此说明。

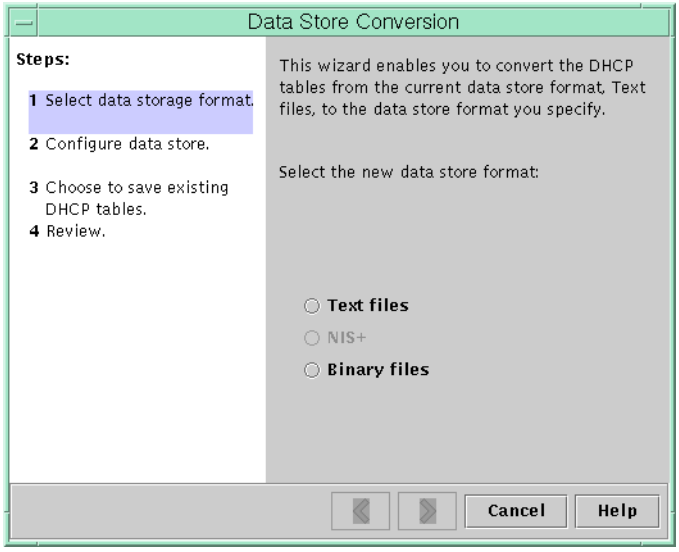
当您在安装 Oracle Solaris 之后运行任何 DHCP 工具时，系统便会提示您转换为新的数据存储库。需要进行转换的原因是存储在 Solaris 8 7/01 发行版中的文件和 NIS+ 内的数据格式发生了更改。如果没有转换为新的数据存储，则 DHCP 服务器会继续读取旧的数据表。但是，此服务器只能延长现有客户机的租用期。您无法注册新的 DHCP 客户机或者针对旧的数据表使用 DHCP 管理工具。

对于从 Sun 提供的数据存储转换为第三方数据存储的站点，转换实用程序也非常有用。转换实用程序会在现有数据存储中查找项，并将包含相同数据的新项添加到新的数据存储中。对于每个数据存储，可以在单独的模块中实现数据存储访问。使用这种模块化方法，转换实用程序可以将 DHCP 数据从任意一种数据存储格式转换为其他任何数据存储格式。每个数据存储必须具有 DHCP 服务可以使用的模块。有关如何编写模块来支持第三方数据存储的更多信息，请参见《Solaris DHCP Service Developer's Guide》 。

可以使用 DHCP 管理程序（通过 "Data Store Conversion"（数据存储转换）向导），或者使用 `dhcpconfig -C` 命令来完成数据存储转换。

下图显示了 "Data Store Conversion"（数据存储转换）向导的初始对话框。

图 15–19 DHCP 管理程序中的 "Data Store Conversion"（数据存储转换）向导对话框



在开始转换之前，您必须指定是否保存旧数据存储的表（`dhcptab` 表和网络表）。然后，转换实用程序便会停止 DHCP 服务器，转换数据存储，并在成功完成转换之后重新启动此服务器。如果您没有指定保存旧表，则此实用程序在确定转换成功之后便会删除这些表。转换过程可能会相当耗时。转换将在后台运行，通过指示器显示其进度。

▼ 如何转换 DHCP 数据存储（DHCP 管理程序）

1 **在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Convert Data Store"（转换数据存储）。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

将打开 "Data Store Conversion"（数据存储转换）向导。

2 **回答向导的提示。**

如果您在提供所需的信息时遇到问题，请单击 "Help"（帮助）以查看有关每个对话框的详细信息。

3 **查看您的选择，然后单击 "Finish"（完成）以转换数据存储。**

DHCP 服务器将在转换完成之后重新启动。此服务器会立即使用新数据存储。

▼ 如何转换 DHCP 数据存储 (dhcpconfig -C)

1 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **通过键入如下格式的命令来转换数据存储：**

```
# /usr/sbin/dhcpconfig -C -r resource -p path
```

resource	是新数据存储类型，例如 `SUNWbinfiles`
path	是数据的路径，例如 `/var/dhcp`

请注意，如果您要在转换之后保留旧数据存储中的原始数据，请指定 `-k` 选项。例如，要将数据存储转换为 `SUNWbinfiles` 并保存旧数据存储，请键入：

```
# /usr/sbin/dhcpconfig -C -r SUNWbinfiles -p /var/dhcp -k
```

有关 `dhcpconfig` 实用程序的更多信息，请参见 `dhcpconfig(1M)` 手册页。

在 DHCP 服务器之间移动配置数据（任务列表）

使用 DHCP 管理程序和 dhcpconfig 实用程序，可以在 DHCP 服务器之间移动部分或全部 DHCP 配置数据。您可以移动整个网络以及所有与网络关联的 IP 地址、宏和选项。或者，您可以选择特定的 IP 地址、宏和选项进行移动。您还可以复制宏和选项，而无需从第一台服务器中删除这些宏和选项。

如果您要执行以下任务之一，则可能需要移动数据：

- 添加要共享 DHCP 功能的服务器。
- 替换 DHCP 服务器系统。
- 更改数据存储的路径，同时仍使用同一个数据存储。

以下任务列表提供了移动 DHCP 配置数据时必须执行的过程。此列表包括指向用于执行这些任务的过程的链接。

任务	说明	参考
1. 从第一台服务器中导出数据。	选择要移动到其他服务器的数据，并为导出数据创建文件。	第 369 页中的“如何从 DHCP 服务器中导出数据（DHCP 管理程序）” 第 369 页中的“如何从 DHCP 服务器中导出数据 (dhcpconfig -X)”
2. 将数据导入第二台服务器。	将导出数据复制到其他 DHCP 服务器的数据存储。	第 370 页中的“如何在 DHCP 服务器上导入数据（DHCP 管理程序）” 第 370 页中的“如何在 DHCP 服务器上导入数据 (dhcpconfig -I)”
3. 针对新的服务器环境修改导入的数据。	将特定于服务器的配置数据更改为与新服务器的信息相匹配。	第 371 页中的“如何修改导入的 DHCP 数据（DHCP 管理程序）” 第 372 页中的“如何修改导入的 DHCP 数据 (pntadm, dhtadm)”

在 DHCP 管理程序中，您可以使用 "Export Data"（导出数据）向导和 "Import Data"（导入数据）向导在服务器之间移动数据。然后可在 "Macros"（宏）选项卡中修改宏。以下两个图显示了这两个向导的初始对话框。

图 15–20 DHCP 管理程序中的 "Export Data"（导出数据）向导对话框

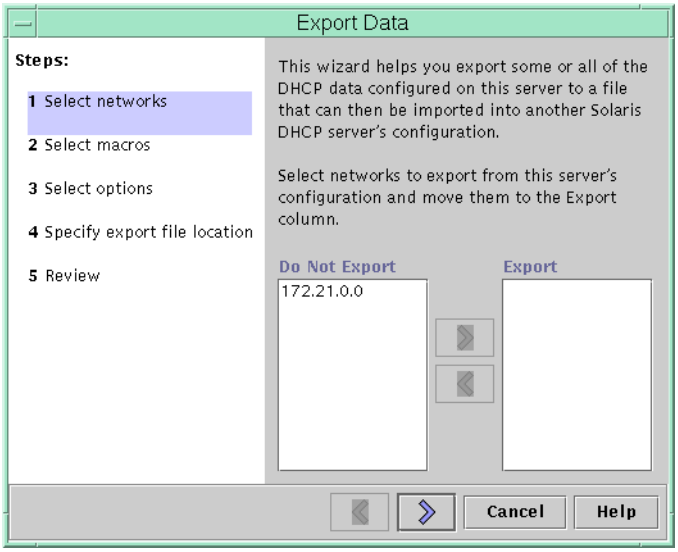
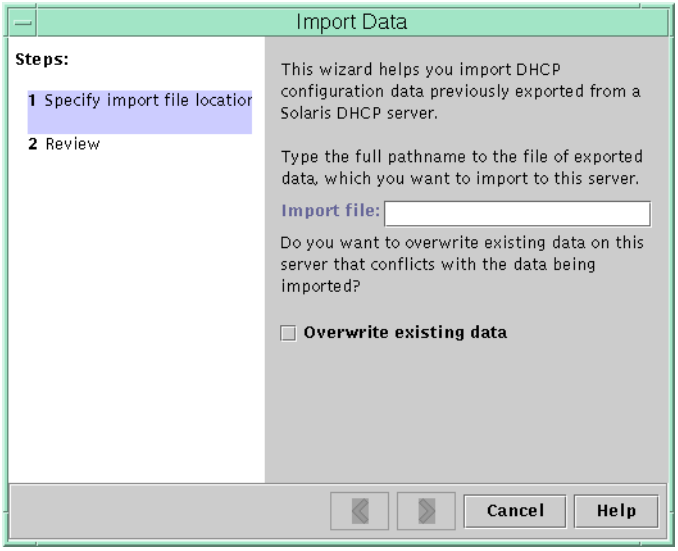


图 15–21 DHCP 管理程序中的 "Import Data"（导入数据）向导对话框



▼ 如何从 DHCP 服务器中导出数据（DHCP 管理程序）

1. **在要从其中移动或复制数据的服务器上启动 DHCP 管理程序。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的“如何启动和停止 DHCP 管理程序”。
2. **从 "Service"（服务）菜单中选择 "Export Data"（导出数据）。**
   将打开 "Export Data"（导出数据）向导，如图 15–20中所示。
3. **回答向导的提示。**
   如果遇到问题，请单击 "Help"（帮助）以获取有关提示的详细信息。
4. **将导出文件移动到必须导入文件的 DHCP 服务器可访问的文件系统中。**

另请参见 导入数据，如第 370 页中的“如何在 DHCP 服务器上导入数据（DHCP 管理程序）”中所述。

▼ 如何从 DHCP 服务器中导出数据 (dhcpconfig -X)

1. **登录到要从其中移动或复制数据的服务器。**
2. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**
   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的“设置用户访问 DHCP 命令的权限”。

   角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。
3. **导出数据。**
   您可以导出所有的 DHCP 数据，也可以导出特定的数据部分。

   - **要导出特定的地址、宏和选项，请键入采用如下格式的命令：**

     ```
     # dhcpconfig -X filename -a network-addresses -m macros -o options
     ```

     filename 是要用于存储压缩的导出数据的完整路径名。您可以指定逗号分隔列表中的特定网络地址、DHCP 宏和 DHCP 选项。以下示例显示如何导出特定的网络、宏和选项。

     ```
     # dhcpconfig -X /var/dhcp/0dhcp1065_data \ -a 10.63.0.0,10.62.0.0 \
     -m 10.63.0.0,10.62.0.0,SUNW.Sun-Blade-100 -o Sterm
     ```

- **要导出所有的 DHCP 数据，请键入使用 ALL 关键字的命令。**

  ```
  # dhcpconfig -X filename -a ALL -m ALL -o ALL
  ```

  filename 是要用于存储压缩的导出数据的完整路径名。可以结合使用关键字 ALL 与命令选项来导出所有的网络地址、宏或选项。以下示例显示如何使用 ALL 关键字。

  ```
  # dhcpconfig -X /var/dhcp/dhcp1065_data -a ALL -m ALL -o ALL
  ```

提示 – 您可以通过不为特定的数据类型指定 dhcpconfig 命令选项来避免导出此数据类型。例如，如果没有指定 -m 选项，则不会导出任何 DHCP 宏。

有关 dhcpconfig 命令的更多信息，请参见 dhcpconfig(1M) 手册页。

4. **将导出文件移动到必须导入数据的服务器可访问的位置中。**

另请参见 导入数据，如第 370 页中的 “如何在 DHCP 服务器上导入数据 (dhcpconfig -I)”中所述。

▼ 如何在 DHCP 服务器上导入数据（DHCP 管理程序）

1. **在要接收先前从 DHCP 服务器导出的数据的服务器上启动 DHCP 管理程序。**
   有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2. **从 "Service"（服务）菜单中选择 "Import Data"（导入数据）。**
   将打开 "Import Data"（导入数据）向导，如图 15–21 中所示。

3. **回答向导的提示。**
   如果遇到问题，请单击 "Help"（帮助）以获取有关提示的详细信息。

4. **如有必要，修改导入数据。**
   请参见第 371 页中的 “如何修改导入的 DHCP 数据（DHCP 管理程序）”

▼ 如何在 DHCP 服务器上导入数据 (dhcpconfig -I)

1. **登录到要将数据导入其中的服务器。**

2. **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**
   有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

3 **通过键入如下格式的命令来导入数据：**

```
# dhcpconfig -I filename
```

filename 是包含导出数据的文件的名称。

4 **如有必要，修改导入数据。**

请参见第 372 页中的 “如何修改导入的 DHCP 数据 (pntadm, dhtadm)”。

▼ 如何修改导入的 DHCP 数据（DHCP 管理程序）

1 **在已将数据导入其中的服务器上启动 DHCP 管理程序。**

有关 DHCP 管理程序的信息，请参见第 302 页中的 “如何启动和停止 DHCP 管理程序”。

2 **检查导入数据以了解需要修改的特定于网络的信息。**

例如，如果您移动了网络，则必须打开 "Addresses"（地址）选项卡，并在导入网络中更改地址的所属服务器。您还可能需要打开 "Macros"（宏）选项卡，以便在某些宏中为 NIS、NIS+ 或 DNS 指定正确的域名。

3 **打开 "Addresses"（地址）选项卡并选择已导入的网络。**

4 **要选择所有地址，请单击第一个地址，按住 Shift 键，然后单击最后一个地址。**

5 **从 "Edit"（编辑）菜单中选择 "Properties"（属性）。**

将打开 "Modify Multiple Addresses"（修改多个地址）对话框。

6 **在 "Managing Server"（管理服务器）提示符下，选择新服务器名称。**

7 **在 "Configuration Macro"（配置宏）提示符下，选择应该用于此网络上所有客户机的宏，然后单击 "OK"（确定）。**

8 **打开 "Macros"（宏）选项卡。**

9 **使用 "Find"（查找）按钮查找可能需要修改值的选项。**

"Find"（查找）按钮位于窗口的底部。

`DNSdmain`、`DNSserv`、`NISservs`、`NIS+serv` 和 `NISdmain` 便是几个可能需要在新服务器上修改的选项示例。

10 **在适当的宏中更改选项。**

有关更改选项的过程，请参见第 359 页中的 “如何修改 DHCP 选项属性（DHCP 管理程序）”。

▼ 如何修改导入的 DHCP 数据 (pntadm, dhtadm)

1 **登录到已将数据导入其中的服务器。**

2 **成为超级用户、承担指定给 DHCP 管理配置文件的角色或者使用指定给 DHCP 管理配置文件的用户名。**

有关 DHCP 管理配置文件的更多信息，请参见第 303 页中的 “设置用户访问 DHCP 命令的权限”。

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

3 **检查网络表以了解需要修改的数据。**

如果您移动了网络，请使用 `pntadm -P` *network-address* 命令针对已移动的网络列显出网络表。

4 **使用 `pntadm` 命令修改 IP 地址信息。**

您可能需要针对导入地址更改所属服务器和配置宏。例如，要针对地址 `10.63.0.2` 更改所属服务器 (`10.60.3.4`) 和宏 (`dhcpsrv-1060`)，请使用以下命令：

```
pntadm -M 10.63.0.2 -s 10.60.3.4 -m dhcpsrv-1060 10.60.0.0
```

如果具有大量地址，则应创建包含命令的脚本文件以修改每个地址。使用 `pntadm -B` 命令执行脚本，此脚本以批处理模式运行 `pntadm`。请参见 `pntadm(1M)` 手册页。

5 **检查 `dhcptab` 宏以了解需要修改值的选项。**

使用 `dhtadm -P` 命令将整个 `dhcptab` 表列显到屏幕上。使用 `grep` 或其他某种工具搜索可能要更改的选项或值。

6 **如有必要，使用 `dhtadm -M` 命令修改宏中的选项。**

例如，您可能需要修改某些宏，以便为 NIS、NIS+ 或 DNS 指定正确的域名和服务器。例如，以下命令更改宏 `mymacro` 内 `DNSdmain` 和 `DNSserv` 的值：

```
dhtadm -M -m mymacro -e 'DNSserv=dnssrv2:DNSdmain=example.net' -g
```

◆ ◆ ◆ 第 16 章

配置和管理 DHCP 客户机

本章介绍属于 Oracle Solaris 的动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 客户机。同时还说明客户机的 DHCPv4 和 DHCPv6 协议如何工作，以及您如何可以影响客户机的行为。

协议之一 DHCPv4 长期以来是 Oracle Solaris 的一部分，通过它 DHCP 服务器可以将配置参数（如 IPv4 网络地址）传递到 IPv4 节点。

通过另一协议 DHCPv6，DHCP 服务器可以将配置参数（如 IPv6 网络地址）传递到 IPv6 节点。DHCPv6 是与“IPv6 无状态地址自动配置”(RFC 2462) 对应的有状态协议。它既可以单独使用，也可以与无状态地址自动配置同时使用，来获取配置参数。

本章包含以下信息：

关于 DHCP 客户机

DHCP 客户机为 `dhcpagent` 守护进程。安装 Oracle Solaris 时，系统会提示您使用 DHCP 来配置网络接口。如果您为 DHCPv4 指定 "Yes"（是），则会在安装 Oracle Solaris 期间在系统上启用该协议。没有专用于 DHCPv6 的安装时选项。但是，有一个与 IPv6 相关的问题。如果启用 IPv6，则在支持 DHCPv6 的本地网络上也将启用 DHCPv6。

无需为了使用 DHCP 而对 Oracle Solaris 客户机执行其他任何操作。DHCP 服务器配置会确定为使用 DHCP 服务的 DHCP 客户机系统提供的信息。

如果客户机系统已经运行 Oracle Solaris，但是没有使用 DHCP，则可以将客户机系统重新配置为使用 DHCP。您还可以重新配置 DHCP 客户机系统，以使其停止使用 DHCP 而使用您提供的静态网络信息。有关更多信息，请参见第 380 页中的 “启用和禁用 DHCP 客户机”。

DHCPv6 服务器

Sun Microsystems 在 Oracle Solaris 中未提供 DHCPv6 服务器。由第三方提供的服务器与 Sun 的 DHCPv6 兼容；如果网络中存在 DHCPv6 服务器，则 Sun 的 DHCPv6 客户机将使用它。

有关 Sun DHCPv4 服务器的信息，请参见第 268 页中的 “DHCP 服务器”。

DHCPv4 和 DHCPv6 之间的差异

DHCPv4 和 DHCPv6 之间存在以下两个主要差异：

- **管理模型**
  - DHCPv4－管理员为每个接口启用 DHCP。管理基于每个逻辑接口。
  - DHCPv6－不需要显式配置。此协议在给定物理接口上启用。
- **协议详细信息**
  - DHCPv4－DHCP 服务器提供每个地址的子网掩码。主机名选项设置系统范围的节点名称。
  - DHCPv6－子网掩码由路由器通告（而不是 DHCPv6 服务器）提供。没有 DHCPv6 主机名选项。

DHCP 管理模型

DHCPv4 需要显式客户机配置。必须设置 DHCPv4 系统以便在需要时进行寻址，这通常是在初始系统安装期间完成或通过使用 `ifconfig(1M)` 选项动态完成。

DHCPv6 不需要显式客户机配置。相反，使用 DHCP 是网络的特性，并且使用 DHCP 的信号在来自本地路由器的路由器通告消息中传送。DHCP 客户机根据需要自动创建和销毁逻辑接口。

DHCPv6 机制与现有的 IPv6 无状态（自动）地址配置在管理方面非常类似。对于无状态地址配置，可以在本地路由器上设置一个标志以指明：如果有一组给定的前缀，每个客户机都应使用通告前缀以及本地接口标记或随机数独立地自动配置一个地址。对于 DHCPv6，需要相同的前缀，但地址是通过 DHCPv6 服务器而不是“随机”指定来获取和管理的。

MAC 地址和客户机 ID

为了方便指定地址，**DHCPv4** 使用 MAC 地址和可选的客户机 ID 来标识客户机。每次同一客户机到达网络时，它都会获取同一地址（如果可能）。

DHCPv6 使用基本相同的方案，但是它必须使用客户机 ID，并强制它采用特定结构。DHCPv6 中的客户机 ID 由以下两部分组成：DHCP 唯一标识符 (DHCP Unique Identifier, DUID) 和身份关联标识符 (Identity Association Identifier, IAID)。DUID 标识客户机**系统**（而不是如 DHCPv4 中那样仅标识接口），而 IAID 标识该系统上的接口。

如 RFC 3315 中所述，身份关联是服务器和客户机用来标识、分组和管理一组相关 IPv6 地址的方法。客户机必须将至少一个不同 IA 与其每个网络接口关联，然后使用指定的 IA 从服务器获取该接口的配置信息。有关 IA 的其他信息，请参见下一节“协议详细信息”。

DUID+IAID 也可以用于 DHCPv4。可以明确地将它们串联在一起，以便它们可以用作客户机 ID。由于兼容性原因，没有对常规 IPv4 接口执行此操作。但是，对于逻辑接口 (`hme0:1`)，如果未配置客户机 ID，则使用 DUID+IAID。

与 IPv4 DHCP 不同，DHCPv6 未提供 "client name"（客户机名称）选项，因此无法仅基于 DHCPv6 来命名系统。相反，如果需要知道与 DHCPv6 提供的地址相配的 DNS 名称，请使用 DNS 反向解析（通过 `getaddrinfo(3SOCKET)` 函数的地址到名称查询）查找对应的名称信息。这意味着，如果仅使用 DHCPv6 且希望节点具有特定的名称，则必须在系统上设置 `/etc/nodename`。

协议详细信息

如果使用 DHCPv4，DHCP 服务器会提供要用于已指定地址的子网掩码。如果使用 DHCPv6，子网掩码（也称为“前缀长度”）由路由器通告指定，且不受 DHCP 服务器的控制。

DHCPv4 有一个用于设置系统范围节点名称的主机名选项。DHCPv6 没有这样的选项。

要为 DHCPv6 配置客户机 ID，必须指定 DUID，而不是允许系统自动选择一个。对于守护进程，可以按全局方式执行此操作，也可以基于每个接口来执行。使用以下格式设置全局 DUID（请注意初始点）：

`.v6.CLIENT_ID=`*DUID*

将特定接口设置为使用给定 DUID（并使系统看起来是 DHCPv6 服务器的多个独立客户机）：

`hme0.v6.CLIENT ID=`*DUID*

每种身份关联 (Identity Association, IA) 持有一种类型的地址。例如，临时地址的身份关联 (Identity Association for Temporary Addresses, IA_TA) 持有临时地址，而非临时地址的身份关联 (Identity Association for Non-temporary Addresses, IA_NA) 持有指定的永久性地址。本指南介绍的 DHCPv6 版本仅提供 IA_NA 关联。

Oracle Solaris 根据需要仅为每个接口指定一个 IAID，并将 IAID 存储在根文件系统的文件中，以便它在计算机的使用期限内保持不变。

逻辑接口

在 DHCPv4 客户机中，每个逻辑接口都是独立的，而且是一个管理单元。除了第零个逻辑接口（缺省情况下是作为标识符的接口 MAC 地址）外，用户还可以通过在 `dhcpagent` 配置文件中指定 CLIENT_ID，将特定的逻辑接口配置为运行 DHCP。例如：

```
hme0:1.CLIENT_ID=orangutan
```

DHCPv6 以不同的方式工作。与 IPv4 不同，IPv6 接口上的第零个逻辑接口始终是本地链路。在没有其他可用的指定方法（如 DHCP 服务器）时，将使用本地链路自动将 IP 地址指定给 IP 网络中的设备。第零个逻辑接口不能由 DHCP 控制，因此，虽然 DHCPv6 在第零个逻辑接口（也称为“物理”接口）上运行，但是它仅在非零逻辑接口上指定地址。

作为对 DHCPv6 客户机请求的响应，DHCPv6 服务器返回客户机要配置的地址的列表。

选项协商

在 DHCPv6 中存在 "Option Request"（选项请求）选项，该选项为服务器提供关于客户机首选查看内容的提示。如果所有的可能选项都是从服务器发送到客户机，则可以发送的信息如此之多，以致于必须在发往客户机的过程中删除其中一些信息。服务器可能会使用该提示从要包括在答复中的选项中进行选择。或者，服务器可能忽略提示，而选择要包括的其他项。例如，在 Oracle Solaris 中，首选选项可能包括 Oracle Solaris DNS 地址域或 NIS 地址域，但不大可能包括网络 BIOS 服务器。

为 DHCPv4 也提供了相同类型的提示，但是未提供特殊的 "Option Request"（选项请求）选项。DHCPv4 改用 `/etc/default/dhcpagent` 中的 `PARAM_REQUEST_LIST`。

配置语法

如果使用 `/etc/default/dhcpagent`，则配置 DHCPv6 客户机的方式与配置现有 DHCPv4 客户机的方式非常类似。

通过在接口名称（如果有）和要配置的参数之间增加一个 ".v6" 标记。例如，可以按以下所示设置全局 IPv4 选项请求列表：

```
PARAM_REQUEST_LIST=1,3,6,12,15,28,43
```

可以按以下所示将单个接口配置为省略主机名选项：

```
hme0.PARAM_REQUEST_LIST=1,3,6,15,28,43
```

要为 DHCPv6 设置全局请求列表，请注意前导点：

```
.v6.PARAM_REQUEST_LIST=23,24
```

或者，要设置单个接口，请按照以下示例操作：

```
hme0.v6.PARAM_REQUEST_LIST=21,22,23,24
```

以下是 DHCPv6 配置的实际 /etc/default/dhcpagent 文件，供您参考：

```
# The default DHCPv6 parameter request list has preference (7), unicast (12),
# DNS addresses (23), DNS search list (24), NIS addresses (27), and
# NIS domain (29).  This may be changed by altering the following parameter-
# value pair.  The numbers correspond to the values defined in RFC 3315 and
# the IANA dhcpv6-parameters registry.
.v6.PARAM_REQUEST_LIST=7,12,23,24,27,29
```

DHCP 客户机启动

大多数情况下，在 DHCPv6 客户机启动时您无需执行任何操作。如果需要，in.ndpd 守护进程会自动启动 DHCPv6。您可能需要编辑 /etc/hostname6.$IFNAME 以配置在引导时要为 IPv6 检测的接口。但是，如果安装时在系统上启用了 IPv6，则安装程序已执行此操作。

不过，对于 DHCPv4，如果在 Oracle Solaris 安装过程中未执行此操作，则必须请求客户机启动。请参见第 380 页中的 “如何启用 DHCP 客户机”。

dhcpagent 守护进程可获取引导系统时所涉及的其他进程所需的配置信息。因此，系统启动脚本会在引导过程前期启动 dhcpagent 并处于等待状态，直到来自 DHCP 服务器的网络配置信息到达。

虽然缺省设置是运行 DHCPv6，但是可以选择不运行 DHCPv6。DHCPv6 开始运行后，您可以使用 ifconfig 命令停止它。也可以通过修改 /etc/inet/ndpd.conf 文件禁用 DHCPv6，以便在重新引导时它不会启动。

以下示例说明如何在名为 hme0 的接口上立即关闭 DHCPv6：

```
ex# echo ifdefault StatefulAddrConf false >> /etc/inet/ndpd.conf
ex# pkill -HUP -x in.ndpd
ex# ifconfig hme0 inet6 dhcp release
```

如果存在 /etc/dhcp.*interface* 文件（例如，Sun Fire 880 系统上的 /etc/dhcp.ce0），则意味着向启动脚本指明要在指定接口上使用 DHCPv4。找到 dhcp.*interface* 文件后，启动脚本便会启动 dhcpagent。

启动之后，dhcpagent 便会处于等待状态，直到收到配置网络接口的指令。启动脚本将发出 ifconfig *interface* dhcp start 命令，此命令指示 dhcpagent 按照第 265 页中的“DHCP 的工作原理”中所述来启动 DHCPv4。如果这些命令包含在 dhcp.*interface* 文件中，则会将它们附加到 ifconfig 的 dhcp start 选项后面。有关与 ifconfig *interface* dhcp 命令一起使用的选项的更多信息，请参见 ifconfig(1M) 手册页。

DHCPv6 通信

与通过手动配置调用的 DHCPv4 不同，DHCPv6 是通过路由器通告 (Router Advertisement, RA) 调用的。根据配置路由器的方式，系统在收到路由器通告消息的接口上自动调用 DHCPv6，并使用 DHCP 获取地址和其他参数，或者系统通过 DHCPv6 仅请求除地址之外的数据（例如 DNS 服务器）。

in.ndpd 守护进程接收路由器通告消息。在系统中为 IPv6 检测的所有接口上，它自动执行此操作。在 in.ndpd 检测到指定 DHCPv6 应该运行的 RA 时，它将调用该 RA。

要阻止 in.ndpd 启动 DHCPv6，可以更改 /etc/inet/ndpd.conf 文件。

使用以下 ifconfig 版本之一，还可以在 DHCPv6 启动后停止它：

```
ifconfig <interface>inet6 dhcp drop
```

或者：

```
ifconfig <interface> inet6 dhcp release
```

DHCP 客户机协议如何管理网络配置信息

DHCPv4 和 DHCPv6 客户机协议以不同的方式管理网络配置信息。主要差异在于，对于 DHCPv4，协商是针对单个地址和一些与之配套的选项的租用。对于 DHCPv6，协商是针对一批地址和一批选项。

有关 DHCPv4 客户机和服务器之间的交互的背景信息，请参见第 12 章，关于 DHCP（概述）。

DHCPv4 客户机如何管理网络配置信息

从 DHCP 服务器获取信息包之后，dhcpagent 便会配置网络接口并初启它。此守护进程在 IP 地址的租用时间内对接口进行控制，并在内部表中维护配置数据。系统启动脚本使用 dhcpinfo 命令从内部表中提取配置选项值。这些值用于配置系统并使其在网络上进行通信。

dhcpagent 守护进程将被动地等待一段时间，通常为租用时间的一半。然后此守护进程从 DHCP 服务器请求延长租用期。如果系统将接口关闭或 IP 地址已更改的信息通知给 dhcpagent，则守护进程不会对接口进行控制，除非 ifconfig 命令指示这样做。如果 dhcpagent 发现接口启动并且 IP 地址没有更改，则它会向服务器发送续订租用的请求。如果无法续订租用，则 dhcpagent 将在租用到期时关闭接口。

dhcpagent 每次执行与租用相关的操作时，该守护进程都会查找名为 /etc/dhcp/eventhook 的可执行文件。如果找到具有此名称的可执行文件，则 dhcpagent 将调用此可执行文件。有关使用可执行事件的更多信息，请参见第 389 页中的 “DHCP 客户机事件脚本”。

DHCPv6 客户机如何管理网络配置信息

客户机和服务器之间的 DHCPv6 通信以客户机发出要查找服务器的请求消息开始。作为响应，可用于 DHCP 服务的所有服务器都发送通告消息。服务器消息包含多个 IA_NA（Identity Association Non-Temporary Address，身份关联非临时地址）记录，以及服务器可以提供的其他选项（如 DNS 服务器地址）。

通过在其请求消息中设置自己的 IA_NA/IAADDR 记录，客户机可以请求特定地址（而且可以请求多个特定地址）。如果客户机记录了旧地址且它希望服务器提供相同的地址（如果可能），则它通常请求特定的地址。不管客户机如何工作（即使根本未请求地址），服务器都可以为单个 DHCPv6 事务向客户机提供任意数量的地址。

以下是在客户机和服务器之间发生的消息对话。

- 客户机发送要查找服务器的请求消息。
- 服务器发送通告消息，以指示它们可用于 DHCP 服务。
- 客户机发送请求消息，以便从具有最大优先级值的服务器请求配置参数（其中包括 IP 地址）。服务器优先级值由管理员设置，其范围为从 0（最小）到 255（最大）。
- 服务器发送包含地址租用和配置数据的回复消息。

如果通告消息中的优先级值为 255，则 DHCPv6 客户机立即选择该服务器。如果优先级最高的服务器没有响应，或者无法成功回复请求消息，则客户机会继续查找优先级较低（按顺序）的服务器，直到不再出现通告消息。此时，客户机开始再次发送请求消息。

所选服务器发送包含指定地址和配置参数的回复消息，以响应请求消息。

DHCP 客户机关闭

客户机在关闭时将释放消息发送到将地址指定给该客户机的服务器，以指示该客户机将不再使用一个或多个指定地址。DHCPv4 客户机系统正常关闭时，dhcpagent 会将当前配置信息写入一个文件（如果该文件存在的话）。对于 DHCPv4，该文件名为 /etc/dhcp/*interface*.dhc，对于 DHCPv6，该文件名为 /etc/dhcp/*interface* .dh6。缺省情况下，会保存而不是释放租用，因此 DHCP 服务器无法检测到 IP 地址未处于活动使用状态，这样在下次引导时客户机就可以轻松地重新获取地址了。此缺省操作相当于执行 ifconfig <interface> dhcp drop 命令。

如果重新引导系统时该文件中的租用仍然有效，则 dhcpagent 将发送要求使用相同 IP 地址和网络配置信息的简短请求。对于 DHCPv4，这是 "Request"（请求）消息。对于 DHCPv6，则消息为 "Confirm"（确认）消息。

如果 DHCP 服务器允许此请求，则 dhcpagent 可以使用它在系统关闭时写入磁盘的信息。如果服务器不允许客户机使用此信息，则 dhcpagent 将启动第 265 页中的 “DHCP 的工作原理”中所述的 DHCP 协议序列。因此，客户机将获取新的网络配置信息。

启用和禁用 DHCP 客户机

要在已经运行 Oracle Solaris 但没有使用 DHCP 的系统上启用 DHCP 客户机，您必须首先取消配置系统。引导系统时，您必须发出某些命令以设置系统并启用 DHCP 客户机。

注 – 在许多部署中，常见做法是使用静态 IP 地址设置基础结构的关键部分，而不是使用 DHCP。确定网络中哪些设备（例如路由器和某些服务器）应该是客户机，哪些设备不应该是客户机，不在本指南讨论范围内。

▼ 如何启用 DHCP 客户机

仅当安装 Oracle Solaris 期间没有启用 DHCPv4 时，才有必要执行此过程。对于 DHCPv6，从来不需要执行此过程。

1. **成为客户机系统上的超级用户。**

2. **如果此系统使用预配置而不是交互式配置，请编辑 sysidcfg 文件。将 dhcp 子项添加到 sysidcfg 文件内的 network_interface 关键字中。**

   例如，network_interface=hme0 {dhcp}。有关更多信息，请参见 sysidcfg(4) 手册页。

3. **取消配置系统并关闭系统。**

   ```
   # sys-unconfig
   ```

   有关使用此命令删除的配置信息的更多信息，请参见 sys-unconfig(1M) 手册页。

4. **完成关闭之后重新引导系统。**

   如果系统使用预配置，则 sysidcfg 文件中的 dhcp 子项将系统配置为在引导系统时使用 DHCP 客户机。

   如果系统不使用预配置，则重新引导系统时，sysidtool 程序会提示您提供系统配置信息。有关更多信息，请参见 sysidtool(1M) 手册页。

5. **当提示使用 DHCP 来配置网络接口时，指定 "Yes"（是）。**

▼ 如何禁用 DHCP 客户机

1. **成为客户机系统上的超级用户。**
2. **如果使用 `sysidcfg` 文件预配置了系统，请从 `network_interface` 关键字中删除 `dhcp` 子项。**
3. **取消配置系统并关闭系统。**

   ```
   # sys-unconfig
   ```

   有关使用此命令删除的配置信息的更多信息，请参见 `sys-unconfig(1M)` 手册页。
4. **完成关闭之后重新引导系统。**

   如果系统使用预配置，则不会提示您提供配置信息，并且也不会配置 DHCP 客户机。

   如果系统不使用预配置，则重新引导系统时，`sysidtool` 程序会提示您提供系统配置信息。有关更多信息，请参见 `sysidtool(1M)` 手册页。
5. **当提示使用 DHCP 来配置网络接口时，指定 "No"（否）。**

DHCP 客户机管理

在系统正常操作的情况下，不需要对 DHCP 客户机软件进行管理。`dhcpagent` 守护进程会在引导系统时自动启动，重新协商租用，并在关闭系统时停止。您不应直接手动启动和停止 `dhcpagent` 守护进程。相反，作为客户机系统上的超级用户，您可以在必要时使用 `ifconfig` 命令来影响 `dhcpagent` 对网络接口的管理。

用于 DHCP 客户机的 `ifconfig` 命令选项

本节汇总了 `ifconfig(1M)` 手册页中介绍的命令选项。这些命令的 DHCPv4 版本与 DHCPv6 版本之间的唯一区别是 "inet6" 关键字。运行 DHCPv6 时使用 "inet6" 关键字，但是在运行 DHCPv4 时忽略它。

使用 `ifconfig`命令可以执行以下操作：

- **启动 DHCP 客户机**－`ifconfig` *interface* `[inet6] dhcp start` 命令可启动 `dhcpagent` 与 DHCP 服务器之间的交互，来获取 IP 地址以及一组新的配置选项。当您更改希望客户机立即使用的信息时（例如添加 IP 地址或更改子网掩码时），此命令非常有用。
- **仅请求网络配置信息**－`ifconfig` *interface* `[inet6] dhcp inform` 命令使 `dhcpagent` 发出对网络配置参数（但不包括 IP 地址）的请求。当网络接口具有静态 IP 地址，但是客户机系统需要更新的网络选项时，此命令非常有用。例如，如果您不使用 DHCP 管理 IP 地址，但是使用它在网络上配置主机，则此命令非常有用。

- **请求租用期延长**－`ifconfig` *interface* `[inet6] dhcp extendipadm refresh-addr` *dhcp-addrobj* 命令使 `dhcpagent` 发出续订租用的请求。客户机会自动发出续订租用的请求。但是，在以下情况下您可能需要使用此命令：您更改了租用时间并希望客户机立即使用新的租用时间，而不是等到下次尝试续订租用时使用。
- **释放 IP 地址**－`ifconfig` *interface* `[inet6] dhcp release` 命令使 `dhcpagent` 放弃由网络接口使用的 IP 地址。当租用过期时，将自动释放 IP 地址。您可能希望通过手提电脑发出此命令，例如，在离开一个网络并计划在新网络上启动系统时。另请参见 `/etc/default/dhcpagent` 配置文件 `RELEASE_ON_SIGTERM` 属性。
- **删除 IP 地址**－`ifconfig` *interface* `[inet6] dhcp drop` 命令使 `dhcpagent` 关闭网络接口而不通知 DHCP 服务器，并在文件系统中高速缓存租用。借助此命令，客户机可以在重新引导时使用相同的 IP 地址。
- **对网络接口执行 Ping 命令**－`ifconfig` *interface* `[inet6] dhcp ping` 命令可用于确定接口是否在 DHCP 控制之下。
- **查看网络接口的 DHCP 配置状态**－`ifconfig` *interface* `[inet6] dhcp status` 命令显示 DHCP 客户机的当前状态。显示内容指示以下各项：
  - IP 地址是否已绑定到客户机
  - 发送、接收和拒绝的请求数
  - 此接口是否为主接口
  - 租用的获取时间、过期时间以及安排开始续订尝试的时间

  例如：

```
# ifconfig hme0 dhcp status
Interface  State         Sent  Recv  Declined  Flags
hme0       BOUND            1     1         0   [PRIMARY]
(Began,Expires,Renew)=(08/16/2005 15:27, 08/18/2005 13:31, 08/17/2005 15:24)

# ifconfig hme0 inet6 dhcp status
Interface  State         Sent  Recv  Declined  Flags
hme0       BOUND            1     0         0   [PRIMARY]
(Began,Expires,Renew)=(11/22/2006 20:39, 11/22/2006 20:41, 11/22/2006 20:40)
```

设置 DHCP 客户机配置参数

客户机系统上的 `/etc/default/dhcpagent` 文件包含 `dhcpagent` 的可调参数。您可以使用文本编辑器来更改多个影响客户机操作的参数。`/etc/default/dhcpagent` 文件记录完好，因此，有关更多信息，请参阅此文件和 dhcpagent(1M) 手册页。

`/etc/dhcp.`*interface* 文件是另一个设置影响 DHCP 客户机的参数的位置。系统启动脚本将在此文件中设置的参数与 `ifconfig` 命令一起使用。但是，这仅影响 DHCPv4。没有 DHCPv6 等效项。

缺省情况下，DHCP 客户机配置如下：

对于 DHCPv4

- 客户机系统不需要特定的主机名。

  如果您希望客户机请求特定的主机名，请参见第 384 页中的 “DHCPv4 客户机主机名”。
- 客户机的缺省请求在 `/etc/default/dhcpagent` 中提供，包括 DNS 服务器、DNS 域和广播地址。

  可以在 `/etc/default/dhcpagent` 文件内的 `PARAM_REQUEST_LIST` 关键字中将 DHCP 客户机的参数文件设置为请求更多选项。可以将 DHCP 服务器配置为提供没有经过专门请求的选项。有关使用 DHCP 服务器宏将信息发送到客户机的信息，请参见 `dhcpd`(8) 手册页和第 343 页中的 “使用 DHCP 宏（任务列表）”。

对于 DHCPv4 和 DHCPv6

- 客户机系统在一个物理网络接口上使用 DHCP。

  如果您希望在多个物理网络接口上使用 DHCP，请参见第 383 页中的 “具有多个网络接口的 DHCP 客户机系统”。
- 如果在安装 Oracle Solaris 之后配置了 DHCP 客户机，则客户机不会自动配置为名称服务客户机。

  有关将名称服务用于 DHCP 客户机的信息，请参见第 385 页中的 “DHCP 客户机系统和名称服务”。

具有多个网络接口的 DHCP 客户机系统

DHCP 客户机可以同时管理一个系统上的多个不同接口。接口可以是物理接口，也可以是逻辑接口。每个接口都有自己的 IP 地址和租用时间。如果为 DHCP 配置多个网络接口，则客户机会分别发出请求以配置这些接口。客户机为每个接口分别保留了一组网络配置参数。虽然参数是分别存储的，但是某些参数仍具有全局性。全局参数应用于整个系统，而不是应用于特定的网络接口。

主机名、NIS 域名和时区是全局参数的示例。对于每个接口，全局参数通常具有不同的值。但是，只有一个值可以用于与每个系统关联的每个全局参数。为确保对每个全局参数的查询只有一个答案，仅使用主网络接口的参数。对于要视为主接口的接口，您可以在 `/etc/dhcp.`*interface* 文件中插入关键字 `primary`。如果没有使用 `primary` 关键字，则会将按字母顺序列出的第一个接口视为主接口。

DHCP 客户机针对逻辑接口和物理接口执行同样的租用管理，但对逻辑接口存在以下限制：

- DHCP 客户机不管理与逻辑接口关联的缺省路由。

  Oracle Solaris 内核将路由与物理接口而不是逻辑接口进行关联。建立物理接口的 IP 地址之后，应该将所需的缺省路由放在路由表中。如果随后使用 DHCP 来配置与此物理接口关联的逻辑接口，则应已存在所需的路由。逻辑接口使用相同的路由。

当物理接口上的租用过期时，DHCP 客户机便会删除与此接口关联的缺省路由。当逻辑接口上的租用过期时，DHCP 客户机不会删除与此逻辑接口关联的缺省路由。关联的物理接口（也可能包括其他逻辑接口）可能需要使用相同的路由。

如果您需要添加或删除与 DHCP 控制的接口关联的缺省路由，则可以使用 DHCP 客户机事件脚本机制。请参见第 389 页中的 “DHCP 客户机事件脚本”。

DHCPv4 客户机主机名

缺省情况下，DHCPv4 客户机不提供自己的主机名，因为此客户机希望 DHCP 服务器提供主机名。缺省情况下，DHCPv4 服务器配置为向 DHCPv4 客户机提供主机名。当您同时使用 DHCPv4 客户机和服务器时，这些缺省设置会正常发挥作用。但是，当您将 DHCPv4 客户机与某些第三方 DHCP 服务器一起使用时，客户机可能不会从服务器接收主机名。如果 DHCP 客户机不通过 DHCP 接收主机名，客户机系统会在 `/etc/nodename` 文件中检查名称来用作主机名。如果此文件为空，则主机名将设置为 `unknown`。

如果 DHCP 服务器在 DHCP `Hostname`（主机名）选项中提供了名称，客户机将使用该主机名，即使 `/etc/nodename` 文件中放置了其他值也是如此。如果您希望客户机使用特定的主机名，则可以使客户机请求此名称。请参见以下过程。

注 – 以下过程并不适用于所有 DHCP 服务器。虽然在此过程中您要求客户机将特定的主机名发送到 DHCP 服务器并期望返回相同名称，

但是 DHCP 服务器不必考虑此请求，而且许多 DHCP 服务器都不考虑此请求。它们只是返回不同的名称。

▼ 如何使 DHCPv4 客户机请求特定的主机名

1. **在客户机系统上，以超级用户的身份编辑 `/etc/default/dhcpagent` 文件。**

2. **在 `/etc/default/dhcpagent` 文件中查找 `REQUEST_HOSTNAME` 关键字，并按如下方式修改此关键字：**

   ```
   REQUEST_HOSTNAME=yes
   ```

   如果 `REQUEST_HOSTNAME` 前面存在注释符号 (`#`)，请删除 `#`。如果 `REQUEST_HOSTNAME` 关键字不存在，请插入此关键字。

3. **在客户机系统上编辑 `/etc/hostname.`*interface* 文件以添加以下行：**

   `inet` *hostname*

   hostname 是您希望客户机使用的名称。

4 **键入以下命令使客户机在重新引导时执行完整的 DHCP 协商：**

```
# ifconfig interface dhcp release
# reboot
```

将删除在客户机上高速缓存的 DHCP 数据。客户机重新启动协议来请求新的配置信息，其中包含新的主机名。DHCP 服务器首先确保网络上的其他系统没有使用此主机名，然后将此主机名指定给客户机。如果配置为可以执行此操作，则 DHCP 服务器便可使用客户机的主机名来更新名称服务。

如果您随后要更改此主机名，请重复步骤 3 和步骤 4。

DHCP 客户机系统和名称服务

Oracle Solaris 系统支持以下名称服务：DNS、NIS、NIS+ 和本地文件存储 (`/etc/inet/hosts`)。每个名称服务都只有在进行某些配置后才可用。还必须相应地设置名称服务转换器配置文件（请参见 `nsswitch.conf(4)`），以指明要使用的名称服务。

在 DHCP 客户机系统可以使用名称服务之前，您必须将系统配置为此名称服务的客户机。缺省情况下，除非在系统安装过程中另行配置，否则仅使用本地文件。

下表概述了与每个名称服务以及 DHCP 相关的问题。该表中还提供了一些文档交叉引用，可帮助您针对每个名称服务设置客户机。

表 16–1 DHCP 客户机系统的名称服务客户机设置信息

名称服务	客户机设置信息

NIS	如果您使用 DHCP 将 Oracle Solaris 网络安装信息发送到客户机系统，则可以使用包含 `NISservs` 和 `NISdmain` 选项的配置宏。这些选项将 NIS 服务器的 IP 地址和 NIS 域名传递到客户机。然后客户机便会自动成为 NIS 客户机。
	
	如果 DHCP 客户机系统已在运行 Oracle Solaris，则当 DHCP 服务器将 NIS 信息发送到 DHCP 客户机时，不会在该系统上自动配置 NIS 客户机。
	
	如果 DHCP 服务器配置为将 NIS 信息发送到 DHCP 客户机系统，则在客户机上按如下方式使用 `dhcpinfo` 命令时，您可以看到为客户机提供的值：
	
	`# /usr/sbin/dhcpinfo NISdmain`
	
	`# /usr/sbin/dhcpinfo NISservs`
	
	注 – 对于 DHCPv6，在命令中包括 `-v6` 和不同的协议关键字，如下所示：
	
	`# /usr/sbin/dhcpinfo -v6 NISDomain`
	
	`# /usr/sbin/dhcpinfo -v6 NISServers`
	
	将系统设置为 NIS 客户机时，请使用针对 NIS 域名和 NIS 服务器返回的值。
	
	您可以使用标准方法为 DHCP 客户机系统设置 NIS 客户机，如《System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)》中的第 5 章 “Setting Up and Configuring NIS Service”所述。
	
	提示 – 您可以编写一个使用 `dhcpinfo` 和 `ypinit` 的脚本，以便在 DHCP 客户机系统上自动配置 NIS 客户机。

NIS+	如果 DHCP 客户机系统的 NIS+ 客户机是使用常规方法设置的，则 DHCP 服务器在不同的时间可能会为客户机提供不同的地址。这就产生了安全性问题，因为 NIS+ 安全性包括作为配置组成部分的 IP 地址。为确保客户机每次都具有相同的地址，请使用非标准方法设置 DHCP 客户机系统的 NIS+ 客户机，此方法在第 387 页中的 “将 DHCP 客户机设置为 NIS+ 客户机”中介绍。 如果已经手动为 DHCP 客户机系统指定了 IP 地址，则客户机的地址始终不变。您可以使用标准方法设置 NIS+ 客户机，此方法在《System Administration Guide: Naming and Directory Services (NIS+)》中的“Setting Up NIS+ Client Machines”中介绍。
`/etc/inet/hosts`	对于要使用 `/etc/inet/hosts` 作为其名称服务的 DHCP 客户机系统，您必须设置 `/etc/inet/hosts` 文件。 DHCP 客户机系统的主机名将由 DHCP 工具添加到其 `/etc/inet/hosts` 文件中。但是，您必须将此主机名手动添加到网络中其他系统的 `/etc/inet/hosts` 文件中。如果 DHCP 服务器系统使用 `/etc/inet/hosts` 进行名称解析，则您还必须在系统上手动添加客户机的主机名。

表 16–1 DHCP 客户机系统的名称服务客户机设置信息 （续）

名称服务	客户机设置信息
DNS	如果 DHCP 客户机系统通过 DHCP 接收 DNS 域名，则会自动配置客户机系统的 `/etc/resolv.conf` 文件。`/etc/nsswitch.conf` 文件也自动更新，以便将 `dns` 按搜索顺序附加到 `hosts` 行中其他名称服务的后面。有关 DNS 的更多信息，请参见《系统管理指南：名称和目录服务（DNS、NIS 和 LDAP）》。

将 DHCP 客户机设置为 NIS+ 客户机

您可以在作为 DHCP 客户机的 Oracle Solaris 系统上使用 NIS+ 名称服务。但是，如果 DHCP 服务器可以在不同的时间提供不同的地址，则这将在某种程度上绕过 NIS+ 的安全增强功能之一，即创建数据加密标准 (Data Encryption Standard, DES) 证书。出于安全考虑，请将 DHCP 服务器配置为始终提供相同地址。当您设置不使用 DHCP 的 NIS+ 客户机时，请将此客户机的专有 DES 凭证添加到 NIS+ 服务器中。可以使用多种方法来创建凭证，例如使用 `nisclient` 脚本或 `nisaddcred` 命令。

生成 NIS+ 凭证时，要求客户机具有静态主机名以创建和存储这些凭证。如果您要使用 NIS+ 和 DHCP，则必须为 DHCP 客户机的所有主机名创建完全相同的凭证。这样，无论 DHCP 客户机接收何种 IP 地址和关联主机名，此客户机都可以使用相同的 DES 凭证。

以下过程显示如何为所有 DHCP 主机名创建相同的凭证。仅当您知道 DHCP 客户机使用的主机名时，此过程才有效。例如，当 DHCP 服务器生成主机名时，您便会知道客户机可能收到的主机名。

▼ 如何将 DHCP 客户机设置为 NIS+ 客户机

要使 DHCP 客户机系统成为 NIS+ 客户机，该系统必须使用属于 NIS+ 域中其他 NIS+ 客户机系统的凭证。此过程仅为系统生成凭证，而这些凭证仅适用于登录到系统的超级用户。其他登录到 DHCP 客户机系统的用户必须在 NIS+ 服务器中具有自己专有的凭证。这些凭证根据《System Administration Guide: Naming and Directory Services (NIS+)》中的过程进行创建。

1 通过在 NIS+ 服务器上键入以下命令来为客户机创建凭证：

```
# nisgrep nisplus-client-name cred.org_dir > /tmp/file
```

此命令将 NIS+ 客户机的 `cred.org_dir` 表项写入临时文件中。

2 使用 `cat` 命令查看此临时文件的内容。

或者，使用文本编辑器。

3 复制要用于 DHCP 客户机的凭证。

您必须复制公钥和私钥，它们是包含以冒号分隔的数字和字母的长字符串。这些凭证将粘贴到下一步发出的命令中。

4 **通过键入以下命令来为 DHCP 客户机添加凭证：**

```
# nistbladm -a cname=" dhcp-client-name@nisplus-domain" auth_type=DES \
auth_name="unix.dhcp-client-name@nisplus-domain" \
public_data=copied-public-key \
private_data=copied-private-key
```

对于 *copied-public-key*，请粘贴您从临时文件中复制的公钥信息。对于 *copied-private-key*，请粘贴您从临时文件中复制的私钥信息。

5 **通过在 DHCP 客户机系统上键入以下命令来将文件从 NIS+ 客户机系统远程复制到 DHCP 客户机系统：**

```
# rcp nisplus-client-name:/var/nis/NIS_COLD_START /var/nis
# rcp nisplus-client-name:/etc/.rootkey /etc
# rcp nisplus-client-name:/etc/defaultdomain /etc
```

如果您收到 "permission denied"（权限被拒绝）消息，则说明系统可能没有设置为允许远程复制。在这种情况下，您可以以一般用户的身份将文件复制到中间位置。然后，以超级用户的身份，将这些文件从中间位置复制到 DHCP 客户机系统上的适当位置。

6 **通过在 DHCP 客户机系统上键入以下命令来为 NIS+ 复制正确的名称服务转换器文件：**

```
# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
```

7 **重新引导 DHCP 客户机系统。**

现在 DHCP 客户机系统应该可以使用 NIS+ 服务。

示例 16–1 将 DHCP 客户机系统设置为 NIS+ 客户机

以下示例假设您有一个系统 `nisei`，此系统是 NIS+ 域 `dev.example.net` 中的 NIS+ 客户机。您还有一个 DHCP 客户机系统 `dhow`，并且希望将 `dhow` 作为 NIS+ 客户机。

```
    (First log in as superuser on the NIS+ server)
# nisgrep nisei cred.org_dir > /tmp/nisei-cred
# cat /tmp/nisei-cred
nisei.dev.example.net.:DES:unix.nisei@dev.example.net:46199279911a84045b8e0
c76822179138173a20edbd8eab4:90f2e2bb6ffe7e3547346dda624ec4c7f0fe1d5f37e21cff63830
c05bc1c724b
# nistbladm -a cname="dhow@dev.example.net." \
auth_type=DES auth_name="unix.dhow@dev.example.net" \
public_data=46199279911a84045b8e0c76822179138173a20edbd8eab4 \
private_data=90f2e2bb6ffe7e3547346dda624ec4c7f0fe1d5f37e21cff63830\
c05bc1c724b
# rlogin dhow
    (Log in as superuser on dhow)
# rcp nisei:/var/nis/NIS_COLD_START /var/nis
# rcp nisei:/etc/.rootkey /etc
# rcp nisei:/etc/defaultdomain /etc
```

```
# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
# reboot
```

现在 DHCP 客户机系统 dhow 应该可以使用 NIS+ 服务。

示例 16–2 使用脚本添加凭证

如果您要将大量的 DHCP 客户机系统都设置为 NIS+ 客户机，则可以编写脚本。脚本可以快速将项添加到 cred.org_dir NIS+ 表中。以下示例显示了一个脚本样例。

```
#! /usr/bin/ksh
#
# Copyright (c) by Sun Microsystems, Inc. All rights reserved.
#
# Sample script for cloning a credential. Hosts file is already populated
# with entries of the form dhcp-[0-9][0-9][0-9]. The entry we're cloning
# is dhcp-001.
#
#
PUBLIC_DATA=6e72878d8dc095a8b5aea951733d6ea91b4ec59e136bd3b3
PRIVATE_DATA=3a86729b685e2b2320cd7e26d4f1519ee070a60620a93e48a8682c5031058df4
HOST="dhcp-"
DOMAIN="mydomain.example.com"

for
i in 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019
do
     print - ${HOST}${i}
     #nistbladm -r [cname="${HOST}${i}.${DOMAIN}."]cred.org_dir
     nistbladm -a cname="${HOST}${i}.${DOMAIN}." \
         auth_type=DES auth_name="unix.${HOST}${i}@${DOMAIN}" \
         public_data=${PUBLIC_DATA} private_data=${PRIVATE_DTA} cred.org_Dir
done

exit 0
```

DHCP 客户机事件脚本

您可以将 DHCP 客户机设置为运行可执行程序或脚本，这些程序或脚本可执行任何适用于客户机系统的操作。出现特定的 DHCP 租用事件之后，便会自动执行称为**事件脚本**的程序或脚本。可以使用事件脚本来运行其他命令、程序或者脚本以响应特定的租用事件。您必须提供自己的事件脚本以使用此功能。

dhcpagent 使用以下事件关键字来标识 DHCP 租用事件：

事件关键字	说明
BOUND 和 BOUND6	将接口配置为用于 DHCP。客户机从 DHCP 服务器接收确认消息 (DHCPv4 ACK) 或 (DHCPv6 Reply)，此消息同意租用 IP 地址的请求。成功配置接口之后，便会立即调用事件脚本。

EXTEND 和 EXTEND6	客户机成功地延长了租用期。当客户机从 DHCP 服务器接收续订请求的确认消息之后，便会立即调用事件脚本。
EXPIRE 和 EXPIRE6	租用时间结束时租用即到期。对于 DHCPv4，将在从接口中删除租用地址，并将接口标记为关闭之前的瞬间调用事件脚本。对于 DHCPv6，将在从接口中删除最后剩余的租用地址之前调用事件脚本。
DROP 和 DROP6	客户机结束租用，将接口从 DHCP 控制中删除。系统会在将接口从 DHCP 控制中删除之前的瞬间调用事件脚本。
RELEASE 和 RELEASE6	客户机放弃 IP 地址。在客户机释放接口上的地址并将 DHCPv4 RELEASE 或 DHCPv6 Release 包发送到 DHCP 服务器之前的瞬间调用事件脚本。
INFORM 和 INFORM6	接口通过 DHCPv4 INFORM 或 DHCPv6 Information-Request 消息从 DHCP 服务器获取新的或更新的配置信息。当 DHCP 客户机从服务器仅获取配置参数而不获取 IP 地址租用时，将出现这些事件。
LOSS6	在租用失效期间，如果仍然存在一个或多个有效租用，则在删除失效地址之前将调用该事件脚本。被删除的那些地址标记有 IFF_DEPRECATED 标志。

如果出现以上任意一种事件，dhcpagent 都会调用以下命令：

```
/etc/dhcp/eventhook interface event
```

其中，*interface* 是使用 DHCP 的接口，而 *event* 是前面所述的事件关键字之一。例如，当 ce0 接口首次配置为用于 DHCP 时，dhcpagent 会按如下方式调用事件脚本：

```
/etc/dhcp/eventhook net0 BOUND
```

要使用事件脚本功能，您必须执行以下操作：

- 将可执行文件命名为 /etc/dhcp/eventhook。
- 将文件的属主设置为 root。
- 将权限设置为 755 (rwxr-xr-x)。
- 编写脚本或程序来执行一系列操作以响应记录的事件。由于 Sun 可能会添加新事件，因此，程序必须自动忽略所有无法识别或不需要执行操作的事件。例如，如果是 RELEASE 事件，则程序或脚本就会向日志文件中写入消息，如果是其他事件，则会忽略。
- 使脚本或程序不进行交互。调用事件脚本之前，stdin、stdout 和 stderr 将连接到 /dev/null。要查看输出或错误，您必须重定向到文件。

事件脚本从 dhcpagent 继承其程序环境，并以 root 特权运行。如有必要，脚本可以使用 dhcpinfo 实用程序来获取有关接口的更多信息。有关更多信息，请参见 dhcpinfo(1) 手册页。

dhcpagent 守护进程将等待事件脚本在所有事件上退出。如果事件脚本在 55 秒内没有退出，则 dhcpagent 会向脚本进程发送 SIGTERM 信号。如果又经过 3 秒之后进程仍没有退出，则守护进程会发送 SIGKILL 信号以中止此进程。

dhcpagent(1M) 手册页包含一个事件脚本的示例。

示例 16–3 显示了如何使用 DHCP 事件脚本使 /etc/resolv.conf 文件的内容保持最新。当出现 BOUND 和 EXTEND 事件时，脚本便会替换域服务器和名称服务器的名称。当出现 EXPIRE、DROP 和 RELEASE 事件时，该脚本便会从文件中删除域服务器和名称服务器的名称。

注 – 此脚本示例假设 DHCP 是域服务器名称和名称服务器名称的授权源。此脚本还假设 DHCP 控制之下的所有接口都返回相同且最新的信息。以上假设也许并不能反映您系统的具体情况。

示例 16–3 用于更新 /etc/resolv.conf 文件的事件脚本

```
#!/bin/ksh -p

PATH=/bin:/sbin export PATH
umask 0222

# Refresh the domain and name servers on /etc/resolv.conf

insert ()
{
    dnsservers=‘dhcpinfo -i $1 DNSserv‘
    if [ -n "$dnsservers" ]; then
        # remove the old domain and name servers
        if [ -f /etc/resolv.conf ]; then
            rm -f /tmp/resolv.conf.$$
            sed -e ’/^domain/d’ -e ’/^nameserver/d’ \
                /etc/resolv.conf > /tmp/resolv.conf.$$
        fi

        # add the new domain
        dnsdomain=‘dhcpinfo -i $1 DNSdmain‘
        if [ -n "$dnsdomain" ]; then
            echo "domain $dnsdomain" >> /tmp/resolv.conf.$$
        fi

        # add new name servers
        for name in $dnsservers; do
            echo nameserver $name >> /tmp/resolv.conf.$$
        done
        mv -f /tmp/resolv.conf.$$ /etc/resolv.conf
    fi
}
```

示例 16–3 用于更新 /etc/resolv.conf 文件的事件脚本 （续）

```
# Remove the domain and name servers from /etc/resolv.conf

remove ()
{
    if [ -f /etc/resolv.conf ]; then
        rm -f /tmp/resolv.conf.$$
        sed -e '/^domain/d' -e '/^nameserver/d' \
            /etc/resolv.conf > /tmp/resolv.conf.$$
        mv -f /tmp/resolv.conf.$$ /etc/resolv.conf
    fi
}

case $2 in
BOUND | EXTEND)
    insert $1
    exit 0
    ;;
EXPIRE | DROP | RELEASE)
    remove
    exit 0
    ;;
*)
    exit 0
    ;;
esac
```

◆ ◆ ◆ 第 17 章

对 DHCP 问题进行故障排除（参考）

本章提供的信息可帮助您解决在配置 DHCP 服务器或客户机时可能遇到的问题，还可帮助您解决配置完成后使用 DHCP 时可能遇到的问题。

本章包含以下信息：

有关配置 DHCP 服务器的信息，请参见第 14 章，配置 DHCP 服务（任务）。有关配置 DHCP 客户机的信息，请参见第 380 页中的 “启用和禁用 DHCP 客户机”。

对 DHCP 服务器问题进行故障排除

在配置服务器时，您可能会遇到以下几类问题：

NIS+ 问题和 DHCP 数据存储

如果使用 NIS+ 作为 DHCP 数据存储，则可能遇到以下几类问题：

无法选择 NIS+ 作为 DHCP 数据存储

如果您尝试使用 NIS+ 作为数据存储，则 DHCP 管理程序可能不会提供 NIS+ 作为数据存储的选择。如果使用 `dhcpconfig` 命令，则可能会看到一条消息，指出 NIS+ 似乎未安

装，它当前未运行。这两种症状表示尽管网络可能正在使用 NIS+，但尚未针对此服务器配置 NIS+。必须先将服务器系统配置为 NIS+ 客户机，然后才能选择 NIS+ 作为数据存储。

在将 DHCP 服务器系统设置为 NIS+ 客户机之前，以下语句必须成立：

- 域必须已经配置。
- NIS+ 域的主服务器必须正在运行。
- 主服务器的表必须已填充。
- 主机表必须包含新客户机系统（即 DHCP 服务器系统）的项。

有关配置 NIS+ 客户机的详细信息，请参见《System Administration Guide: Naming and Directory Services (NIS+)》中的"Setting Up NIS+ Client Machines"。

未完全配置用于 DHCP 数据存储的 NIS+

将 NIS+ 成功用于 DHCP 之后，如果对 NIS+ 进行更改，则可能会遇到错误。这些更改可能会引起配置问题。请使用以下问题说明和解决方法来帮助确定配置问题的原因。

疑难问题: NIS+ 域中不存在根对象。

解决方法: 键入以下命令：

```
/usr/lib/nis/nisstat
```

此命令会显示域的统计信息。如果不存在根对象，则不会返回任何统计信息。

使用《System Administration Guide: Naming and Directory Services (NIS+)》设置 NIS+ 域。

疑难问题: NIS+ 无法用于 `passwd` 和 `publickey` 信息。

解决方法: 键入以下命令来查看名称服务转换器的配置文件：

```
cat /etc/nsswitch.conf
```

在 `passwd` 和 `publickey` 项中检查 "nisplus" 关键字。有关配置名称服务转换器的信息，请参阅《System Administration Guide: Naming and Directory Services (NIS+)》。

疑难问题: 域名为空。

解决方法: 键入以下命令：

```
domainname
```

如果此命令列出一个空字符串，则表明域尚未设置域名。请使用本地文件进行数据存储或者为网络设置 NIS+ 域。请参阅《System Administration Guide: Naming and Directory Services (NIS+)》。

疑难问题: `NIS_COLD_START` 文件不存在。

解决方法: 在服务器系统上键入以下命令来确定此文件是否存在：

```
cat /var/nis/NIS_COLD_START
```

使用本地文件进行数据存储或者创建 NIS+ 客户机。请参阅《System Administration Guide: Naming and Directory Services (NIS+)》。

有关 DHCP 数据存储的 NIS+ 访问问题

NIS+ 访问问题可能会导致有关 DES 凭证不正确或者更新 NIS+ 对象或表的权限不足的错误消息。请使用以下问题说明和解决方案来确定所遇到的 NIS+ 访问错误的原因。

疑难问题: DHCP 服务器系统没有创建对 NIS+ 域中 `org_dir` 对象的访问权限。

解决方法: 键入以下命令：

```
nisls -ld org_dir
```

访问权限以 `r---rmcdrmcdr---` 形式列出，其中权限分别应用于无人 (nobody)、所有者 (owner)、组 (group) 和全局 (world)。对象的所有者将在随后列出。

通常，`org_dir` 目录对象为所有者和组提供了完全权限。完全权限包括读取、修改、创建和销毁。`org_dir` 目录对象仅会为全局和无人这两个类提供读取访问权限。

DHCP 服务器名必须作为 `org_dir` 对象的所有者列出，或者作为组的主体列出。组必须具有创建访问权限。请使用以下命令列出组：

```
nisls -ldg org_dir
```

如有必要，可使用 `nischmod` 命令更改 `org_dir` 的权限。例如，要向组添加创建权限，可键入以下命令：

```
nischmod g+c org_dir
```

有关更多信息，请参见 `nischmod(1)` 手册页。

疑难问题: DHCP 服务器没有在 `org_dir` 对象下创建表的权限。

通常，此问题表示服务器系统的主体名不是 `org_dir` 对象的所属组的成员，或者不存在所属组。

解决方法: 键入以下命令查找所属组的名称：

```
niscat -o org_dir
```

查找显示如下内容的行：

```
Group : "admin.example.com."
```

使用以下命令列出组中的主体名：

```
nisgrpadm -l groupname
```

例如，以下命令可列出组 `admin.example.com` 的主体名：

```
nisgrpadm -l admin.example.com
```

服务器系统名应作为组的显式成员列出，或者作为隐式成员包括在组中。如有必要，可使用 nisgrpadm 命令将服务器系统的名称添加到组中。

例如，要将服务器名称 pacific 添加到组 admin.example.com 中，可键入以下命令：

```
nisgrpadm -a admin.example.com pacific.example.com
```

有关更多信息，请参见 nisgrpadm(1) 手册页。

疑难问题: DHCP 服务器在 NIS+ cred 表中没有有效的数据加密标准 (Data Encryption Standard, DES) 凭证。

解决方法: 如果存在凭证问题，则会显示错误消息，表明用户在 NIS+ 名称服务中没有 DES 凭证。

使用 nisaddcred 命令可为 DHCP 服务器系统添加安全凭证。

以下示例说明如何在域 example.com 中为系统 mercury 添加 DES 凭证：

```
nisaddcred -p unix.mercury@example.com \
-P mercury.example.com. DES example.com.
```

此命令会提示输入 root 用户口令，生成加密密钥时需要使用此口令。

有关更多信息，请参见 nisaddcred(1M) 手册页。

DHCP 中的 IP 地址分配错误

当客户机尝试获取或检验 IP 地址时，可能会出现一些问题，这些问题记录到 syslog 中或以服务器调试模式输出。以下列出的常见错误消息后指明了可能的原因和解决方案。

`There is no` *n.n.n.n* `dhcp-network table for DHCP client's network`

原因: 客户机正在请求特定的 IP 地址或请求延长其当前 IP 地址的租用期。DHCP 服务器无法找到此地址的 DHCP 网络表。

解决方法: DHCP 网络表可能已被错误地删除。您可以通过使用 DHCP 管理程序或 dhcpconfig 命令再次添加网络来重新创建网络表。

`ICMP ECHO reply to OFFER candidate:` *n.n.n.n*`, disabling`

原因: 要提供给 DHCP 客户机的 IP 地址已处于使用状态。如果多台 DHCP 服务器拥有此地址，则可能会出现此问题。如果已为非 DHCP 网络客户机手动配置了地址，也可能会出现此问题。

解决方法: 确定正确的地址所有权。更正 DHCP 服务器数据库或主机的网络配置。

`ICMP ECHO reply to OFFER candidate:` *n.n.n.n*`.No corresponding dhcp network record.`

原因: 要提供给 DHCP 客户机的 IP 地址在网络表中没有记录。此错误表示选择了 IP 地址之后，便从 DHCP 网络表中删除了此地址记录。此错误仅会在完成重复地址检查之前的短时间内出现。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令查看 DHCP 网络表。如果 IP 地址缺失，请从 DHCP 管理程序的 "Address"（地址）选项卡上的 "Edit"（编辑）菜单中选择 "Create"（创建）来创建地址。您还可以使用 `pntadm` 创建 IP 地址。

`DHCP network record for` *n.n.n.n* `is unavailable, ignoring request.`

原因: 所请求的 IP 地址的记录不在 DHCP 网络表中，因此服务器将删除此请求。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令查看 DHCP 网络表。如果 IP 地址缺失，请从 DHCP 管理程序的 "Address"（地址）选项卡上的 "Edit"（编辑）菜单中选择 "Create"（创建）来创建地址。您还可以使用 `pntadm` 创建地址。

n.n.n.n `currently marked as unusable.`

原因: 无法提供所请求的 IP 地址，因为此地址已在网络表中标记为不可用。

解决方法: 您可以使用 DHCP 管理程序或 `pntadm` 命令来使此地址可用。

n.n.n.n `was manually allocated. No dynamic address will be allocated.`

原因: 已经为客户机 ID 指定了一个手动分配的地址，并且此地址标记为不可用。服务器无法为此客户机分配其他地址。

解决方法: 您可以使用 DHCP 管理程序或 `pntadm` 命令使此地址可用，或者为此客户机手动分配其他地址。

`Manual allocation (`*n.n.n.n, client ID*`) has` *n* `other records. Should have 0.`

原因: 已经为具有指定客户机 ID 的客户机手动指定了多个 IP 地址。一台客户机只能指定一个地址。服务器将选择网络表中最近手动指定的地址。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令修改 IP 地址，从而删除其他手动分配的地址。

`No more IP addresses on` *n.n.n.n* `network.`

原因: 在指定网络中，当前由 DHCP 管理的所有 IP 地址均已分配。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令为该网络创建新的 IP 地址。

`Client:` *clientid* `lease on` *n.n.n.n* `expired.`

原因: 租用期不可协商并已超时。

解决方法: 客户机应该自动重新启动协议以获取新的租用期。

`Offer expired for client:` *n.n.n.n*

原因: 服务器为客户机提供 IP 地址，但客户机响应时间太长而导致所提供的地址过期。

解决方法: 客户机应该再次自动发出一条搜索消息。如果此消息也超时，请增加 DHCP 服务器对所提供的地址进行高速缓存的超时时间。在 DHCP 管理程序中，从 "Service"（服务）菜单中选择 "Modify"（修改）。

`Client: `*clientid*` REQUEST is missing requested IP option.`

原因: 客户机的请求未指定所提供的 IP 地址，所以 DHCP 服务器忽略了该请求。如果使用的第三方 DHCP 客户机不符合更新的 DHCP 协议 RFC 2131，则可能会出现此问题。

解决方法: 更新客户机软件。

`Client: `*clientid*` is trying to renew `*n.n.n.n*`, an IP address it has not leased.`

原因: 此客户机在 DHCP 网络表中的 IP 地址与该客户机在其更新请求中指定的 IP 地址不匹配。DHCP 服务器未更新租期。如果在客户机仍在使用该 IP 地址时删除该客户机的记录，则可能会出现此问题。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令检查网络表，并在必要时更正客户机的记录。应将客户机 ID 绑定到指定的 IP 地址。如果客户机 ID 未绑定，请编辑地址属性以添加客户机 ID。

`Client: `*clientid*` is trying to verify unrecorded address: `*n.n.n.n*`, ignored.`

原因: 指定的客户机尚未在 DHCP 网络表中注册此地址，所以此 DHCP 服务器忽略了该请求。

网络中的其他 DHCP 服务器可能已为此客户机指定了该地址。但是，也可能是由于在客户机仍在使用 IP 地址时删除了该客户机的记录。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令检查此服务器以及网络中任何其他 DHCP 服务器上的网络表。如有必要，请进行更正。

您也可以不执行任何操作并使租用过期。客户机会自动请求新的地址租用期。

如果希望客户机立即获取新的租用期，请键入以下命令，在客户机上重新启动 DHCP 协议：

```
ifconfig interface dhcp release
ifconfig interface dhcp start
```

对 DHCP 客户机配置问题进行故障排除

您可能会遇到以下几类 DHCP 客户机问题：

- 第 399 页中的 “与 DHCP 服务器通信时出现的问题”
- 第 406 页中的 “DHCP 配置信息不准确时出现的问题”

与 DHCP 服务器通信时出现的问题

本节介绍了将 DHCP 客户机添加到网络时可能会遇到的问题。

启用客户机软件并重新引导系统之后，客户机会尝试访问 DHCP 服务器以获取其网络配置。如果客户机无法访问服务器，则可能会看到如下错误消息：

```
DHCP or BOOTP server not responding
```

在确定问题之前，必须同时从客户机和服务器收集诊断信息。要收集信息，可以执行以下任务：

1. 第 399 页中的 “如何在调试模式下运行 DHCP 客户机”
2. 第 400 页中的 “如何在调试模式下运行 DHCP 服务器”
3. 第 400 页中的 “如何使用 snoop 监视 DHCP 网络通信流量”

您可以单独执行这些操作，也可以同时执行这些操作。

所收集的信息可以帮助确定是客户机、服务器还是中继代理出现了问题。然后，即可寻找解决方案。

▼ 如何在调试模式下运行 DHCP 客户机

如果客户机不是 DHCP 客户机，请参阅相应客户机文档，以获取有关如何在调试模式下运行客户机的信息。

如果您拥有 DHCP 客户机，请使用以下步骤。

1 以超级用户身份登录 DHCP 客户机系统。

2 中止 DHCP 客户机守护进程。

```
# pkill -x dhcpagent
```

3 在调试模式下重新启动守护进程。

```
# /sbin/dhcpagent -d1 -f &
```

`-d` 开关可将 DHCP 客户机置于详细级别为 1 的调试模式。`-f` 开关会导致输出发送到控制台而不是 `syslog`。

4 配置接口以启动 DHCP 协商。

```
# ifconfig interface dhcp start
```

使用客户机的网络接口名称（如 `ge0`）替换 *interface*。

在调试模式下运行时，客户机守护进程会在执行 DHCP 请求时将消息显示到屏幕上。有关客户机调试模式输出的信息，请参见第 401 页中的 “DHCP 客户机在调试模式下的输出”。

▼ 如何在调试模式下运行 DHCP 服务器

1. **以超级用户的身份登录服务器系统。**
2. **临时停止 DHCP 服务器。**

   ```
   # svcadm disable -t svc:/network/dhcp-server
   ```

   也可以使用 DHCP 管理程序或 `dhcpconfig` 停止服务器。
3. **在调试模式下重新启动守护进程。**

   ```
   # /usr/lib/inet/in.dhcpd -d -v
   ```

   您还应该使用在运行守护进程时通常使用的所有 `in.dhcpd` 命令行选项。例如，如果将守护进程作为 BOOTP 中继代理运行，请在 `in.dhcpd -d -v` 命令中包括 `-r` 选项。

   在调试模式下运行时，守护进程会在处理 DHCP 或 BOOTP 请求时将消息显示到屏幕上。有关服务器调试模式输出的信息，请参见第 401 页中的 “DHCP 服务器在调试模式下的输出”。

▼ 如何使用 snoop 监视 DHCP 网络通信流量

1. **以超级用户的身份登录 DHCP 服务器系统。**
2. **启动 `snoop` 以开始跟踪服务器的网络接口间的网络通信流量。**

   ```
   # /usr/sbin/snoop -d interface -o snoop-output-filename udp port 67 or udp port 68
   ```

   例如，可以键入以下命令：

   ```
   #  /usr/sbin/snoop -d hme0 -o /tmp/snoop.output udp port 67 or udp port 68
   ```

   `snoop` 会继续监视接口，直到您获取所需信息之后按 Ctrl-C 组合键停止 `snoop`。
3. **引导客户机系统，或者在客户机系统上重新启动 `dhcpagent`。**

   第 399 页中的 “如何在调试模式下运行 DHCP 客户机”介绍了如何重新启动 `dhcpagent`。
4. **在服务器系统上，使用 `snoop` 显示包含网络包内容的输出文件：**

   ```
   # /usr/sbin/snoop -i snoop-output-filename -x0 -v
   ```

   例如，可以键入以下命令：

   ```
   # /usr/sbin/snoop -i /tmp/snoop.output -x0 -v
   ```

另请参见 有关解释输出的信息，请参见第 404 页中的 “DHCP snoop 输出”。

DHCP 客户机在调试模式下的输出

以下示例显示了 DHCP 客户机在调试模式下发送 DHCP 请求并从 DHCP 服务器接收配置信息时的标准输出。

示例 17–1 DHCP 客户机在调试模式下的标准输出

```
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x27fc8 (DHCP filter)
/sbin/dhcpagent: debug: init_ifs: initted interface hme0
/sbin/dhcpagent: debug: insert_ifs: hme0: sdumax 1500, optmax 1260, hwtype 1, hwlen 6
/sbin/dhcpagent: debug: insert_ifs: inserted interface hme0
/sbin/dhcpagent: debug: register_acknak: registered acknak id 5
/sbin/dhcpagent: debug: unregister_acknak: unregistered acknak id 5
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x26018 (ARP reply filter)
/sbin/dhcpagent: info: setting IP netmask on hme0 to 255.255.192.0
/sbin/dhcpagent: info: setting IP address on hme0 to 10.23.3.233
/sbin/dhcpagent: info: setting broadcast address on hme0 to 10.23.63.255
/sbin/dhcpagent: info: added default router 10.23.0.1 on hme0
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x28054 (blackhole filter)
/sbin/dhcpagent: debug: configure_if: bound ifsp->if_sock_ip_fd
/sbin/dhcpagent: info: hme0 acquired lease, expires Tue Aug 10 16:18:33 2006
/sbin/dhcpagent: info: hme0 begins renewal at Tue Aug 10 15:49:44 2006
/sbin/dhcpagent: info: hme0 begins rebinding at Tue Aug 10 16:11:03 2006
```

如果客户机无法访问 DHCP 服务器，则看到的调试模式输出可能会类似于以下示例所显示的输出。

示例 17–2 指明 DHCP 客户机在调试模式下出现问题的输出

```
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x27fc8 (DHCP filter)
/sbin/dhcpagent: debug: init_ifs: initted interface hme0
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
```

如果看到此消息，则表明客户机请求永远无法到达服务器，或者服务器无法将响应发送到客户机。请按照第 400 页中的 “如何使用 snoop 监视 DHCP 网络通信流量”中所述，在服务器上运行 snoop，以确定来自客户机的包是否已到达服务器。

DHCP 服务器在调试模式下的输出

守护进程启动时，标准的服务器调试模式输出将显示服务器配置信息，后跟有关每个网络接口的信息。守护进程启动后，调试模式输出显示有关请求该守护进程的信息。示例 17–3 显示了刚刚启动的 DHCP 服务器的调试模式输出。对于使用其他未响应的 DHCP 服务器拥有的地址的客户机，服务器将延长其租用期。

示例 17–3 DHCP 服务器在调试模式下的标准输出

```
Daemon Version: 3.1
Maximum relay hops: 4
Transaction logging to console enabled.
```

示例 17–3 DHCP 服务器在调试模式下的标准输出 （续）

```
Run mode is: DHCP Server Mode.
Datastore: nisplus
Path: org_dir.dhcp.test..:dhcp.test..:$
DHCP offer TTL: 10
Ethers compatibility enabled.
BOOTP compatibility enabled.
ICMP validation timeout: 1000 milliseconds, Attempts: 2.
Monitor (0005/hme0) started...
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.21.255.255
Netmask: 255.255.0.0
Address: 10.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
Broadcast: 10.22.255.255
Netmask: 255.255.0.0
Address: 10.22.0.1
Monitor (0007/qfe0) started...
Thread Id: 0007 - Monitoring Interface: qfe0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.23.63.255
Netmask: 255.255.192.0
Address: 10.23.0.1
Read 33 entries from DHCP macro database on Tue Aug 10 15:10:27 2006
Datagram received on network device: qfe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qfe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qfe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qfe0
Client: 0800201DBA3A maps to IP: 10.23.3.233
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
DHCP EXTEND 0934312543 0934316143 10.23.3.233 10.21.0.2
          0800201DBA3A SUNW.Ultra-5_10 0800201DBA3A
```

示例 17–4 显示了作为 BOOTP 中继代理启动的 DHCP 守护进程的调试模式输出。此代理将客户机的请求中继到 DHCP 服务器，并将服务器的响应中继到客户机。

示例 17–4 BOOTP 中继在调试模式下的标准输出

```
Relay destination: 10.21.0.4 (blue-servr2)     network: 10.21.0.0
Daemon Version: 3.1
Maximum relay hops: 4
Transaction logging to console enabled.
Run mode is: Relay Agent Mode.
Monitor (0005/hme0) started...
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.21.255.255
Netmask: 255.255.0.0
```

示例 17–4 BOOTP 中继在调试模式下的标准输出 （续）

```
Address: 10.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
Broadcast: 10.22.255.255
Netmask: 255.255.0.0
Address: 10.22.0.1
Monitor (0007/qfe0) started...
Thread Id: 0007 - Monitoring Interface: qfe0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.23.63.255
Netmask: 255.255.192.0
Address: 10.23.0.1
Relaying request 0800201DBA3A to 10.21.0.4, server port.
BOOTP RELAY-SRVR 0934297685 0000000000 0.0.0.0 10.21.0.4 0800201DBA3A
N/A 0800201DBA3A
Packet received from relay agent: 10.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
BOOTP RELAY-CLNT 0934297688 0000000000 10.23.0.1 10.23.3.233 0800201DBA3A
N/A 0800201DBA3A
Relaying request 0800201DBA3A to 10.21.0.4, server port.
BOOTP RELAY-SRVR 0934297689 0000000000 0.0.0.0 10.21.0.4 0800201DBA3A
N/A 0800201DBA3A
Packet received from relay agent: 10.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
```

如果 DHCP 存在问题，则调试模式输出可能会显示警告或错误消息。请使用以下 DHCP 服务器错误消息列表来寻找解决方案。

`ICMP ECHO reply to OFFER candidate:` *ip_address* `disabling`

原因: 在 DHCP 服务器向客户机提供 IP 地址之前，它会对此地址执行 ping 操作以检验地址是否正在使用。如果客户机回复，则表明此地址正在使用。

解决方法: 确保配置的地址尚未使用。您可以使用 `ping` 命令。有关更多信息，请参见 `ping(1M)` 手册页。

`No more IP addresses on` *network-address* `network.`

原因: 在 DHCP 网络表中没有与客户机网络关联的 IP 地址。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令创建更多的 IP 地址。如果 DHCP 守护进程正在监视多个子网，请确保其他地址是客户机所在子网的地址。有关更多信息，请参见第 333 页中的 “将 IP 地址添加到 DHCP 服务”。

`No more IP addresses for` *network-address* `network when you are running the DHCP daemon in BOOTP compatibility mode.`

原因: BOOTP 没有使用租用时间，因此 DHCP 服务器查找设置了 BOOTP 标志的空闲地址以将其分配给 BOOTP 客户机。

解决方法: 使用 DHCP 管理程序分配 BOOTP 地址。请参见第 327 页中的 “通过 DHCP 服务支持 BOOTP 客户机（任务列表）”。

`Request to access nonexistent per network database:` *database-name* `in datastore:` *datastore.*

原因: 在配置 DHCP 服务器的过程中，未创建子网的 DHCP 网络表。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令来创建 DHCP 网络表和新的 IP 地址。请参见第 320 页中的 “添加 DHCP 网络”。

`There is no` *table-name* `dhcp-network table for DHCP client's network.`

原因: 在配置 DHCP 服务器的过程中，未创建子网的 DHCP 网络表。

解决方法: 使用 DHCP 管理程序或 `pntadm` 命令来创建 DHCP 网络表和新的 IP 地址。请参见第 320 页中的 “添加 DHCP 网络”。

`Client using non_RFC1048 BOOTP cookie.`

原因: 网络中的某个设备正在尝试访问不受支持的 BOOTP 实现。

解决方法: 如果无需配置此设备，则忽略此消息。如果您希望支持此设备，请参见第 327 页中的 “通过 DHCP 服务支持 BOOTP 客户机（任务列表）”获取更多信息。

DHCP snoop 输出

在 `snoop` 输出中，您应会看到在 DHCP 客户机系统和 DHCP 服务器系统之间交换了包。每个包中指明了各系统的 IP 地址，另外，还包括包路径中的所有路由器或中继代理的 IP 地址。如果两个系统未交换包，则客户机系统可能根本无法访问服务器系统。因此，不会出现太严重的问题。

要评估 `snoop` 输出，您必须知道预期的行为。例如，必须知道请求是否应通过 BOOTP 中继代理，还必须知道所涉及的系统的 MAC 地址和 IP 地址，以便可以确定这些值是否为期望的值。如果存在多个网络接口，则还必须知道这些网络接口的地址。

以下示例显示了从 `blue-servr2` 上的 DHCP 服务器发送到 MAC 地址为 `8:0:20:8e:f3:7e` 的客户机的 DHCP 确认消息的标准 `snoop` 输出。在此消息中，服务器为客户机指定了 IP 地址 `192.168.252.6` 和主机名 `white-6`。此消息还包括多个标准网络选项以及几个用于客户机的供应商特定选项。

示例 17–5 单个包的 snoop 样例输出

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 26 arrived at 14:43:19.14
```

示例 17–5 单个包的 snoop 样例输出 （续）

```
ETHER:  Packet size = 540 bytes
ETHER:  Destination = 8:0:20:8e:f3:7e, Sun
ETHER:  Source      = 8:0:20:1e:31:c1, Sun
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 526 bytes
IP:   Identification = 64667
IP:   Flags = 0x4 IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 254 seconds/hops
IP:   Protocol = 17 (UDP)
IP:   Header checksum = 157a
IP:   Source address = 10.21.0.4, blue-servr2
IP:   Destination address = 192.168.252.6, white-6
IP:   No options
IP:   UDP:  ----- UDP Header -----
UDP:
UDP:  Source port = 67
UDP:  Destination port = 68 (BOOTPC)
UDP:  Length = 506
UDP:  Checksum = 5D4C
UDP:
DHCP: ----- Dynamic Host Configuration Protocol -----
DHCP:
DHCP: Hardware address type (htype) =  1 (Ethernet (10Mb))
DHCP: Hardware address length (hlen) = 6 octets
DHCP: Relay agent hops = 0
DHCP: Transaction ID = 0x2e210f17
DHCP: Time since boot = 0 seconds
DHCP: Flags = 0x0000
DHCP: Client address (ciaddr) = 0.0.0.0
DHCP: Your client address (yiaddr) = 192.168.252.6
DHCP: Next server address (siaddr) = 10.21.0.2
DHCP: Relay agent address (giaddr) = 0.0.0.0
DHCP: Client hardware address (chaddr) = 08:00:20:11:E0:1B
DHCP:
DHCP: ----- (Options) field options -----
DHCP:
DHCP: Message type = DHCPACK
DHCP: DHCP Server Identifier = 10.21.0.4
DHCP: Subnet Mask = 255.255.255.0
DHCP: Router at = 192.168.252.1
DHCP: Broadcast Address = 192.168.252.255
DHCP: NISPLUS Domainname = dhcp.test
DHCP: IP Address Lease Time = 3600 seconds
DHCP: UTC Time Offset = -14400 seconds
```

示例 17–5 单个包的 snoop 样例输出 （续）

```
DHCP: RFC868 Time Servers at = 10.21.0.4
DHCP: DNS Domain Name = sem.example.com
DHCP: DNS Servers at = 10.21.0.1
DHCP: Client Hostname = white-6
DHCP: Vendor-specific Options (166 total octets):
DHCP:   (02) 04 octets  0x8194AE1B (unprintable)
DHCP:   (03) 08 octets  "pacific"
DHCP:   (10) 04 octets  0x8194AE1B (unprintable)
DHCP:   (11) 08 octets  "pacific"
DHCP:   (15) 05 octets  "xterm"
DHCP:   (04) 53 octets  "/export/s2/base.s2s/latest/Solaris_8/Tools/Boot"
DHCP:   (12) 32 octets  "/export/s2/base.s2s/latest"
DHCP:   (07) 27 octets  "/platform/sun4u/kernel/unix"
DHCP:   (08) 07 octets  "EST5EDT"
  0: 0800 208e f37e 0800 201e 31c1 0800 4500    .. .ó~.. .1...E.
 16: 020e fc9b 4000 fe11 157a ac15 0004 c0a8    ....@....z......
 32: fc06 0043 0044 01fa 5d4c 0201 0600 2e21    ...C.D..]L.....!
 48: 0f17 0000 0000 0000 0000 c0a8 fc06 ac15    ................
 64: 0002 0000 0000 0800 2011 e01b 0000 0000    ........ .......
 80: 0000 0000 0000 0000 0000 0000 0000 0000    ................
 96: 0000 0000 0000 0000 0000 0000 0000 0000    ................
112: 0000 0000 0000 0000 0000 0000 0000 0000    ................
128: 0000 0000 0000 0000 0000 0000 0000 0000    ................
144: 0000 0000 0000 0000 0000 0000 0000 0000    ................
160: 0000 0000 0000 0000 0000 0000 0000 0000    ................
176: 0000 0000 0000 0000 0000 0000 0000 0000    ................
192: 0000 0000 0000 0000 0000 0000 0000 0000    ................
208: 0000 0000 0000 0000 0000 0000 0000 0000    ................
224: 0000 0000 0000 0000 0000 0000 0000 0000    ................
240: 0000 0000 0000 0000 0000 0000 0000 0000    ................
256: 0000 0000 0000 0000 0000 0000 0000 0000    ................
272: 0000 0000 0000 6382 5363 3501 0536 04ac    ......c.Sc5..6..
288: 1500 0401 04ff ffff 0003 04c0 a8fc 011c    ................
304: 04c0 a8fc ff40 0964 6863 702e 7465 7374    .....@.dhcp.test
320: 3304 0000 0e10 0204 ffff c7c0 0404 ac15    3...............
336: 0004 0f10 736e 742e 6561 7374 2e73 756e    ....sem.example.
352: 2e63 6f6d 0604 ac15 0001 0c07 7768 6974    com.........whit
368: 652d 362b a602 0481 94ae 1b03 0861 746c    e-6+.........pac
384: 616e 7469 630a 0481 94ae 1b0b 0861 746c    ific.........pac
400: 616e 7469 630f 0578 7465 726d 0435 2f65    ific...xterm.5/e
416: 7870 6f72 742f 7332 382f 6261 7365 2e73    xport/sx2/bcvf.s
432: 3238 735f 776f 732f 6c61 7465 7374 2f53    2xs_btf/latest/S
448: 6f6c 6172 6973 5f38 2f54 6f6f 6c73 2f42    olaris_x/Tools/B
464: 6f6f 740c 202f 6578 706f 7274 2f73 3238    oot. /export/s2x
480: 2f62 6173 652e 7332 3873 5f77 6f73 2f6c    /bcvf.s2xs_btf/l
496: 6174 6573 7407 1b2f 706c 6174 666f 726d    atest../platform
512: 2f73 756e 346d 2f6b 6572 6e65 6c2f 756e    /sun4u/kernel/un
528: 6978 0807 4553 5435 4544 54ff              ix..EST5EDT.
```

DHCP 配置信息不准确时出现的问题

如果 DHCP 客户机在其网络配置信息中收到不准确的信息，请查看 DHCP 服务器数据。您必须检查 DHCP 服务器为此客户机处理的宏的选项值。不准确的信息示例可能包括错误的 NIS 域名或路由器 IP 地址。

使用以下一般原则可帮助确定不准确信息的来源：

- 按照第 345 页中的 “如何查看在 DHCP 服务器上定义的宏（DHCP 管理程序）”中所述，查看在服务器上定义的宏。查看第 274 页中的 “宏处理的顺序”中的信息，并确定为此客户机自动处理的宏。
- 查看网络表以确定作为配置宏指定给客户机 IP 地址的宏（如果存在）。有关更多信息，请参见第 329 页中的 “在 DHCP 服务中处理 IP 地址（任务列表）”。
- 记录在多个宏中出现的所有选项。请确保在最后一个处理的宏中设置所需的选项值。
- 编辑相应的一个或多个宏以确保将正确的值传送到客户机。请参见第 346 页中的 “修改 DHCP 宏”。

DHCP 客户机提供的主机名存在的问题

本节介绍了 DHCP 客户机提供要向 DNS 注册的自身主机名时，可能会遇到的各种问题。

DHCP 客户机无法请求主机名

如果客户机不是 DHCP 客户机，请参阅相应客户机文档以确定如何将客户机配置为请求主机名。有关 DHCP 客户机，请参见第 384 页中的 “如何使 DHCPv4 客户机请求特定的主机名”。

DHCP 客户机无法获取所请求的主机名

以下列表介绍了客户机在获取其请求的主机名时可能会遇到的问题以及建议的解决方案。

疑难问题: 客户机接受了无法发布 DNS 更新的 DHCP 服务器所提供的地址。

解决方法: 如果有两台 DHCP 服务器可用于客户机，则这两台服务器都应配置为提供 DNS 更新。有关配置 DHCP 服务器和 DNS 服务器的信息，请参见第 313 页中的 “通过 DHCP 服务器启用动态 DNS 更新”。

要确定是否将 DHCP 服务器配置为提供 DNS 更新，请执行下列操作：

1. 确定客户机的 DHCP 服务器的 IP 地址。在客户机系统上，使用 `snoop` 或其他应用程序来捕获网络包。请参见第 400 页中的 “如何使用 snoop 监视 DHCP 网络通信流量”，并在客户机而不是服务器上执行此过程。在 `snoop` 输出中，查找 DHCP 服务器标识符以获取服务器的 IP 地址。
2. 登录到 DHCP 服务器系统以检验此系统是否配置为进行 DNS 更新。请以超级用户身份键入以下命令：

   ```
   dhcpconfig -P
   ```

如果将 UPDATE_TIMEOUT 列为服务器参数，则会将 DHCP 服务器配置为进行 DNS 更新。

3. 在 DNS 服务器上，查看 /etc/named.conf 文件。在相应域的 zone 部分中查找 allow-update 关键字。如果服务器允许通过 DHCP 服务器进行 DNS 更新，则 DHCP 服务器的 IP 地址将在 allow-update 关键字中列出。

疑难问题: 客户机使用 FQDN 选项来指定主机名。DHCP 当前不支持 FQDN 选项，因为此选项没有正式包含在 DHCP 协议中。

解决方法: 在服务器上，使用 snoop 或其他应用程序来捕获网络包。请参见第 400 页中的 “如何使用 snoop 监视 DHCP 网络通信流量”。在 snoop 输出中，查找来自客户机的包中的 FQDN 选项。

将客户机配置为使用 Hostname 选项指定主机名。Hostname 的选项代码为 12。请参阅客户机文档以获取有关说明。

有关 Oracle Solaris 客户机，请参见第 384 页中的 “如何使 DHCPv4 客户机请求特定的主机名”。

疑难问题: 为客户机提供地址的 DHCP 服务器不知道客户机的 DNS 域。

解决方法: 在 DHCP 服务器上，查找具有有效值的 DNSdmain 选项。在为此客户机处理的宏中，将 DNSdmain 选项设置为正确的 DNS 域名。DNSdmain 通常包含在网络宏中。有关更改宏中的选项值的信息，请参见第 346 页中的 “修改 DHCP 宏”。

疑难问题: 客户机所请求的主机名对应于未由 DHCP 服务器管理的 IP 地址。DHCP 服务器无法对不属于该服务器管理的 IP 地址执行 DNS 更新。

解决方法: 检查 syslog 以获取 DHCP 服务器发出的以下消息之一：

- There is no *n.n.n.n* dhcp-network table for DHCP client’s network.
- DHCP network record for *n.n.n.n* is unavailable, ignoring request.

将客户机配置为请求其他名称。请参见第 384 页中的 “如何使 DHCPv4 客户机请求特定的主机名”。选择映射到由 DHCP 服务器管理的地址的名称。您可以在 DHCP 管理程序的 "Addresses"（地址）选项卡中看到地址映射。或者，选择未映射到任何 IP 地址的地址。

疑难问题: 客户机所请求的主机名对应于当前不可用的 IP 地址。此地址可能正在使用，已租用给其他客户机或者准备提供给其他客户机。

解决方法: 检查 syslog 以获取 DHCP 服务器发出的以下消息： ICMP ECHO reply to OFFER candidate: *n.n.n.n*。

将客户机配置为选择对应于其他 IP 地址的名称。或者，从使用此地址的客户机中回收此地址。

疑难问题: 无法将 DNS 服务器配置为接受来自 DHCP 服务器的更新。

解决方法: 在 DNS 服务器上，检查 `/etc/named.conf` 文件。在 DHCP 服务器域相应的 `zone` 部分内的 `allow-update` 关键字中，查找 DHCP 服务器的 IP 地址。如果不存在此 IP 地址，则无法将 DNS 服务器配置为接受来自 DHCP 服务器的更新。

有关配置 DHCP 服务器的信息，请参见第 314 页中的 “如何针对 DHCP 客户机启用动态 DNS 更新”。

如果 DHCP 服务器具有多个接口，则可能需要将 DNS 服务器配置为接受来自所有 DHCP 服务器地址的更新。在 DNS 服务器上，启用调试以查看更新是否到达 DNS 服务器。如果 DNS 服务器收到更新请求，请检查调试模式输出以确定无法进行更新的原因。有关 DNS 调试模式的信息，请参见 `in.named.1M` 手册页。

疑难问题: DNS 更新可能未在分配的时间内完成。如果 DNS 更新未在已配置的时间限制内完成，则 DHCP 服务器不会将主机名返回给客户机。但是，它会继续尝试完成 DNS 更新。

解决方法: 使用 `nslookup` 命令确定更新是否已成功完成。请参见 `nslookup(1M)` 手册页。

例如，假定 DNS 域为 `hills.example.org`，DNS 服务器的 IP 地址为 `10.76.178.11`。客户机要注册的主机名为 `cathedral`。您可以使用以下命令确定是否已向此 DNS 服务器中注册了 `cathedral`：

```
nslookup cathedral.hills.example.org 10.76.178.11
```

如果更新成功完成，但用时超过了分配的时间，则需要增大超时值。请参见第 314 页中的 “如何针对 DHCP 客户机启用动态 DNS 更新”。在此过程中，应该增加在超时之前等待 DNS 服务器发出响应的秒数。

◆ ◆ ◆ 第 18 章

18 DHCP 命令和文件（参考信息）

本章介绍了 DHCP 命令和 DHCP 文件之间的关系，但并未介绍如何使用这些命令。

本章包含以下信息：

DHCP 命令

下表列出了可用于在网络中管理 DHCP 的命令。

表 18–1 用于 DHCP 的命令

命令	说明
`/usr/lib/inet/dhcpd`	仅 ISC DHCP：ISC DHCP 服务器守护进程。有关更多信息，请参见 `dhcpd`(8) 手册页。
`/usr/lib/inet/dhcrelay`	仅 ISC DHCP：启用一种方法将来自不包含 DHCP 服务器的网络上客户机的 DHCP 和 BOOTP 请求中继到其他网络上的服务器。有关更多信息，请参见 `dhcrelay`(8) 手册页。
`/usr/lib/inet/in.dhcpd`	DHCP 服务器守护进程。该守护进程在启动系统时启动。请勿直接启动服务器守护进程。使用 DHCP 管理器、`svcadm` 命令或 `dhcpconfig` 可启动和停止该守护进程。仅当以调试模式运行服务器来解决问题时，才可以直接调用该守护进程。有关更多信息，请参见 `in.dhcpd`(1M) 手册页。
`/usr/sadm/admin/bin/dhcpmgr`	DHCP 管理器，一种用于配置和管理 DHCP 服务的图形用户界面 (Graphical User Interface, GUI) 工具。推荐将 DHCP 管理程序作为 DHCP 管理工具。有关更多信息，请参见 `dhcpmgr`(1M) 手册页。
`/usr/sbin/dhcpagent`	DHCP 客户机守护进程，用于实现 DHCP 协议的客户端。有关更多信息，请参见 `dhcpagent`(1M) 手册页。

表 18–1 用于 DHCP 的命令 （续）

命令	说明
/usr/sbin/dhcpconfig	用于配置和取消配置 DHCP 服务器及 BOOTP 中继代理。另外，还可用于转换为另一种数据存储格式，以及导入和导出 DHCP 配置数据。有关更多信息，请参见 dhcpconfig(1M) 手册页。
/usr/sbin/dhcpinfo	由 Oracle Solaris客户机系统的系统启动脚本用于从 DHCP 客户机守护进程 dhcpagent中获取信息（如主机名）。您也可以在脚本或命令行中使用 dhcpinfo 来获取指定的参数值。有关更多信息，请参见 dhcpinfo(1) 手册页。
/usr/sbin/dhtadm	用于对 dhcptab 表中的选项和宏进行更改。此命令最适用于在创建的脚本中自动更改 DHCP 信息。使用带有 -P 选项的 dhtadm，并通过 grep 命令传输输出，可以在 dhcptab 表中快速搜索特定的选项值。有关更多信息，请参见 dhtadm(1M) 手册页。
/usr/sbin/ifconfig	在系统引导时用于为网络接口指定 IP 地址或配置网络接口参数，或者同时执行这两种操作。在 DHCP 客户机上，ifconfig 可启动 DHCP 以获取配置网络接口所需的参数（包括 IP 地址）。有关更多信息，请参见 ifconfig(1M) 手册页。
/usr/sbin/omshell	仅 ISC DHCP：提供了一种使用对象管理 API (Object Management API, OMAPI) 来查询和更改 ISC DHCP 服务器的状态的方式。有关更多信息，请参见 omshell(1) 手册页。
/usr/sbin/pntadm	用于更改将客户机 ID 映射到 IP 地址的 DHCP 网络表，还可选择将配置信息与 IP 地址进行关联。有关更多信息，请参见 pntadm(1M) 手册页。
/usr/sbin/snoop	用于捕获和显示网络中传递的包的内容。snoop 用于对 DHCP 服务问题进行故障排除。有关更多信息，请参见 snoop(1M) 手册页。

在脚本中运行 DHCP 命令

dhcpconfig、dhtadm 和 pntadm 命令为便于在脚本中使用进行了优化。特别是，pntadm 命令对于在 DHCP 网络表中创建大量 IP 地址项非常有用。以下样例脚本在批处理模式下使用 pntadm 来创建 IP 地址。

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本

```
#! /usr/bin/ksh
#
# This script utilizes the pntadm batch facility to add client entries
# to a DHCP network table. It assumes that the user has the rights to
# run pntadm to add entries to DHCP network tables.

#
# Based on the switch setting, query the netmasks table for a netmask.
# Accepts one argument, a dotted IP address.
#
get_netmask()
{
    MTMP=‘getent netmasks ${1} | awk ’{ print $2 }’‘
```

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本 （续）

```
    if [ ! -z "${MTMP}" ]
    then
        print - ${MTMP}
    fi
}

#
# Based on the network specification, determine whether or not network is
# subnetted or supernetted.
# Given a dotted IP network number, convert it to the default class
# network.(used to detect subnetting). Requires one argument, the
# network number. (e.g. 10.0.0.0) Echos the default network and default
# mask for success, null if error.
#
get_default_class()
{
    NN01=${1%%.*}
    tmp=${1#*.}
    NN02=${tmp%%.*}
    tmp=${tmp#*.}
    NN03=${tmp%%.*}
    tmp=${tmp#*.}
    NN04=${tmp%%.*}
    RETNET=""
    RETMASK=""

    typeset -i16 ONE=10#${1%%.*}
    typeset -i10 X=$((${ONE}&16#f0))
    if [ ${X} -eq 224 ]
    then
        # Multicast
        typeset -i10 TMP=$((${ONE}&16#f0))
        RETNET="${TMP}.0.0.0"
        RETMASK="240.0.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#80))
    if [ -z "${RETNET}" -a ${X} -eq 0 ]
    then
        # Class A
        RETNET="${NN01}.0.0.0"
        RETMASK="255.0.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#c0))
    if [ -z "${RETNET}" -a ${X} -eq 128 ]
    then
        # Class B
        RETNET="${NN01}.${NN02}.0.0"
        RETMASK="255.255.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#e0))
    if [ -z "${RETNET}" -a ${X} -eq 192 ]
    then
        # Class C
        RETNET="${NN01}.${NN02}.${NN03}.0"
        RETMASK="255.255.255.0"
    fi
```

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本 （续）

```
    print - ${RETNET} ${RETMASK}
    unset NNO1 NNO2 NNO3 NNO4 RETNET RETMASK X ONE
}

#
# Given a dotted form of an IP address, convert it to its hex equivalent.
#
convert_dotted_to_hex()
{
    typeset -i10 one=${1%%.*}
    typeset -i16 one=${one}
    typeset -Z2 one=${one}
    tmp=${1#*.}

    typeset -i10 two=${tmp%%.*}
    typeset -i16 two=${two}
    typeset -Z2 two=${two}
    tmp=${tmp#*.}

    typeset -i10 three=${tmp%%.*}
    typeset -i16 three=${three}
    typeset -Z2 three=${three}
    tmp=${tmp#*.}

    typeset -i10 four=${tmp%%.*}
    typeset -i16 four=${four}
    typeset -Z2 four=${four}

     hex=‘print - ${one}${two}${three}${four} | sed -e ’s/#/0/g’‘
     print - 16#${hex}
     unset one two three four tmp
}

#
# Generate an IP address given the network address, mask, increment.
#
get_addr()
{
    typeset -i16 net=‘convert_dotted_to_hex ${1}‘
    typeset -i16 mask=‘convert_dotted_to_hex ${2}‘
    typeset -i16 incr=10#${3}

    # Maximum legal value - invert the mask, add to net.
    typeset -i16 mhosts=~${mask}
    typeset -i16 maxnet=${net}+${mhosts}

    # Add the incr value.
    let net=${net}+${incr}

    if [ $((${net} < ${maxnet})) -eq 1 ]
    then
        typeset -i16 a=${net}\&16#ff000000
        typeset -i10 a="${a}>>24"

        typeset -i16 b=${net}\&16#ff0000
        typeset -i10 b="${b}>>16"
```

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本 （续）

```
        typeset -i16 c=${net}\&16#ff00
        typeset -i10 c="${c}>>8"

        typeset -i10 d=${net}\&16#ff
        print - "${a}.${b}.${c}.${d}"
    fi
    unset net mask incr mhosts maxnet a b c d
}

# Given a network address and client address, return the index.
client_index()
{
    typeset -i NNO1=${1%%.*}
    tmp=${1#*.}
    typeset -i NNO2=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO3=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO4=${tmp%%.*}

    typeset -i16 NNF1
    let NNF1=${NNO1}
    typeset -i16 NNF2
    let NNF2=${NNO2}
    typeset -i16 NNF3
    let NNF3=${NNO3}
    typeset -i16 NNF4
    let NNF4=${NNO4}
    typeset +i16 NNF1
    typeset +i16 NNF2
    typeset +i16 NNF3
    typeset +i16 NNF4
    NNF1=${NNF1#16\#}
    NNF2=${NNF2#16\#}
    NNF3=${NNF3#16\#}
    NNF4=${NNF4#16\#}
    if [ ${#NNF1} -eq 1 ]
    then
        NNF1="0${NNF1}"
    fi
    if [ ${#NNF2} -eq 1 ]
    then
        NNF2="0${NNF2}"
    fi
    if [ ${#NNF3} -eq 1 ]
    then
        NNF3="0${NNF3}"
    fi
    if [ ${#NNF4} -eq 1 ]
    then
        NNF4="0${NNF4}"
    fi
    typeset -i16 NN
    let NN=16#${NNF1}${NNF2}${NNF3}${NNF4}
    unset NNF1 NNF2 NNF3 NNF4
```

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本 （续）

```
    typeset -i NNO1=${2%%.*}
    tmp=${2#*.}
    typeset -i NNO2=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO3=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO4=${tmp%%.*}
    typeset -i16 NNF1
    let NNF1=${NNO1}
    typeset -i16 NNF2
    let NNF2=${NNO2}
    typeset -i16 NNF3
    let NNF3=${NNO3}
    typeset -i16 NNF4
    let NNF4=${NNO4}
    typeset +i16 NNF1
    typeset +i16 NNF2
    typeset +i16 NNF3
    typeset +i16 NNF4
    NNF1=${NNF1#16\#}
    NNF2=${NNF2#16\#}
    NNF3=${NNF3#16\#}
    NNF4=${NNF4#16\#}
    if [ ${#NNF1} -eq 1 ]
    then
        NNF1="0${NNF1}"
    fi
    if [ ${#NNF2} -eq 1 ]
    then
        NNF2="0${NNF2}"
    fi
    if [ ${#NNF3} -eq 1 ]
    then
        NNF3="0${NNF3}"
    fi
    if [ ${#NNF4} -eq 1 ]
    then
        NNF4="0${NNF4}"
    fi
    typeset -i16 NC
    let NC=16#${NNF1}${NNF2}${NNF3}${NNF4}
    typeset -i10 ANS
    let ANS=${NC}-${NN}
    print - $ANS
}

#
# Check usage.
#
if [ "$#" != 3 ]
then
    print "This script is used to add client entries to a DHCP network"
    print "table by utilizing the pntadm batch facilty.\n"
    print "usage: $0 network start_ip entries\n"
    print "where: network is the IP address of the network"
```

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本 （续）

```
        print "       start_ip is the starting IP address \n"
        print "       entries is the number of the entries to add\n"
    print "example: $0 10.148.174.0 10.148.174.1 254\n"
    return
fi

#
# Use input arguments to set script variables.
#
NETWORK=$1
START_IP=$2
typeset -i STRTNUM=‘client_index ${NETWORK} ${START_IP}‘
let ENDNUM=${STRTNUM}+$3
let ENTRYNUM=${STRTNUM}
BATCHFILE=/tmp/batchfile.$$
MACRO=‘uname -n‘

#
# Check if mask in netmasks table. First try
# for network address as given, in case VLSM
# is in use.
#
NETMASK=‘get_netmask ${NETWORK}‘
if [ -z "${NETMASK}" ]
then
    get_default_class ${NETWORK} | read DEFNET DEFMASK
    # use the default.
    if [ "${DEFNET}" != "${NETWORK}" ]
    then
        # likely subnetted/supernetted.
        print - "\n\n###\tWarning\t###\n"
        print - "Network ${NETWORK} is netmasked, but no entry was found  \n
              in the ’netmasks’ table; please update the ’netmasks’  \n
              table in the appropriate nameservice before continuing. \n
              (See /etc/nsswitch.conf.) \n" >&2
        return 1
    else
        # use the default.
        NETMASK="${DEFMASK}"
    fi
fi

#
# Create a batch file.
#
print -n "Creating batch file "
while [ ${ENTRYNUM} -lt ${ENDNUM} ]
do
    if [ $((${ENTRYNUM}-${STRTNUM}))%50 -eq 0 ]
    then
        print -n "."
    fi

    CLIENTIP=‘get_addr ${NETWORK} ${NETMASK} ${ENTRYNUM}‘
    print "pntadm -A ${CLIENTIP} -m ${MACRO} ${NETWORK}" >> ${BATCHFILE}
    let ENTRYNUM=${ENTRYNUM}+1
```

示例 18–1 使用 pntadm 命令的 addclient.ksh 脚本 （续）

```
done
print " done.\n"

#
# Run pntadm in batch mode and redirect output to a temporary file.
# Progress can be monitored by using the output file.
#
print "Batch processing output redirected to ${BATCHFILE}"
print "Batch processing started."

pntadm -B ${BATCHFILE} -v > /tmp/batch.out 2 >&1

print "Batch processing completed."
```

DHCP 服务使用的文件

下表列出了与 DHCP 关联的文件。

表 18–2 DHCP 守护进程和命令使用的文件和表

文件名或表名	说明
dhcptab	仅旧版 Sun DHCP：DHCP 配置信息表的通称，这些配置信息以选项及指定值的形式进行记录，而这些选项及指定值随后会组合为宏。dhcptab 表的名称及其位置由用于 DHCP 信息的数据存储确定。有关更多信息，请参见 dhcptab(4) 手册页。
DHCP 网络表	仅旧版 Sun DHCP：将 IP 地址映射到客户机 ID 和配置选项。DHCP 网络表根据网络的 IP 地址（如 10.21.32.0）来命名。不存在名为 dhcp_network 的文件。DHCP 网络表的名称和位置由用于 DHCP 信息的数据存储确定。有关更多信息，请参见 dhcp_network(4) 手册页。
/etc/dhcp/eventhook	仅传统 Sun DHCP：dhcpagent 守护进程可以自动运行的脚本或可执行文件。有关更多信息，请参见 dhcpagent(1M) 手册页。
/etc/inet/dhcpd4.conf /etc/inet/dhcpd6.conf	仅 ISC DHCP：包含 ISC DHCP 服务器 dhcpd 的配置信息。有关更多信息，请参见 dhcpd.conf(5) 手册页。
/etc/inet/dhcpsvc.conf	仅旧版 Sun DHCP：存储 DHCP 守护进程的启动选项和数据存储信息。此文件决不能手动编辑。使用 dhcpconfig 命令可更改启动选项。有关更多信息，请参见 dhcpsvc.conf(4) 手册页。
nsswitch.conf	指定名称服务数据库的位置以及在名称服务中搜索各种信息的顺序。配置 DHCP 服务器时，会读取 nsswitch.conf 文件以获取准确的配置信息。此文件位于 /etc 目录中。有关更多信息，请参见 nsswitch.conf(4) 手册页。
resolv.conf	包含用于解析 DNS 查询的信息。在 DHCP 服务器配置过程中，会查看此文件以获取有关 DNS 域和 DNS 服务器的信息。此文件位于 /etc 目录中。有关更多信息，请参见 resolv.conf(4) 手册页。

表 18–2 DHCP 守护进程和命令使用的文件和表 （续）

文件名或表名	说明
dhcp.*interface*	表示将在 dhcp.*interface* 文件名中指定的客户机网络接口上使用 DHCP。例如，存在名为 dhcp.qe0 的文件表示将在 qe0 接口上使用 DHCP。dhcp.*interface* 文件可能包含一些命令，这些命令将作为选项传递给用于在客户机上启动 DHCP 的 ifconfig 命令。此文件位于 DHCP 客户机系统上的 /etc 目录中。没有特定的手册页，请参见 dhcp(5)。
/etc/dhcp/*interface*.dhc /etc/dhcp/*interface*.dh6	包含从 DHCP 中为给定网络接口获取的配置参数。对于 DHCPv4，该文件名以 dhc 结尾。对于 DHCPv6，该文件名以 dh6 结尾。当删除此接口的 IP 地址租用时，客户机会将当前配置信息高速缓存至 /etc/dhcp/*interface*.dhc 中。例如，如果在 qe0 接口上使用 DHCP，则 dhcpagent 会将配置信息高速缓存至 /etc/dhcp/qe0.dhc 中。下次在此接口上启动 DHCP 时，如果租用未过期，客户机会请求使用高速缓存的配置。如果 DHCP 服务器拒绝此请求，则客户机会启动标准的 DHCP 租用协商进程。
/etc/default/dhcpagent	设置 dhcpagent 客户机守护进程的参数值。有关参数的信息，请参见 /etc/default/dhcpagent 文件或 dhcpagent(1M) 手册页。
/etc/dhcp/inittab /etc/dhcp/inittab6	仅旧版 Sun DHCP：定义 DHCP 选项代码的各个方面（如数据类型）以及指定助记标签。有关此文件语法的更多信息，请参见 dhcp_inittab(4) 手册页。/etc/dhcp/inittab6 由 DHCPv6 客户机使用。 在客户机上，dhcpinfo 命令会使用 /etc/dhcp/inittab 文件中的信息来为此信息的读者提供更多有意义的信息。在 DHCP 服务器系统上，DHCP 守护进程和管理工具会使用此文件来获取 DHCP 选项信息。 /etc/dhcp/inittab 文件将替换在先前的发行版中使用的 /etc/dhcp/dhcptags 文件。
/var/db/isc-dhcp/dhcp4.leases /var/db/isc-dhcp/dhcp4.leases- /var/db/isc-dhcp/dhcp6.leases /var/db/isc-dhcp/dhcp6.leases-	仅 ISC DHCP：列出对 DHCPv4 和 DHCPv6 服务器的租用。文件名以 "-" 结尾的文件为先前副本。

DHCP 选项信息

过去，DHCP 选项信息存储在多个位置，包括服务器的 dhcptab 表、客户机的 dhcptags 文件和各种程序的内部表。从 Solaris 8 发行版开始，选项信息已合并在 /etc/dhcp/inittab 文件中。有关此文件的详细信息，请参见 dhcp_inittab(4) 手册页。

DHCP 客户机使用 DHCP inittab 文件代替 dhcptags 文件。客户机使用此文件可获取有关在 DHCP 包中收到的选项代码的信息。DHCP 服务器上的 in.dhcpd、snoop 和 dhcpmgr 程序也将使用 inittab 文件。

确定站点是否受到影响

大多数使用 DHCP 的站点**不会**受到转换为 /etc/dhcp/inittab 文件这一操作的影响。如果满足以下所有条件，则站点将会受到影响：

- 计划从 Solaris 8 发行版之前的 Oracle Solaris 发行版进行升级。
- 之前已创建了新的 DHCP 选项。
- 修改了 /etc/dhcp/dhcptags 文件并希望保留更改。

升级时，升级日志会通知您 dhcptags 文件已经修改，因此应对 DHCP inittab 文件进行更改。

dhcptags 和 inittab 文件之间的差异

inittab 文件比 dhcptags 文件包含更多的信息。inittab 文件使用的语法也不同。

以下是 dhcptags 项的一个样例：

```
33 StaticRt - IPList Static_Routes
```

33 为在 DHCP 包中传送的数字代码。StaticRt 为选项名。IPList 表示 StaticRt 的数据类型必须为 IP 地址列表。Static_Routes 是一个描述性更强的名称。

inittab 文件由说明每个选项的单行记录组成。其格式类似于在 dhcptab 中定义符号的格式。下表说明了 inittab 文件的语法。

选项	说明
option-name	选项名。选项名在其选项类别中必须是唯一的，不可与 "Standard"（标准）、"Site"（站点）和 "Vendor"（供应商）等类别中的其他选项重名。例如，不能有两个同名的 "Site"（站点）选项，不能创建与 "Standard"（标准）选项名称相同的 "Site"（站点）选项。
category	标识选项所属的名称空间。必须为以下名称空间之一：标准、站点、供应商、字段或内部。
code	当选项在网络中发送时标识此选项。大多数情况下，代码唯一标识选项，与类别无关。但是，在涉及内部类别（如字段选项或内部选项）的情况下，代码可能会用于其他目的。代码可能不是全局唯一的。代码在选项类别中应是唯一的，并且不会与标准字段和站点字段中的代码重名。
type	说明与此选项关联的数据。有效类型为 IP、ASCII、Octet、Boolean、Unumber8、Unumber16、Unumber32、Unumber64、Snumber8、Snumber16、Snumber32 和 Snumber64 等。对数字来说，首字母 U 或 S 表示该数字为无符号或带符号的数字。末尾数字指明数字中的二进制位数。例如，Unumber8 即为无符号的 8 位二进制数字。类型不区分大小写。

granularity 说明组成此选项的完整值的数据单元的数量。

maximum 说明此选项允许使用的完整值的数量。0 表示一个无穷大的数字。

consumers 说明可使用此信息的程序。使用者应设置为 sdmi，其中：

```
s    snoop
d    in.dhcpd
m    dhcpmgr
i    dhcpinfo
```

以下是 inittab 项的一个样例：

```
StaticRt - Standard, 33, IP, 2, 0, sdmi
```

该项介绍了名为 StaticRt 的选项。此选项属于标准类别，选项代码为 33。由于类型为 IP、粒度为 2、最大值为无穷大 (0)，因此预期的数据是一个可能为无穷大的 IP 地址对数量。此选项的使用者为 sdmi： snoop、in.dhcpd、dhcpmgr 和 dhcpinfo。

将 dhcptags 项转换为 inittab 项

如果之前向 dhcptags 文件中添加了项，则必须向新的 inittab 文件中添加对应的项，这样才能继续使用添加到站点中的选项。以下示例说明如何以 inittab 格式表示 dhcptags 项的一个样例。

假定已经为连接到网络的传真机添加了以下 dhcptags 项：

```
128 FaxMchn - IP Fax_Machine
```

代码 128 意味着此选项一定属于站点类别。选项名为 FaxMchn，数据类型为 IP。

对应的 inittab 项可能是：

```
FaxMchn SITE, 128, IP, 1, 1, sdmi
```

粒度 1 和最大值 1 表示此选项应有一个 IP 地址。

第 4 部 分

IP 安全性

本部分重点介绍网络安全性。IP 安全体系结构 (IPsec) 在包级别保护网络。Internet 密钥管理 (IKE) 管理 IPsec 的密钥。Oracle Solaris 的 IP 过滤器功能提供防火墙。

◆ ◆ ◆ 第 19 章

19 IP 安全体系结构（概述）

IP 安全体系结构 (IPsec) 为 IPv4 和 IPv6 网络包中的 IP 数据报提供加密保护。

本章包含以下信息：

有关如何在网络中实现 IPsec 的信息，请参见第 20 章，配置 IPsec（任务）。有关参考信息，请参见第 21 章，IP 安全体系结构（参考信息）。

IPsec 中的新增功能

Solaris 10 4/09：从此发行版开始，服务管理工具 (Service Management Facility, SMF) 将 IPsec 作为一组服务来管理。

缺省情况下，在系统引导时，将启用以下两个 IPsec 服务：

- `svc:/network/ipsec/policy:default`
- `svc:/network/ipsec/ipsecalgs:default`

缺省情况下，在系统引导时，将禁用以下密钥管理服务：

- `svc:/network/ipsec/manual-key:default`
- `svc:/network/ipsec/ike:default`

要在 SMF 下激活 IPsec 策略，请执行以下步骤：

1. 将 IPsec 策略项添加到 `ipsecinit.conf` 文件。
2. 配置 Internet 密钥交换 (Internet Key Exchange, IKE) 或手动配置密钥。
3. 刷新 IPsec 策略服务。
4. 启用密钥管理服务。

有关 SMF 的更多信息，请参见《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。另请参见 `smf(5)` 和`svcadm(1M)`手册页。

从此发行版开始，`ipsecconf` 和 `ipseckey` 命令具有 `-c` 选项，以用于检查它们各自配置文件的语法。另外，还提供了网络 IPsec 管理权限配置文件以用于管理 IPsec 和 IKE。

Solaris 10 7/07：从此发行版开始，IPsec 以隧道模式全面实现隧道，并且支持隧道的实用程序有所修改。

- IPsec 在虚拟专用网络 (Virtual Private Network, VPN) 中以隧道模式实现了隧道。在隧道模式下，IPsec 支持位于一个 NAT 后的多个客户机。在隧道模式下，IPsec 可与其他供应商提供的 IP-in-IP 隧道实现方式交互使用。IPsec 继续以传输模式支持隧道，所以它与早期的 Solaris 发行版兼容。
- 创建隧道的语法已得到简化。为了管理 IPsec 策略，扩展了 `ipsecconf` 命令。对管理 IPsec 策略而言，`ifconfig` 命令已过时。
- 从此发行版起，删除了 `/etc/ipnodes` 文件。可使用 `/etc/hosts` 文件来配置网络 IPv6 地址。

Solaris 10 1/06：从此发行版开始，IKE 与 NAT 遍历支持完全兼容，如 RFC 3947 和 RFC 3948 中所述。IKE 操作使用加密框架中的 PKCS #11 库，从而提高了性能。

加密框架为使用 metaslot 的应用程序提供了 softtoken 密钥库。IKE 使用 metaslot 时，可以选择在磁盘上、已连接的板上或在 softtoken 密钥库中存储密钥。

- 有关如何使用 softtoken 密钥库的信息，请参见 `cryptoadm(1M)` 手册页。
- 有关 Oracle Solaris 新增功能的完整列表，请参见《Oracle Solaris 10 1/13 新增功能》。

IPsec 介绍

IPsec 通过验证包、加密包或同时执行这两种操作来保护 IP 包。IPsec 在 IP 模块内执行。因此，Internet 应用程序可以直接利用 IPsec，而不必配置自身以使用 IPsec。若使用得当，IPsec 是保证网络通信安全的有效工具。

IPsec 保护涉及以下主要组件：

- **安全协议**－IP 数据报保护机制。authentication header（验证头）(AH) 包括 IP 包的散列并确保完整性。数据报的内容没有加密，但是可以向接收者保证包的内容尚未更改，还可以向接收者保证包已由发送者发送。encapsulating security payload, ESP（封装安全有效负荷）对 IP 数据进行加密，因此在包传输过程中会遮蔽内容。ESP 还可以通过验证算法选项来确保数据的完整性。
- **安全关联 (Security Association, SA)**－应用于特定网络通信流的加密参数和 IP 安全协议。每个 SA 都有一个称为安全参数索引 (Security Parameters Index, SPI) 的唯一引用。
- **安全关联数据库 (Security Associations Database, SADB)**－将安全协议与 IP 目标地址和索引号进行关联的数据库。索引号称为 security parameter index, SPI（安全参数索引）。这三个元素（安全协议、目标地址和 SPI）会唯一标识合法的 IPsec 包。此数据库确保到达包目的地的受保护包可由接收者识别。接收者还可使用数据库中的信息解密通信、检验包未曾受到更改、重新组装包并将包发送到其最终目的地。
- **密钥管理**－针对加密算法和 SPI 生成和分发密钥。
- **安全机制**－用于保护 IP 数据报中的数据的验证和加密算法。
- **安全策略数据库 (Security Policy Database, SPD)**－用于指定要应用到包的保护级别的数据库。SPD 过滤 IP 通信来确定应该如何处理包。包可能被废弃，可以毫无阻碍地进行传送，或者也可以受到 IPsec 的保护。对于外发包，SPD 和 SADB 确定要应用的保护级别。对于传入包，SPD 帮助确定包的保护级别是否可接受。如果包受 IPsec 保护，将在对包进行解密和验证之后参考 SPD。

IPsec 将安全机制应用于发往 IP 目标地址的 IP 数据报。接收者使用其 SADB 中的信息来检验到达的包是否合法并对其进行解密。应用程序也可以调用 IPsec，以便在每个套接字级别将安全机制应用于 IP 数据报。

如果端口上的套接字为连接状态，且随后对此端口应用 IPsec 策略，则使用此套接字的通信不受 IPsec 保护。当然，将 IPsec 策略应用于端口**之后**，在此端口上打开的套接字将受 IPsec 策略保护。

IPsec RFC

Internet 工程任务组 (Internet Engineering Task Force, IETF) 已经发布了许多介绍 IP 层安全体系结构的互联网信息文档和标准 (Requests for Comment, RFC)。所有 RFC 均受 Internet 协会版权保护。有关指向 RFC 的链接，请参见 http://www.ietf.org/。以下 RFC 列表包含更为常见的 IP 安全参考：

- RFC 2411，"IP Security Document Roadmap"，1998 年 11 月
- RFC 2401，"Security Architecture for the Internet Protocol"，1998 年 11 月
- RFC 2402，"IP Authentication Header"，1998 年 11 月
- RFC 2406，"IP Encapsulating Security Payload (ESP)"，1998 年 11 月
- RFC 2408，"Internet Security Association and Key Management Protocol (ISAKMP)"，1998 年 11 月
- RFC 2407，"The Internet IP Security Domain of Interpretation for ISAKMP"，1998 年 11 月
- RFC 2409，"The Internet Key Exchange (IKE)"，1998 年 11 月
- RFC 3554，"On the Use of Stream Control Transmission Protocol (SCTP) with IPsec"，2003 年 7 月 [未在 Oracle Solaris 10 发行版中实现]

IPsec 术语

IPsec RFC 定义许多用于识别何时在系统上实现 IPsec 的术语。下表列出了 IPsec 术语，提供了常用的首字母缩略词并定义了每个术语。有关在密钥协商中使用的术语的列表，请参见表 22–1。

表 19–1 IPsec 术语、首字母缩略词和用法

IPsec 术语	首字母缩略词	定义
Security Association（安全关联）	SA	应用于特定网络通信流的加密参数和 IP 安全协议。SA 由三个元素定义：安全协议、唯一安全参数索引 (Security Parameter Index, SPI) 和 IP 目标。
Security Associations Database（安全关联数据库）	SADB	包含所有活动的安全关联的数据库。
Security parameter index（安全参数索引）	SPI	安全关联的索引值。SPI 是可以将具有相同 IP 目标和安全协议的 SA 区分开来的 32 位的值。
Security policy database（安全策略数据库）	SPD	确定外发包和传入包是否具有指定的保护级别的数据库。

表 19–1 IPsec 术语、首字母缩略词和用法 （续）

IPsec 术语	首字母缩略词	定义
Key exchange（密钥交换）		使用非对称加密算法生成密钥的过程。两种主要方法是 RSA 和 Diffie-Hellman。
Diffie-Hellman	DH	用于密钥生成和密钥验证的密钥交换算法。通常称为**经过验证的密钥交换**。
RSA	RSA	用于密钥生成和密钥分发的密钥交换算法。此协议以其三个创建者 Rivest、Shamir 和 Adleman 命名。
Internet Security Association and Key Management Protocol（Internet 安全关联和密钥管理协议）	ISAKMP	用于建立 SA 属性格式以及协商、修改和删除 SA 的通用框架。ISAKMP 是处理 IKE 交换的 IETF 标准。

IPsec 包流

图 19–1 显示了当已经在外发包上调用 IPsec 时，作为 IP datagram（IP 数据报）一部分的带有 IP 地址的包如何继续传送。此流程图说明了可以对包应用验证头 (Authentication Header, AH) 和封装安全有效负荷 (Encapsulating Security Payload, ESP) 实体的位置。如何应用这些实体以及如何选择算法将在后续各节中进行介绍。

图 19–2 显示了 IPsec 传入过程。

图 19–1 应用于外发包过程的 IPsec

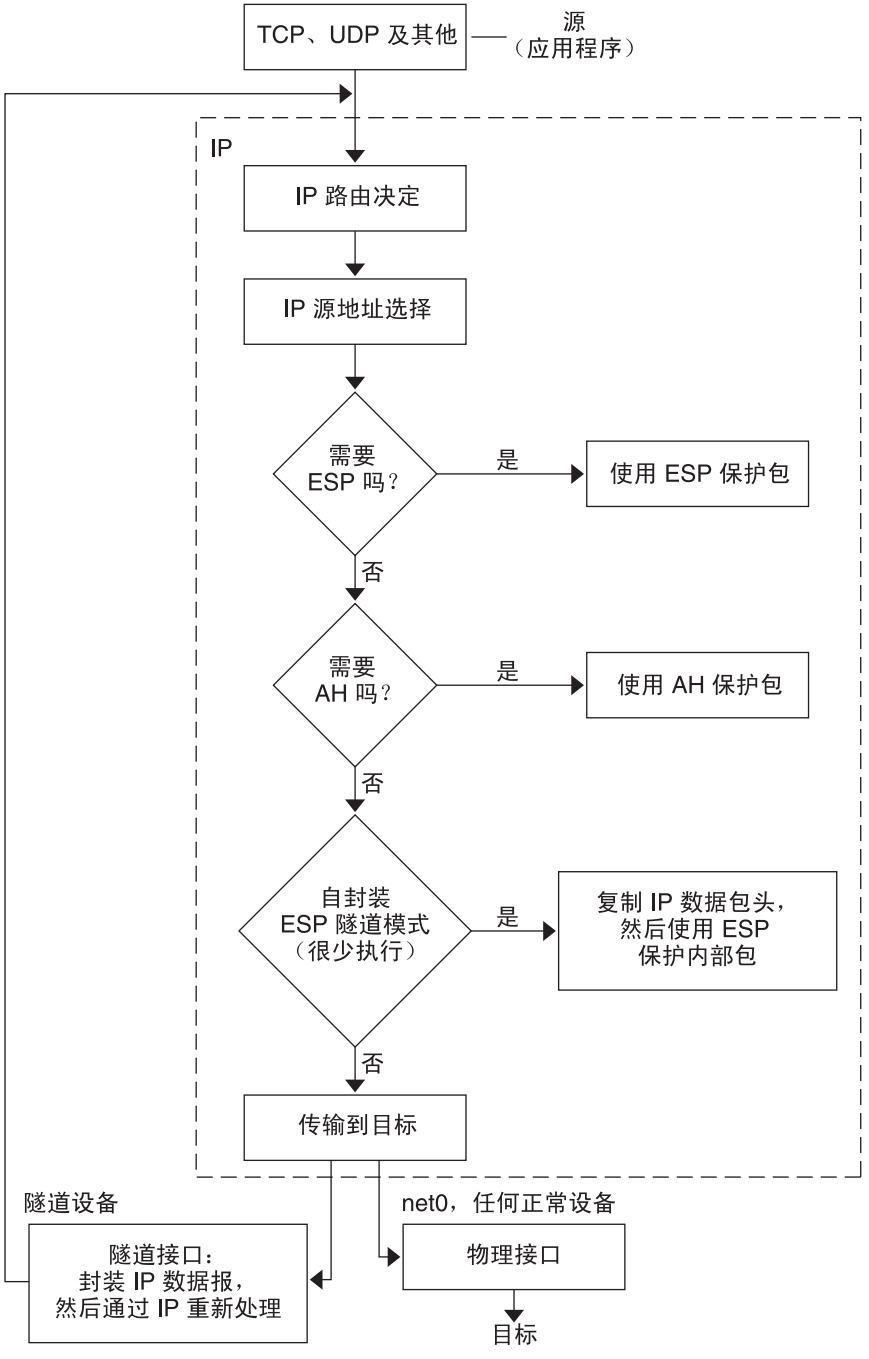
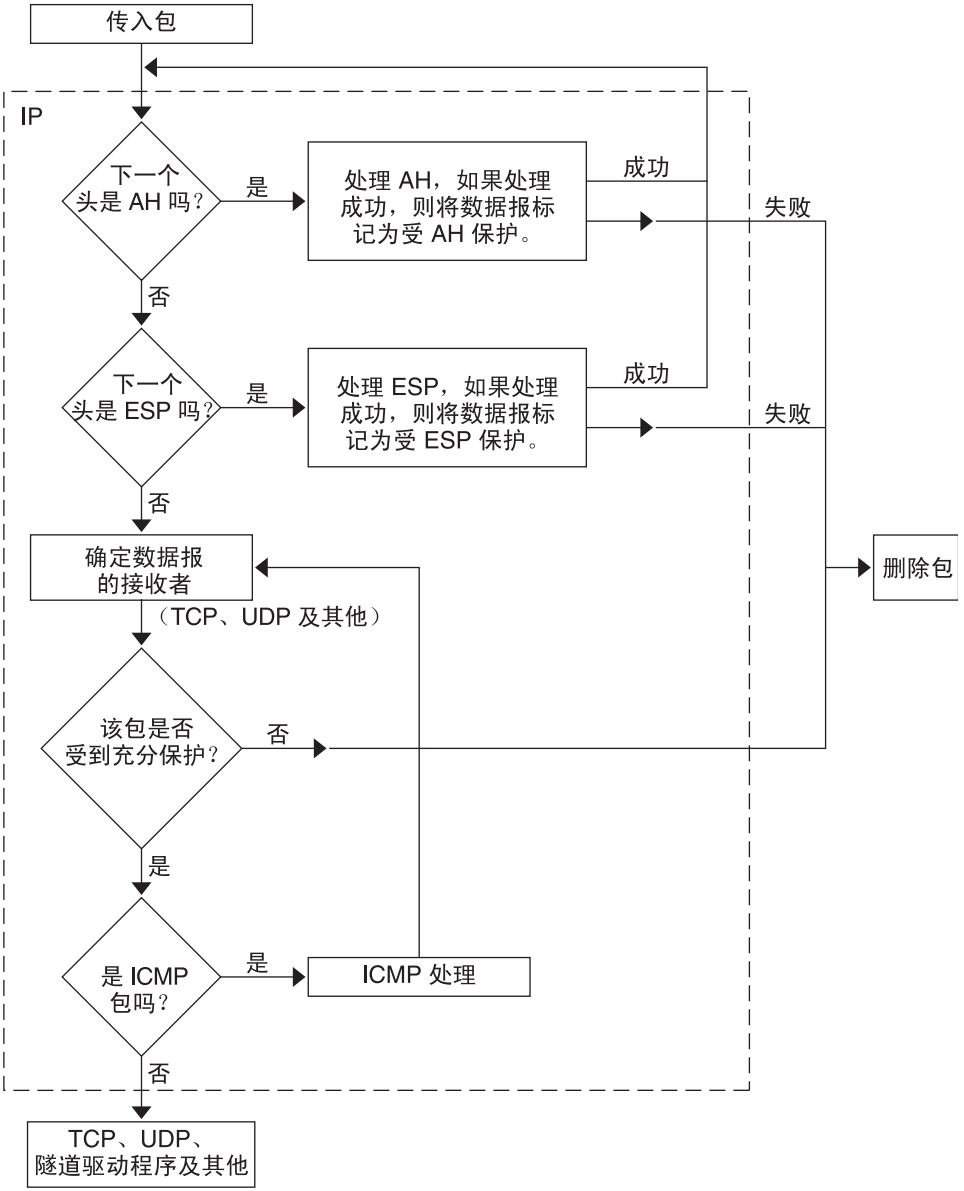


图 19–2 应用于传入包过程的 IPsec



IPsec 安全关联

IPsec **安全关联** (Security Association, SA) 指定由通信主机识别的安全属性。单个 SA 保护单一方向的数据，此保护针对单个主机或一组（多播）地址。由于多数通信为对等通信或客户机/服务器通信，因此，必须存在两个 SA 来保证两个方向的通信安全。

以下三个元素唯一地标识 IPsec SA :

- 安全协议（AH 或 ESP）
- 目标 IP 地址
- security parameter index, SPI（安全参数索引）

SPI 是任意 32 位的值，与 AH 或 ESP 包一起传输。`ipsecah(7P)` 和 `ipsecesp(7P)` 手册页说明了由 AH 和 ESP 提供的保护范围。完整性校验和值用于验证包。如果验证失败，则会丢弃包。

安全关联存储在**安全关联数据库** (Security Associations Database, SADB) 中。基于套接字的管理接口 `PF_KEY` 使特权应用程序可以管理数据库。例如，IKE 应用程序和 `ipseckeys` 命令会使用 `PF_KEY` 套接字接口。

- 有关 IPsec SADB 的更为完整的说明，请参见第 498 页中的 “IPsec 的安全关联数据库”。
- 有关如何管理 SADB 的更多信息，请参见 `pf_key(7P)` 手册页。

IPsec 中的密钥管理

安全关联 (Security Association, SA) 需要加密材料来进行验证和加密。对此加密材料的管理称为**密钥管理**。Internet 密钥交换 (Internet Key Exchange, IKE) 协议自动处理密钥管理。您还可以使用 `ipseckey` 命令手动管理密钥。

IPv4 和 IPv6 包上的 SA 可以使用任一密钥管理方法。除非您有充分的理由使用手动密钥管理，否则，请首选使用 IKE。

Oracle Solaris 的服务管理工具 (Service Management Facility, SMF) 功能为 IPsec 提供以下密钥管理服务 :

- `svc:/network/ipsec/ike:default` 服务－为用于进行自动密钥管理的 SMF 服务。`ike` 服务运行 `in.iked` 守护进程以提供自动密钥管理。有关 IKE 的说明，请参见第 22 章，Internet 密钥交换（概述）。有关 `in.iked` 守护进程的更多信息，请参见 `in.iked(1M)` 手册页。有关 `ike` 服务的信息，请参见第 551 页中的 “IKE 服务”。
- `svc:/network/ipsec/manual-key:default` 服务－为用于进行手动密钥管理的 SMF 服务。`manual-key` 服务运行带有各种选项的 `ipseckey` 命令来手动管理密钥。有关 `ipseckey` 命令的说明，请参见第 498 页中的 “IPsec 中用于生成 SA 的实用程序”。有关 `ipseckey` 命令选项的详细说明，请参见 `ipseckey(1M)` 手册页。

在 Solaris 10 4/09 发行版之前的发行版中，in.iked 和 ipseckey 命令用于管理加密材料。

- in.iked 守护进程提供自动密钥管理。有关 IKE 的说明，请参见第 22 章，Internet 密钥交换（概述）。有关 in.iked 守护进程的更多信息，请参见 in.iked(1M) 手册页。
- ipseckey 命令提供手动密钥管理。有关此命令的说明，请参见第 498 页中的 “IPsec 中用于生成 SA 的实用程序”。有关 ipseckey 命令选项的详细说明，请参见 ipseckey(1M) 手册页。

IPsec 保护机制

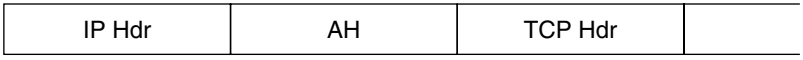
IPsec 提供了两种用于保护数据的安全协议：

- 验证头 (Authentication Header, AH)
- 封装安全有效负荷 (Encapsulating Security Payload, ESP)

AH 使用验证算法来保护数据。ESP 使用加密算法来保护数据。ESP 应只与验证机制一起使用。如果不将遍历 NAT，可以将 ESP 与 AH 结合使用。或者，可以将验证算法和加密机制与 ESP 一起使用。

验证头

authentication header（验证头）为数据报提供了数据验证、高完整性以及重放保护。AH 保护 IP 数据报的更为重要的部分。如下图所示，AH 插在 IP 数据包头和传输头之间。

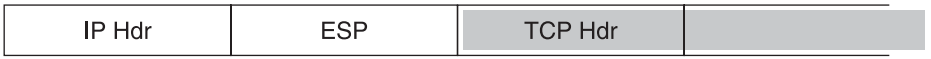


传输头可以是 TCP、UDP、SCTP 或 ICMP。如果使用的是 tunnel（隧道），则传输头可以是另一个 IP 数据包头。

封装安全有效负荷

encapsulating security payload, ESP（封装安全有效负荷）模块为 ESP 所封装的内容提供了保密性。ESP 也提供 AH 提供的服务。但是，ESP 仅为 ESP 所封装的数据报部分提供保护。ESP 提供可选的验证服务以确保受保护的包的完整性。因为 ESP 使用启用了加密的技术，因此提供 ESP 的系统可能会受进出口控制法制约。

由于 ESP 封装其数据，因此 ESP 仅保护数据报中跟在其后的数据，如下图所示。



加密的

在 TCP 包中，ESP 仅封装 TCP 数据包头及其数据。如果包是 IP-in-IP 数据报，则 ESP 会保护内部的 IP 数据报。由于每个套接字的策略允许**自封装**，因此，ESP 可以在需要时封装 IP 选项。

如果设置了自封装，会生成 IP 数据包头的副本来构建 IP-in-IP 数据报。例如，如果未在 TCP 套接字上设置自封装，会以下列格式发送数据报：

```
[ IP(a -> b) options + TCP + data ]
```

如果在 TCP 套接字上设置了自封装，则会以下列格式发送数据报：

```
[ IP(a -> b) + ESP [ IP(a -> b) options + TCP + data ] ]
```

有关进一步介绍，请参见第 436 页中的 “IPsec 中的传输模式和隧道模式”。

使用 AH 和 ESP 时的安全注意事项

下表比较了由 AH 和 ESP 提供的保护。

表 19–2 由 IPsec 中的 AH 和 ESP 提供的保护

协议	包范围	保护	防止的攻击
AH	保护包中从 IP 数据包头到传输层头的内容	提供高完整性、数据验证： ■ 确保接收者接收到的正是发送者发送的内容 ■ 在 AH 没有启用重放保护时容易受到重放攻击影响	重放、剪贴
ESP	保护数据报中紧跟在 ESP 之后的包。	使用加密选项时，对 IP 有效负荷进行加密。保证保密性	窃听
		使用验证选项时，提供与 AH 相同的有效负荷保护	重放、剪贴
		同时使用两个选项时，提供高完整性、数据验证和保密性	重放、剪贴、窃听

IPsec 中的验证算法和加密算法

IPsec 安全协议使用两种类型的算法，即验证和加密。AH 模块使用验证算法。ESP 模块可以使用加密算法以及验证算法。您可以使用 `ipsecalgs` 命令获取系统上的算法及其属性的列表。有关更多信息，请参见 `ipsecalgs(1M)` 手册页。您也可以使用 `getipsecalgbyname(3NSL)` 手册页中介绍的功能来检索算法属性。

IPsec 使用加密框架访问算法。加密框架为算法提供了一个中心系统信息库，同时还提供了其他服务。使用此框架，IPsec 可以利用高性能的加密硬件加速器。

有关更多信息，请参见以下内容：

- 《System Administration Guide: Security Services》中的第 13 章 “Oracle Solaris Cryptographic Framework (Overview)”
- 《Oracle Solaris 10 开发者安全性指南》中的第 8 章 “Oracle Solaris 加密框架介绍”

IPsec 中的验证算法

验证算法将生成完整性校验和值或基于数据和密钥的**摘要**。AH 模块使用验证算法。ESP 模块也可以使用验证算法。

IPsec 中的加密算法

加密算法使用密钥来加密数据。IPsec 中的 ESP 模块使用加密算法。算法以**块大小**为单位对数据进行操作。

不同的 Oracle Solaris 发行版提供不同的缺省加密算法。

从此 Solaris 10 7/07 发行版开始，Solaris 加密工具包的内容由 Solaris 安装介质安装。此发行版会添加以下 SHA2 验证算法：sha256、sha384 和 sha512。SHA2 实现符合 RFC 4868 规范。此发行版还会添加较大的 Diffie-Hellman 组：2048 位（组 14）、3072 位（组 15）和 4096 位（组 16）。请注意，采用酷线程 (CoolThreads) 技术的 Oracle Sun 系统仅会加速 2048 位组。

注意 – 从 Solaris 10 7/07 发行版开始，将不会在您的系统中添加 Solaris 加密工具包。该工具包降低了您系统上加密的修补程序级别。此工具包与系统上的加密不兼容。

IPsec 保护策略

IPsec 保护策略可以使用任何安全机制。IPsec 策略可以在以下级别应用：

- 在系统范围级别
- 在每个套接字级别

IPsec 会将系统范围的策略应用于外发数据报和传入数据报。外发数据报既可以在受保护的情况下发送，也可以在不受保护的情况下发送。如果应用了保护，则算法可能是特定的，也可能是非特定的。由于存在系统可识别的其他数据，因此可以将其他一些规则应用于外发数据报。传入数据报可以被接受或丢弃。确定丢弃还是接受传入数据报时取决于若干个条件，这些条件有时会重叠或冲突。可以通过确定首先要解析的规则来解决冲突。将自动接受通信，但是当策略项表明通信应绕过所有其他策略时除外。

可以绕过通常保护数据报的策略。您既可以在系统范围策略内指定例外，也可以在每个套接字策略中请求绕过。对于系统内的通信，将执行策略，但是不会应用实际的安全机制。相反，应用于系统内部包上的外发策略将转移到应用了那些机制的传入包。

可以使用 `ipsecinit.conf` 文件和 `ipsecconf` 命令来配置 IPsec 策略。有关详细信息和示例，请参见 `ipsecconf(1M)` 手册页。

IPsec 中的传输模式和隧道模式

IPsec 标准定义了 IPsec 操作的两种不同模式：**传输模式**和**隧道模式**。模式不影响包的编码。在每种模式下，包受 AH、ESP，或二者的保护。如果内部包是 IP 包，这两种模式在策略应用程序方面有所不同，如下所示：

- 在传输模式下，外部头决定保护内部 IP 包的 IPsec 策略。
- 在隧道模式下，内部 IP 包决定保护其内容的 IPsec 策略。

在传输模式下，外部头、下一个头以及下一个头支持的任何端口都可用于确定 IPsec 策略。实际上，IPsec 可在一个端口不同粒度的两个 IP 地址之间强制实行不同的传输模式策略。例如，如果下一个头是 TCP（支持端口），则可为外部 IP 地址的 TCP 端口设置 IPsec 策略。类似地，如果下一个头是 IP 数据包头，外部头和内部 IP 数据包头可用于决定 IPsec 策略。

隧道模式仅适用于 IP-in-IP 数据报。如果在家中的计算机用户要连接到中心计算机位置，以隧道模式进行隧道连接将会很有用。在隧道模式下，IPsec 策略强制实施于内部 IP 数据报的内容中。可针对不同的内部 IP 地址强制实施不同的 IPsec 策略。也就是说，内部 IP 数据包头、其下一个头及下一个头支持的端口，可以强制实施策略。与传输模式不同，在隧道模式下，外部 IP 数据包头不指示其内部 IP 数据报的策略。

因此，在隧道模式下，可为路由器后面的 LAN 的子网和这些子网上的端口指定 IPsec 策略。也可在这些子网上为特定的 IP 地址（即主机）指定 IPsec 策略。这些主机的端口也可以具有特定的 IPsec 策略。但是，如果有动态路由协议在隧道上运行，请勿使用子网选择或地址选择，因为对等网络上的网络拓扑的视图可能会更改。更改可能使静态 IPsec 策略失效。有关包括配置静态路由的隧道设置过程示例，请参见第 463 页中的 “使用 IPsec 保护 VPN（任务列表）”。

在 Oracle Solaris 中，只能在 IP 隧道连接网络接口上强制执行隧道模式。`ipsecconf` 命令提供 `tunnel` 关键字来选择 IP 隧道连接网络接口。当规则中出现 `tunnel` 关键字时，在此规则中指定的所有选定器都应用到内部包中。

在传输模式下，ESP、AH 或二者可以保护该数据报。

下图显示了不受保护的 TCP 包的 IP 数据包头。

图 19–3 携带 TCP 信息的不受保护的 IP 包

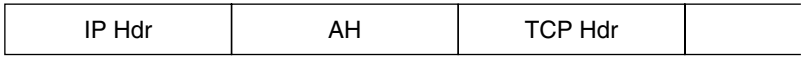
在传输模式下，ESP 按下图所示的方式保护数据。阴影部分表示包的加密部分。

图 19–4 携带 TCP 信息的受保护的 IP 包

加密的

在传输模式下，AH 按下图所示的方式保护数据。

图 19–5 由验证头保护的包



甚至在传输模式下，AH 保护也会涵盖大多数 IP 数据包头。

在隧道模式下，整个数据报处于 IPsec 数据包头的保护**之内**。图 19–3 中的数据报由外部 IPsec 数据包头（在此示例中为 ESP）以隧道模式保护，如下图所示。

图 19–6 以隧道模式保护的 IPsec 包

加密的

`ipsecconf` 命令包括用于将隧道设置为隧道模式或传输模式的关键字。

- 有关每个套接字策略的详细信息，请参见 `ipsec(7P)` 手册页。
- 有关每个套接字策略的示例，请参见第 448 页中的 “如何使用 IPsec 保护 Web 服务器使之免受非 Web 通信影响”。
- 有关隧道的更多信息，请参见 `ipsecconf(1M)` 手册页。
- 有关隧道配置的示例，请参见第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”。

虚拟专用网络和 IPsec

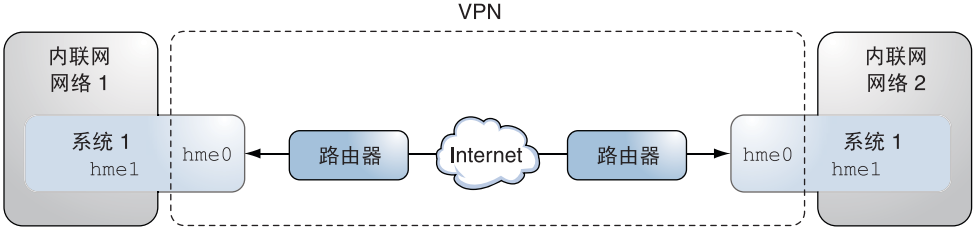
已配置的隧道是点对点接口。使用隧道，可以将一个 IP 包封装到另一个 IP 包中。正确配置的隧道同时要求隧道源和隧道目标。有关更多信息，请参见 tun(7M) 手册页和针对 IPv6 支持配置隧道。

隧道可创建明显的 IP physical interface（物理接口）。物理链接的完整性取决于底层安全协议。如果您安全地设置了安全关联 (Security Association, SA)，则可以信任隧道。退出隧道的包必须源于隧道目标中指定的对等设备。如果此信任存在，则可以使用按接口 IP 转发来创建 virtual private network, VPN（虚拟专用网络）。

您可以对 VPN 添加 IPsec 保护。IPsec 保证连接安全。例如，使用 VPN 技术将办公室与独立网络连接的组织可以添加 IPsec 来保证两个办公室之间的通信安全。

下图说明了两个办公室如何使用其网络系统上部署的 IPsec 来形成 VPN。

图 19–7 虚拟专用网络



有关设置过程的详细示例，请参见第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”。

有关使用 IPv6 地址的类似示例，请参见第 474 页中的 “如何使用 IPv6 在隧道模式下通过 IPsec 隧道保护 VPN”。

IPsec 和 NAT 遍历

IKE 可以通过 NAT 盒 (NAT box) 来协商 IPsec SA。此功能使系统可以从远程网络安全地连接，即使当系统位于 NAT 设备之后也可如此。例如，在家工作或从会议地点登录的雇员可以使用 IPsec 保护其通信。

NAT 表示网络地址转换。NAT 盒 (NAT box) 用于将专用内部地址转换为唯一的 Internet 地址。NAT 常见于 Internet 的公共访问点，例如宾馆。有关更全面的论述，请参见第 565 页中的 “使用 IP 过滤器的 NAT 功能”。

当 NAT 盒 (NAT box) 位于通信系统之间时使用 IKE 的能力称为 NAT 遍历，即 NAT-T。在 Oracle Solaris 10 发行版中，NAT-T 具有以下限制：

- NAT-T 不能利用由 Sun Crypto Accelerator 4000 板提供的 IPsec ESP 加速。但是，Sun Crypto Accelerator 4000 板的 IKE 加速可正常工作。
- 由于 AH 协议取决于未更改的 IP 数据包头，因此 AH 不能用于 NAT-T。ESP 协议可用于 NAT-T。
- NAT 盒 (NAT box) 不使用特殊的处理规则。使用特殊 IPsec 处理规则的 NAT 盒 (NAT box) 可能会干扰 NAT-T 的实现。
- 仅当 IKE 启动器是位于 NAT 盒 (NAT box) 之后的系统时，NAT-T 才运行。IKE 响应者不能位于 NAT 盒 (NAT box) 之后，除非此盒已经过编程可以将 IKE 包转发到位于盒之后的相应单个系统。

以下 RFC 介绍了 NAT 的功能和 NAT-T 的限制。可以从 `http://www.rfc-editor.org` 检索 RFC 的副本。

- RFC 3022，"Traditional IP Network Address Translator (Traditional NAT)"，2001 年 1 月
- RFC 3715，"Psec-Network Address Translation (NAT) Compatibility Requirements"，2004 年 3 月
- RFC 3947，"Negotiation of NAT-Traversal in the IKE"，2005 年 1 月
- RFC 3948，"UDP Encapsulation of IPsec Packets"，2005 年 1 月

有关如何通过 NAT 使用 IPsec 的信息，请参见第 536 页中的 “为移动系统配置 IKE（任务列表）”。

IPsec 和 SCTP

Oracle Solaris 支持流控制传输协议 (Streams Control Transmission Protocol, SCTP)。支持使用 SCTP 协议和 SCTP 端口号来指定 IPsec 策略，但是这种方法不可靠。RFC 3554 中指定的 SCTP 的 IPsec 扩展尚未实现。这些限制可能会使为 SCTP 创建 IPsec 策略的过程更为复杂。

SCTP 可以在单个 SCTP 关联的上下文中使用多个源地址和目标地址。当 IPsec 策略应用于单个源地址或目标地址时，通信可能会在 SCTP 切换此关联的源地址或目标地址时失败。IPsec 策略仅识别初始地址。有关 SCTP 的信息，请阅读 RFC 和 第 37 页中的 “SCTP 协议”。

IPsec 和 Oracle Solaris Zones

对于共享 IP 区域，IPsec 是在全局区域中配置的。IPsec 策略配置文件 `ipsecinit.conf` 仅存在于全局区域中。该文件既可具有应用到非全局区域的项，又可具有应用到全局区域的项。

对于专用 IP 区域，在每个非全局区域中配置 IPsec。

有关如何在区域上使用 IPsec 的信息，请参见第 443 页中的 “使用 IPsec 保护通信（任务列表）”。有关区域的信息，请参见《System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones》中的第 16 章 “Introduction to Solaris Zones”。

IPsec 和逻辑域

IPsec 与逻辑域一同工作。逻辑域必须运行包含 IPsec 的 Oracle Solaris 版本，如 Oracle Solaris 10 发行版。

要创建逻辑域，必须使用 SPARC 的 Oracle VM Server（之前称为 Logical Domains）。有关如何配置逻辑域的信息，请参见《Oracle VM Server for SPARC 2.2 管理指南》或《Oracle VM Server for SPARC 2.0 Administration Guide》。

IPsec 实用程序和文件

表 19–3 介绍了用于配置和管理 IPsec 的文件、命令和服务标识符。为了体现完整性，此表包括密钥管理文件、套接字接口和命令。

从 Solaris 10 4/09 发行版开始，IPsec 由 SMF 来管理。有关服务标识符的更多信息，请参见《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。

- 有关在网络中实现 IPsec 的说明，请参见第 443 页中的 “使用 IPsec 保护通信（任务列表）”。
- 有关 IPsec 实用程序和文件的更多详细信息，请参见第 21 章，IP 安全体系结构（参考信息）。

表 19–3 所选 IPsec 实用程序和文件的列表

IPsec 实用程序、文件或服务	说明	手册页
`svc:/network/ipsec/ipsecalgs`	在当前发行版中，为管理 IPsec 算法的 SMF 服务。	`ipsecalgs(1M)`
`svc:/network/ipsec/manual-key`	在当前发行版中，为手动管理加密 IPsec SA 的 SMF 服务。	`ipseckey(1M)`
`svc:/network/ipsec/policy`	在当前发行版中，为管理 IPsec 策略的 SMF 服务。	`smf(5)`、`ipsecconf(1M)`

表 19–3 所选 IPsec 实用程序和文件的列表 （续）

IPsec 实用程序、文件或服务	说明	手册页
svc:/network/ipsec/ike	在当前发行版中，为自动管理 IPsec SA 的 SMF 服务（使用 IKE）。	smf(5)、in.iked(1M)
/etc/inet/ipsecinit.conf 文件	IPsec 策略文件。在 Solaris 10 4/09 发行版之前的发行版中，如果此文件存在，则会在引导时激活 IPsec。 在当前发行版中，SMF policy 服务在系统引导时使用此文件配置 IPsec 策略。	ipsecconf(1M)
ipsecconf 命令	IPsec 策略命令。用于查看和修改当前的 IPsec 策略，以及进行测试。在 Solaris 10 4/09 发行版之前的发行版中，引导脚本使用 ipsecconf 来读取 /etc/inet/ipsecinit.conf 文件并激活 IPsec。 在当前发行版中，ipsecconf 由 SMF policy 服务使用以在系统引导时配置 IPsec 策略。	ipsecconf(1M)
PF_KEY 套接字接口	安全关联数据库 (Security Associations Database, SADB) 的接口。处理手动密钥管理和自动密钥管理。	pf_key(7P)
ipseckey 命令	IPsec SA 加密命令。ipseckey 是 PF_KEY 接口的命令行前端。ipseckey 可以创建、销毁或修改 SA。	ipseckey(1M)
/etc/inet/secret/ipseckeys 文件	包含手动加密的 SA。在 Solaris 10 4/09 发行版之前的发行版中，如果 ipsecinit.conf 文件存在，则会在引导时自动读取 ipseckeys 文件。 在当前发行版中，ipseckeys 由 SMF manual-key 服务使用以在系统引导时手动配置 SA。	
ipsecalgs 命令	IPsec 算法命令。可用于查看和修改 IPsec 算法及其属性的列表。 在当前发行版中，由 SMF ipsecalgs 服务使用以在系统引导时使已知 IPsec 算法与内核同步。	ipsecalgs(1M)
/etc/inet/ipsecalgs 文件	包含已配置的 IPsec 协议和算法定义。此文件由 ipsecalgs 命令管理，并且决不能手动编辑。	
/etc/inet/ike/config 文件	IKE 配置和策略文件。缺省情况下，此文件不存在。在 Solaris 10 4/09 发行版之前的发行版中，如果此文件存在，IKE 守护进程 in.iked 会提供自动密钥管理。密钥管理基于 /etc/inet/ike/config 文件中的规则和全局参数。请参见 第 507 页中的 “IKE 实用程序和文件”。 在当前发行版中，如果此文件存在，svc:/network/ipsec/ike 服务会启动 IKE 守护进程 in.iked 来提供自动密钥管理。	ike.config(4)

Oracle Solaris 10 发行版中 IPsec 的更改

有关 Oracle Solaris 新增功能的完整列表，请参见《Oracle Solaris 10 1/13 新增功能》。自 Solaris 9 发行版以来，IPsec 包括以下功能：

- 当连接 Sun Crypto Accelerator 4000 板时，此板将自动高速缓存使用此板的以太网接口的包的 IPsec SA。此板还加快了 IPsec SA 的处理速度。
- IPsec 可以利用通过 IPv6 网络使用 IKE 进行的自动密钥管理。有关更多信息，请参见第 22 章，Internet 密钥交换（概述）。

  有关 IKE 新增功能，请参见第 508 页中的 “Oracle Solaris 10 发行版对 IKE 的更改”。
- `ipseckey` 命令的解析器提供更为明确的帮助。`ipseckey monitor` 命令为每个事件标记时间。有关详细信息，请参见 `ipseckey(1M)` 手册页。
- IPsec 算法现在来自中央存储位置，即 Oracle Solaris 的加密框架功能。`ipsecalgs(1M)` 手册页介绍了可用算法的特征。这些算法针对运行它们的体系结构进行了优化。有关加密框架的说明，请参见《System Administration Guide: Security Services》中的第 13 章 “Oracle Solaris Cryptographic Framework (Overview)”。
- IPsec 在全局区域中运行。在全局区域中管理用于非全局区域的 IPsec 策略。在全局区域中为非全局区域手动创建和管理加密材料。不能使用 IKE 为非全局区域生成密钥。有关区域的更多信息，请参见《System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones》中的第 16 章 “Introduction to Solaris Zones”。
- IPsec 策略可以与流控制传输协议 (Streams Control Transmission Protocol, SCTP) 和 SCTP 端口号一起使用。但是，该实现尚不完整。尚未实现 RFC 3554 中指定的 SCTP 的 IPsec 扩展。这些限制可能会使为 SCTP 创建 IPsec 策略的过程更为复杂。有关详细信息，请参阅 RFC。另请参见第 439 页中的 “IPsec 和 SCTP”和第 37 页中的 “SCTP 协议”。
- IPsec 和 IKE 可以保护源于 NAT 盒 (NAT box) 之后的通信。有关详细信息和限制，请参见第 438 页中的 “IPsec 和 NAT 遍历”。有关过程，请参见第 536 页中的 “为移动系统配置 IKE（任务列表）”。

◆ ◆ ◆ 第 20 章

配置 IPsec（任务）

本章提供了在网络中实现 IPsec 的过程。以下任务列表中介绍了这些过程：

- 第 443 页中的“使用 IPsec 保护通信（任务列表）”
- 第 463 页中的“使用 IPsec 保护 VPN（任务列表）”

有关 IPsec 的概述信息，请参见第 19 章，IP 安全体系结构（概述）。有关 IPsec 的参考信息，请参见第 21 章，IP 安全体系结构（参考信息）。

使用 IPsec 保护通信（任务列表）

以下任务列表提供了指向在一个或多个系统之间设置 IPsec 的过程的链接。`ipsecconf(1M)`、`ipseckey(1M)` 和 `ifconfig(1M)` 手册页也在各自的“示例”部分中介绍了有用的过程。

任务	说明	参考
保证两个系统之间的通信安全。	确保系统间传送的包的安全。	第 445 页中的“如何使用 IPsec 保证两个系统之间的通信安全”
使用 IPsec 策略保证 Web 服务器的安全。	要求非 Web 通信使用 IPsec。Web 客户机由特定端口识别，这些端口将跳过 IPsec 检查。	第 448 页中的“如何使用 IPsec 保护 Web 服务器使之免受非 Web 通信影响”
显示 IPsec 策略。	按照执行的顺序显示当前正在执行的 IPsec 策略。	第 451 页中的“如何显示 IPsec 策略”
生成随机数。	为手动创建的安全关联生成加密材料的随机数。	第 451 页中的“如何在 Oracle Solaris 系统上生成随机数” 《System Administration Guide: Security Services》中的“How to Generate a Symmetric Key by Using the pktool Command”

任务	说明	参考
手动创建或替换安全关联。	为安全关联提供原始数据： ■ IPsec 算法名称和加密材料 ■ 安全参数索引 (Security Parameter Index, SPI) ■ IP 源地址和目标地址及其他参数	第 453 页中的 “如何手动创建 IPsec 安全关联”
检查 IPsec 是否正在保护包。	检查 snoop 输出以了解指示如何保护 IP 数据报的特定头。	第 457 页中的 “如何检验包是否受 IPsec 保护”
（可选）创建网络安全角色。	创建可以设置安全网络，但权限级别低于超级用户的角色。	第 458 页中的 “如何配置网络安全角色”
将 IPsec 和加密材料作为一组 SMF 服务来管理。	介绍何时以及如何使用相应命令来启用、禁用、刷新和重新启动服务。此外，还介绍了用于更改服务的属性值的命令。	第 460 页中的 “如何管理 IKE 和 IPsec 服务”
设置安全的虚拟专用网络 (Virtual Private Network, VPN)。	在 Internet 中的两个系统之间设置 IPsec。	第 463 页中的 “使用 IPsec 保护 VPN（任务列表）”

使用 IPsec 保护通信

本节提供保证两个系统之间的通信安全以及保证 Web 服务器的安全的过程。要保护 VPN，请参见第 463 页中的 “使用 IPsec 保护 VPN（任务列表）”。其他过程提供加密材料和安全关联并检验 IPsec 是否按照配置工作。

以下信息适用于所有的 IPsec 配置任务：

- **IPsec 和区域** – 要管理共享 IP 非全局区域的 IPsec 策略和密钥，请在全局区域中创建 IPsec 策略文件，然后从全局区域运行 IPsec 配置命令。请使用对应于要配置的非全局区域的源地址。您还可以在全局区域中为全局区域配置 IPsec 策略和密钥。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。从 Solaris 10 7/07 发行版开始，可以使用 IKE 在非全局区域中管理密钥。
- **IPsec 和 RBAC** – 要使用角色来管理 IPsec，请参见《System Administration Guide: Security Services》中的第 9 章 “Using Role-Based Access Control (Tasks)”。有关示例，请参见第 458 页中的 “如何配置网络安全角色”。
- **IPsec 和 SCTP** – 可以使用 IPsec 来保护流控制传输协议 (Streams Control Transmission Protocol, SCTP) 关联，但使用时必须谨慎。有关更多信息，请参见第 439 页中的 “IPsec 和 SCTP”。

▼ 如何使用 IPsec 保证两个系统之间的通信安全

假设此过程具有以下设置：

- 两个系统的名称为 `enigma` 和 `partym`。
- 每个系统都有两个地址，一个 IPv4 地址和一个 IPv6 地址。
- 每个系统都需要采用 AES 算法的 ESP 加密（此算法需要 128 位的密钥）和采用 SHA1 消息摘要的 ESP 验证（此消息摘要需要 160 位的密钥）。
- 每个系统都使用共享安全关联。

  如果使用共享 SA，则仅需要一对 SA 来保护两个系统。

开始之前 必须位于全局区域中才能为系统或共享 IP 区域配置 IPsec 策略。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。有关示例，请参见示例 20–1。

---

2 在每个系统上，检查主机项。

在当前发行版中，将主机项添加到 `/etc/inet/hosts` 文件。

在运行 Solaris 10 7/07 之前的发行版的系统上，将 IPv4 和 IPv6 项添加到 `/etc/inet/ipnodes` 文件中。一个系统的项在文件中必须是连续的。有关系统配置文件的更多信息，请参见第 205 页中的 “TCP/IP 配置文件”和第 11 章，IPv6 详解（参考）。

如果您仅连接具有 IPv4 地址的系统，则需要修改 `/etc/inet/hosts` 文件。在此示例中，连接的系统运行的是早期的 Solaris 发行版并且使用的是 IPv6 地址。

a. 在名为 `enigma` 的系统上，将以下内容键入到 `hosts` 或 `ipnodes` 文件中：

```
# Secure communication with partym
192.168.13.213 partym
2001::eeee:3333:3333 partym
```

b. 在名为 `partym` 的系统上，将以下内容键入到 `hosts` 或 `ipnodes` 文件中：

```
# Secure communication with enigma
192.168.116.16 enigma
2001::aaaa:6666:6666 enigma
```

将名称服务用于符号名称是不安全的。

3 **在每个系统上，创建 IPsec 策略文件。**

该文件名为 `/etc/inet/ipsecinit.conf`。有关示例，请参见 `/etc/inet/ipsecinit.sample` 文件。

4 **将 IPsec 策略项添加到 `ipsecinit.conf` 文件。**

a. **在 `enigma` 系统上添加以下策略：**

```
{laddr enigma raddr partym} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. **在 `partym` 系统上添加相同的策略：**

```
{laddr partym raddr enigma} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

有关 IPsec 策略项的语法，请参见 `ipsecconf(1M)` 手册页。

5 **在每个系统上，添加两个系统之间的一对 IPsec SA。**

您可以配置 Internet 密钥交换 (Internet Key Exchange, IKE) 来自动创建 SA，也可以手动添加 SA。

注 – 您应该使用 IKE，除非您有充分的理由手动生成和维护密钥。IKE 密钥管理比手动密钥管理更为安全。

- 按照第 509 页中的 “配置 IKE（任务列表）”中的配置过程之一来配置 IKE。有关 IKE 配置文件的语法，请参见 `ike.config(4)` 手册页。
- 要手动添加 SA，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。

6 **启用 IPsec 策略。**

- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请重新引导系统。**

```
# init 6
```

然后，转至第 457 页中的 “如何检验包是否受 IPsec 保护”。

- **从 Solaris 10 4/09 发行版开始，请刷新 IPsec 服务并启用密钥管理服务。**

完成步骤 7 到步骤 10。

7 **检验 IPsec 策略文件的语法。**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

修复任何错误、检验文件的语法，然后继续。

8 **刷新 IPsec 策略。**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

IPsec 策略缺省情况下处于启用状态，因此要对其进行**刷新**。如果您已禁用了 IPsec 策略，请将其启用。

```
# svcadm enable svc:/network/ipsec/policy:default
```

9 **激活 IPsec 的密钥。**

- **如果在步骤 5 中配置了 IKE，请执行以下操作之一：**
  - **如果未启用 `ike` 服务，请将其启用。**

    ```
    # svcadm enable svc:/network/ipsec/ike:default
    ```

  - **如果已启用 `ike` 服务，请重新启动此服务。**

    ```
    # svcadm restart svc:/network/ipsec/ike:default
    ```

- **如果在步骤 5 中手动配置了密钥，请执行以下操作之一：**
  - **如果未启用 `manual-key` 服务，请将其启用。**

    ```
    # svcadm enable svc:/network/ipsec/manual-key:default
    ```

  - **如果已启用 `manual-key` 服务，请刷新此服务。**

    ```
    # svcadm refresh svc:/network/ipsec/manual-key:default
    ```

10 **验证是否对包进行了保护。**

有关过程，请参见第 457 页中的 “如何检验包是否受 IPsec 保护”。

示例 20–1 使用 `ssh` 连接时添加 IPsc 策略

在此示例中，管理员作为超级用户通过使用 `ssh` 命令访问第二个系统，在两个系统上配置 IPsc 策略和密钥。有关更多信息，请参见 `ssh(1)` 手册页。

- 首先，管理员通过执行上述过程的步骤 2 至步骤 5 来配置第一个系统。
- 接着，在不同的终端窗口中，管理员使用 `ssh` 命令登录到第二个系统。

  ```
  local-system # ssh other-system
  other-system #
  ```

- 在 `ssh` 会话的终端窗口中，管理员通过完成步骤 2 至步骤 6 来配置第二个系统的 IPsec 策略和密钥。
- 然后，管理员结束 `ssh` 会话。

  ```
  other-system # exit
  local-system #
  ```

- 最后，管理员通过完成步骤 6 在第一个系统上启用 IPsec 策略。

下次这两个系统进行通信（包括使用 `ssh` 连接）时，此通信将会受 IPsec 保护。

示例 20–2 在不重新引导的情况下使用 IPsec 保证通信安全

以下示例适用于运行 Solaris 10 4/09 发行版之前的发行版的情况。即，在您的发行版中，不将 IPsec 作为服务来管理。此示例介绍了如何在测试环境中实现 IPsec。在生产环境中，重新引导比运行 `ipsecconf` 命令更为安全。有关安全注意事项，请参见此示例的结尾。

选择以下选项之一，而不是在步骤 6 进行重新引导：

- 如果您已使用 IKE 创建加密材料，请停止 `in.iked` 守护进程，然后重新启动它。

```
# pkill in.iked
# /usr/lib/inet/in.iked
```

- 如果您手动添加了密钥，请使用 `ipseckey` 命令将 SA 添加到数据库中。

```
# ipseckey -c -f /etc/inet/secret/ipseckeys
```

然后使用 `ipsecconf` 命令激活 IPsec 策略。

```
# ipsecconf -a /etc/inet/ipsecinit.conf
```

安全注意事项 – 请在执行 `ipsecconf` 命令时阅读警告。已锁定的套接字（即已使用的套接字）提供了通向系统的不安全的后门。有关更广泛的讨论，请参见第 497 页中的 “ipsecinit.conf 和 ipsecconf 的安全注意事项”。

▼ 如何使用 IPsec 保护 Web 服务器使之免受非 Web 通信影响

安全的 Web 服务器允许 Web 客户机与 Web 服务对话。在安全的 Web 服务器上，不属于 Web 通信的通信**必须**通过安全检查。以下过程会绕过 Web 通信。此外，此 Web 服务器可以发出不安全的 DNS 客户机请求。所有其他通信都需要使用 AES 和 SHA-1 算法的 ESP。

开始之前 必须位于全局区域中才能配置 IPsec 策略。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。

您已完成了第 445 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”，因此实际环境符合以下状况：

- 两个系统之间的通信受 IPsec 保护。
- 生成了加密材料（无论是手动生成还是由 IKE 生成）。
- 已检验是否对包进行了保护。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2 **确定哪些服务需要绕过安全策略检查。**

对于 Web 服务器，这些服务包括 TCP 端口 80 (HTTP) 和 443 (安全 HTTP)。如果 Web 服务器提供 DNS（域名系统）名称查找，则服务器还可能需要针对 TCP（传输控制协议）和 UDP（用户数据报协议）包括端口 53。

3 **为 Web 服务器创建 IPsec 策略，并将其启用。**

- **从 Solaris 10 4/09 发行版开始，请执行步骤 4 至步骤 7。**
- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请执行步骤 8 至步骤 11。**

对于所有 Solaris 发行版，步骤 12 均是可选的。

4 **将 Web 服务器策略添加到 IPsec 策略文件。**

将以下行添加到 `/etc/inet/ipsecinit.conf` 文件：

```
# Web traffic that web server should bypass.
{lport  80 ulp tcp dir both} bypass {}
{lport 443 ulp tcp dir both} bypass {}

# Outbound DNS lookups should also be bypassed.
{rport 53 dir both} bypass {}

# Require all other traffic to use ESP with AES and SHA-1.
# Use a unique SA for outbound traffic from the port
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

此配置仅允许安全通信访问系统，跳过检查的例外情况在步骤 4 中进行了介绍。

5 **检验 IPsec 策略文件的语法。**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 **刷新 IPsec 策略。**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

7 **刷新 IPsec 的密钥。**

- **如果在第 445 页中的“如何使用 IPsec 保证两个系统之间的通信安全”的步骤 5 中配置了 IKE，请重新启动 `ike` 服务。**

  ```
  # svcadm restart svc:/network/ipsec/ike
  ```

- **如果在第 445 页中的“如何使用 IPsec 保证两个系统之间的通信安全”的步骤 5 中手动配置了密钥，请刷新 `manual-key` 服务。**

  ```
  # svcadm refresh svc:/network/ipsec/manual-key:default
  ```

您的设置已完成。（可选）您可以执行步骤 12。

8 在 /etc/inet目录中为 Web 服务器策略创建一个文件。

注 – 以下步骤用于配置运行 Solaris 10 4/09 发行版之前的发行版的 Web 服务器。

为此文件指定一个表明其用途的名称，例如 IPsecWebInitFile。在此文件中键入以下行：

```
# Web traffic that web server should bypass.
{lport  80 ulp tcp dir both} bypass {}
{lport 443 ulp tcp dir both} bypass {}

# Outbound DNS lookups should also be bypassed.
{rport 53 dir both} bypass {}

# Require all other traffic to use ESP with AES and SHA-1.
# Use a unique SA for outbound traffic from the port
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

此配置仅允许安全通信访问系统，跳过检查的例外情况在步骤 4 中进行了介绍。

9 将您在步骤 8 中创建的文件的内容复制到 /etc/inet/ipsecinit.conf 文件中。

10 使用只读权限保护 IPsecWebInitFile 文件。

```
# chmod 400 IPsecWebInitFile
```

11 在不重新引导的情况下保证 Web 服务器的安全。

选择以下选项之一：

- 如果您使用 IKE 管理密钥，请停止并重新启动 in.iked 守护进程。

  ```
  # pkill in.iked
  # /usr/lib/inet/in.iked
  ```

- 如果您手动管理密钥，请使用 ipseckey 和 ipsecconf 命令。

  请使用 IPsecWebInitFile 作为 ipsecconf 命令的参数。如果您使用 ipsecinit.conf 文件作为参数，则当文件中的策略已经在系统上实现时，ipsecconf 命令会生成错误。

  ```
  # ipseckey -c -f /etc/inet/secret/ipseckeys
  # ipsecconf -a /etc/inet/IPsecWebInitFile
  ```

注意 – 在执行 ipsecconf 命令时应阅读警告。已锁定的套接字（即已使用的套接字）提供了通向系统的不安全的后门。有关更广泛的讨论，请参见第 497 页中的“ipsecinit.conf 和 ipsecconf 的安全注意事项”。该警告也适用于重新启动 in.iked 守护进程。

您也可以重新引导。重新引导可确保 IPsec 策略在所有 TCP 连接上都有效。重新引导时，TCP 连接使用 IPsec 策略文件中的策略。

12 可选使远程系统与 Web 服务器进行非 Web 通信。

在远程系统的 `ipsecinit.conf` 文件中键入以下策略：

```
# Communicate with web server about nonweb stuff
#
{laddr webserver} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

仅当系统的 IPsec 策略匹配时，远程系统才能与 Web 服务器安全地进行非 Web 通信。

▼ 如何显示 IPsec 策略

当您发出不带任何参数的 `ipsecconf` 命令时，便可以查看在系统中配置的策略。

开始之前 必须在全局区域中运行 `ipsecconf` 命令。对于专用 IP 区域，请在非全局区域中运行 `ipsecconf` 命令。

1 承担拥有网络 IPsec 管理配置文件的角色或成为超级用户。

如果您运行的是 Solaris 10 4/09 发行版之前的发行版，则网络 IPsec 管理配置文件不可用。请使用网络安全配置文件。

有关如何创建拥有网络安全配置文件的角色并将该角色指定给用户的信息，请参见第 458 页中的 “如何配置网络安全角色”。

2 显示 IPsec 策略。

a. 按照全局 IPsec 策略项的添加顺序显示这些项。

```
$ ipsecconf
```

此命令将每项显示为后面跟有一个数字的**索引**。

b. 按照匹配项出现的顺序显示 IPsec 策略项。

```
$ ipsecconf -l -n
```

c. 按照匹配项出现的顺序显示 IPsec 策略项，包括每个隧道的项。

```
$ ipsecconf -L -n
```

▼ 如何在 Oracle Solaris 系统上生成随机数

如果您要手动指定密钥，则加密材料必须是随机的。对于 IPsec 密钥，其加密材料的格式是十六进制的。其他操作系统可能需要 ASCII 加密材料。如果要为 Oracle Solaris 系统生成加密材料，而且此系统与需要 ASCII 的操作系统通信，请参见示例 23–1。

如果站点具有随机数生成器，请使用该生成器。否则，可以使用 od 命令，并将 /dev/random 设备作为输入。有关更多信息，请参见 od(1) 手册页。

在 Solaris 10 4/09 发行版中，您也可以使用 pktool 命令。该命令的语法比 od 命令的语法更为简单。有关详细信息，请参见《System Administration Guide: Security Services》中的"How to Generate a Symmetric Key by Using the pktool Command"。

1 生成十六进制格式的随机数。

```
% od -x|-X -A n file | head -n
```

-x	显示十六进制格式的八进制转储。十六进制格式对加密材料有用。十六进制以 4 个字符的块显示。
-X	显示十六进制格式的八进制转储。十六进制以 8 个字符的块显示。
-A n	从显示中删除输入偏移基址。
file	作为随机数的源。
head -*n*	将显示限制在输出的前 *n* 行中。

2 合并输出以创建适当长度的密钥。

删除一行上的数字之间的空格，来创建 32 字符的密钥。一个 32 字符的密钥为 128 位。对于安全参数索引 (Security Parameter Index, SPI)，您应该使用 8 字符的密钥。密钥应该使用 0x 前缀。

示例 20–3 生成 IPsec 的加密材料

以下示例显示两行八个十六进制字符为一组的密钥。

```
% od -X -A n /dev/random | head -2
        d54d1536 4a3e0352 0faf93bd 24fd6cad
        8ecc2670 f3447465 20db0b0c c83f5a4b
```

通过合并第一行上的四个数字，可以创建一个 32 字符的密钥。以 0x 开头的 8 字符数提供了合适的 SPI 值，如 0xf3447465。

以下示例显示两行四个十六进制字符为一组的密钥。

```
% od -x -A n /dev/random | head -2
        34ce 56b2 8b1b 3677 9231 42e9 80b0 c673
        2f74 2817 8026 df68 12f4 905a db3d ef27
```

通过合并第一行上的八个数字，可以创建一个 32 字符的密钥。

▼ 如何手动创建 IPsec 安全关联

以下过程提供了第 445 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”过程的加密材料。您要为两个系统（partym 和 enigma）生成密钥。您在一个系统上生成密钥，然后在两个系统中使用在第一个系统中生成的密钥。

开始之前 必须位于全局区域中才能手动管理共享 IP 区域的加密材料。

1. **为 SA 生成加密材料。**

   您需要将三个十六进制随机数用于外发通信，三个十六进制随机数用于传入通信。因此，一个系统需要生成以下数字：

   - 两个作为 spi 关键字值的十六进制随机数。一个数字用于外发通信，一个数字用于传入通信。每个数字的长度最大可以为八个字符。
   - 两个用于验证的 SHA1 算法的十六进制随机数。对于 160 位密钥，每个数字的长度必须为 40 个字符。一个数字用于 dst enigma，另一个数字用于 dst partym。
   - 两个用于 ESP 加密的 AES 算法的十六进制随机数。对于 256 位密钥，每个数字的长度必须为 64 个字符。一个数字用于 dst enigma，另一个数字用于 dst partym。

   如果您的站点上有随机数生成器，请使用此生成器。您也可以使用 od 命令。有关过程，请参见第 451 页中的 “如何在 Oracle Solaris 系统上生成随机数”。

2. **在其中一个系统的系统控制台上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

   注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

3. **创建 SA。**

   - **从 Solaris 10 4/09 发行版开始，请执行步骤 8 至步骤 10。**
   - **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请执行步骤 4 至步骤 9。**

4. **启用 ipseckey 命令模式。**

   ```
   # ipseckey
   
   >
   ```

   > 提示符指明您处于 ipseckey 命令模式下。

5 如果您要替换现有的 SA，请刷新当前的 SA。

```
> flush
>
```

为防止入侵者有时间破坏您的 SA，您需要替换加密材料。

注 – 您必须在通信系统上协调密钥替换。当您在一个系统上替换 SA 时，也必须在远程系统上替换此 SA。

---

6 要创建 SA，请键入以下命令。

```
> add protocol spi random-hex-string \
src addr dst addr2 \
protocol-prefix_alg protocol-algorithm   \
protocol-prefixkey random-hex-string-of-algorithm-specified-length
```

也可使用此语法来替换已刷新的 SA。

protocol
指定 `esp` 或 `ah`。

random-hex-string
以十六进制格式指定最多包含八个字符的随机数。字符以 `0x` 开头。如果输入多于安全参数索引 (Security Parameter Index, SPI) 所能接受的数字，系统会忽略多余的数字。如果输入少于 SPI 可接受的数字，系统对您的输入进行填充。

addr
指定一个系统的 IP 地址。

addr2
指定 *addr* 的同级系统的 IP 地址。

protocol-prefix
指定 `encr` 或 `auth` 中的一个。`encr` 前缀与 `esp` 协议一起使用。`auth` 前缀与 `ah` 协议一起使用，用于验证 `esp` 协议。

protocol-algorithm
为 ESP 或 AH 指定算法。每种算法都需要具有特定长度的密钥。

验证算法包括 MD5 和 SHA1。从 Solaris 10 4/09 发行版开始，支持 SHA256 和 SHA512。加密算法包括 DES、3DES、AES 和 Blowfish。

random-hex-string-of-algorithm-specified-length
指定具有算法所要求长度的随机十六进制数。例如，MD5 算法要求其 128 位密钥使用 32 个字符的字符串。3DES 算法要求其 192 位密钥使用 48 个字符的字符串。

a. 例如，在 `enigma` 系统上，保护外发的包。

使用在步骤 1 中生成的随机数。

对于 Solaris 10 1/06 :

```
> add esp spi 0x8bcd1407 \
src 192.168.116.16 dst 192.168.13.213 \
encr_alg aes \
auth_alg sha1 \
encrkey c0c65b888c2ee301c84245c3da63127e92b2676105d5330e85327c1442f37d49 \
authkey 6fab07fec4f2895445500ed992ab48835b9286ff
>
```

注 – 同级系统必须使用相同的加密材料和相同的 SPI。

b. 在 enigma 系统上，仍使用 ipseckey 命令模式保护传入的包。

键入以下命令来保护包 :

```
> add esp spi 0x122a43e4 \
src 192.168.13.213 dst 192.168.116.16 \
encr_alg aes \
auth_alg sha1 \
encrkey a2ea934cd62ca7fa14907cb2ad189b68e4d18c976c14f22b30829e4b1ea4d2ae \
authkey c80984bc4733cc0b7c228b9b74b988d2b7467745
>
```

注 – 对于每个 SA，密钥和 SPI 可以不同。您**应该**为每个 SA 指定不同的密钥和不同的 SPI。

7 要退出 ipseckey 命令模式，请按 Ctrl-D 组合键或键入 quit。

8 将加密材料添加到 /etc/inet/secret/ipseckeys 文件。

在 Solaris 10 4/09 发行版之前的发行版中，此步骤可确保在重新引导时这些加密材料可用于 IPsec。

`/etc/inet/secret/ipseckeys` 文件中的行与 `ipseckey` 命令行语言完全相同。

a. 例如，enigma 系统上的 /etc/inet/secret/ipseckeys 文件的显示与以下内容类似：

```
# ipseckeys - This file takes the file format documented in
#   ipseckey(1m).
#   Note that naming services might not be available when this file
#   loads, just like ipsecinit.conf.
#
# for outbound packets on enigma
add esp spi 0x8bcd1407 \
   src 192.168.116.16 dst 192.168.13.213  \
   encr_alg aes \
   auth_alg sha1  \
   encrkey  c0c65b888c2ee301c84245c3da63127e92b2676105d5330e85327c1442f37d49 \
   authkey  6fab07fec4f2895445500ed992ab48835b9286ff
#
# for inbound packets
add esp spi 0x122a43e4 \
   src 192.168.13.213 dst 192.168.116.16 \
```

```
   encr_alg aes \
   auth_alg sha1  \
   encrkey a2ea934cd62ca7fa14907cb2ad189b68e4d18c976c14f22b30829e4b1ea4d2ae \
   authkey c80984bc4733cc0b7c228b9b74b988d2b7467745
```

b. 使用只读权限保护该文件。

```
# chmod 400 /etc/inet/secret/ipseckeys
```

9 在 partym 系统上重复该过程。

使用 enigma 上使用的相同加密材料。

两个系统上的加密材料**必须**完全相同。如以下示例所示，只有 ipseckeys 文件中的注释不同。注释不同是因为 dst enigma 在 enigma 系统上为传入，在 partym 系统上为外发。

```
# partym ipseckeys file
#
# for inbound packets
add esp spi 0x8bcd1407 \
   src 192.168.116.16 dst 192.168.13.213  \
   encr_alg aes \
   auth_alg sha1  \
   encrkey  c0c65b888c2ee301c84245c3da63127e92b2676105d5330e85327c1442f37d49 \
   authkey  6fab07fec4f2895445500ed992ab48835b9286ff
#
# for outbound packets
add esp spi 0x122a43e4 \
   src 192.168.13.213 dst 192.168.116.16 \
   encr_alg aes \
   auth_alg sha1  \
   encrkey a2ea934cd62ca7fa14907cb2ad189b68e4d18c976c14f22b30829e4b1ea4d2ae \
   authkey c80984bc4733cc0b7c228b9b74b988d2b7467745
```

10 启用 manual-key 服务。

```
# svcadm enable svc:/network/ipsec/manual-key
```

要在当前发行版中替换密钥，请参见示例 20–4。

示例 20–4 替换 IPsec SA

在此示例中，管理员要配置运行当前 Oracle Solaris 10 发行版的系统。管理员生成新密钥，更改 ipseckeys 文件中的加密信息，然后重新启动服务。

- 首先，管理员通过完成第 451 页中的 “如何在 Oracle Solaris 系统上生成随机数”生成密钥。
- 然后，管理员在 /etc/inet/secret/ipseckeys 文件中使用所生成的密钥。

  管理员使用了相同的算法。因此，管理员只更改 SPI、encrkey 和 authkey 的值：

```
add esp spi 0x8xzy1492 \
   src 192.168.116.16 dst 192.168.13.213  \
   encr_alg aes \
   auth_alg sha1  \
   encrkey 0a1f3886b06ebd7d39f6f89e4c29c93f2741c6fa598a38af969907a29ab1b42a \
```

```
    authkey a7230aabf513f35785da73e33b064608be41f69a
#
# add esp spi 0x177xce34\
    src 192.168.13.213 dst 192.168.116.16 \
    encr_alg aes \
    auth_alg sha1  \
    encrkey 4ef5be40bf93498017b2151d788bb37e372f091add9b11149fba42435fefe328 \
    authkey 0e1875d9ff8e42ab652766a5cad49f38c9152821
```

- 最后，管理员重新启动 manual-key 服务。添加新密钥前，重新启动命令会刷新旧密钥。

```
# svcadm restart manual-key
```

▼ 如何检验包是否受 IPsec 保护

要检验包是否受到保护，请使用 snoop 命令来测试连接。以下前缀可以在 snoop 输出中显示：

- AH: 前缀指明 AH 正在保护头。如果已使用 auth_alg 来保护通信，则会看到 AH:。
- ESP: 前缀指明正在发送加密数据。如果已使用 encr_auth_alg 或 encr_alg 来保护通信，则会看到 ESP:。

开始之前 您必须是超级用户或承担等效角色，才能创建 snoop 输出。必须可以同时访问两个系统才能测试连接。

1 在一个系统（如 partym）上，成为超级用户。

```
% su -
Password:       Type root password
#
```

2 在 partym 系统上，准备从远程系统搜寻包。

在 partym 上的一个终端窗口中，从 enigma 系统搜寻包。

```
# snoop -d hme0 -v enigma
Using device /dev/hme (promiscuous mode)
```

3 从远程系统发送包。

在另一个终端窗口中，远程登录到 enigma 系统。提供您的口令。然后，成为超级用户并将包从 enigma 系统发送到 partym 系统。包应该由 snoop -v enigma 命令捕获。

```
% ssh enigma
Password:       Type your password
% su -
Password:       Type root password
# ping partym
```

4 **检查 snoop 输出。**

在 partym 系统上，您应该看到在初始 IP 头信息之后显示 AH 和 ESP 信息的输出。类似以下内容的 AH 和 ESP 信息表明包正在受到保护：

```
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 51 (AH)
IP:   Header checksum = 4e0e
IP:   Source address = 192.168.116.16, enigma
IP:   Destination address = 192.168.13.213, partym
IP:   No options
IP:
AH:  ----- Authentication Header -----
AH:
AH:  Next header = 50 (ESP)
AH:  AH length = 4 (24 bytes)
AH:  <Reserved field = 0x0>
AH:  SPI = 0xb3a8d714
AH:  Replay = 52
AH:  ICV = c653901433ef5a7d77c76eaa
AH:
ESP:  ----- Encapsulating Security Payload -----
ESP:
ESP:  SPI = 0xd4f40a61
ESP:  Replay = 52
ESP:     ....ENCRYPTED DATA....

ETHER:  ----- Ether Header -----
...
```

▼ 如何配置网络安全角色

如果您要使用基于角色的访问控制 (Role-Based Access Control, RBAC) 来管理系统，请使用此过程提供网络管理角色或网络安全角色。

1 **在本地 prof_attr 数据库中查找网络权限配置文件。**

在当前发行版中，输出会显示类似于以下内容的信息：

```
% cd /etc/security
% grep Network prof_attr
Network IPsec Management:::Manage IPsec and IKE...
Network Link Security:::Manage network link security...
Network Management:::Manage the host and network configuration...
Network Security:::Manage network and host security...
Network Wifi Management:::Manage wifi network configuration...
Network Wifi Security:::Manage wifi network security...
```

如果您运行的是 Solaris 10 4/09 发行版之前的发行版，输出会显示类似于以下内容的信息：

```
% cd /etc/security
% grep Network prof_attr
Network Management:::Manage the host and network configuration
Network Security:::Manage network and host security
System Administrator::: Network Management
```

网络管理配置文件是系统管理员配置文件的补充配置文件。如果您将系统管理员权限配置文件纳入角色，则此角色可以执行网络管理配置文件中的命令。

2 **确定网络管理权限配置文件中具有的命令。**

```
% grep "Network Management" /etc/security/exec_attr
Network Management:solaris:cmd:::/usr/sbin/ifconfig:privs=sys_net_config
...
Network Management:suser:cmd:::/usr/sbin/snoop:uid=0
```

使用特权 (`privs=sys_net_config`) 可以运行 `solaris` 策略命令。以超级用户 (`uid=0`) 身份可以运行 `suser` 策略命令。

3 **确定网络安全角色在站点中的作用范围。**

使用步骤 1 中的权限配置文件定义指导您做出决定。

- **要创建处理所有网络安全的角色，请使用 Network Security（网络安全）权限配置文件。**
- **在当前发行版中，要创建只处理 IPsec 和 IKE 的角色，请使用网络 IPsec 管理权限配置文件。**

4 **创建拥有 Network Management（网络管理）权限配置文件的网络安全角色。**

对于拥有除网络管理配置文件外还拥有网络安全或网络 IPsec 管理权限配置文件的角色，除了可执行其他命令外，还可按相应的特权执行 `ifconfig`、`snoop`、`ipsecconf` 和 `ipseckey` 命令。

要创建该角色、将该角色指定给用户以及使用名称服务注册更改，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

示例 20–5 在角色之间划分网络安全职责

在此示例中，管理员要在两个角色之间划分网络安全职责。其中一个角色负责管理 Wifi 和链路安全，另一个角色负责管理 IPsec 和 IKE。为每个角色指定三个人，一人一班。

管理员创建的角色如下：

- 管理员将第一个角色命名为 LinkWifi。
  - 管理员将 Network Wifi（网络 Wifi）、Network Link Security（网络链路安全）和 Network Management（网络管理）权限配置文件指定给该角色。
  - 然后，管理员将 LinkWifi 角色指定给适当的用户。
- 管理员将第二个角色命名为 IPsec Administrator。
  - 管理员将网络 IPsec 管理和网络管理权限配置文件指定给该角色。
  - 然后，管理员将 IPsec Administrator 角色指定给适当的用户。

▼ 如何管理 IKE 和 IPsec 服务

以下步骤提供了最有可能针对 IPsec、IKE 和手动密钥管理使用 SMF 服务的情况。缺省情况下，policy 和 ipsecalgs 服务处于启用状态，而 ike 和 manual-key 服务处于禁用状态。

1 **要管理 IPsec 策略，请执行以下操作之一：**

- **将新策略添加到 ipsecinit.conf 文件后，刷新 policy 服务。**

```
# svcadm refresh svc:/network/ipsec/policy
```

- **更改服务属性的值后，查看属性值，然后刷新并重新启动 policy 服务。**

```
# svccfg -s policy setprop config/config_file=/etc/inet/MyIpsecinit.conf
# svcprop -p config/config_file policy
/etc/inet/MyIpsecinit.conf
# svcadm refresh svc:/network/ipsec/policy
# svcadm restart svc:/network/ipsec/policy
```

2 **要自动管理密钥，请执行以下操作之一：**

- **将项添加到 /etc/inet/ike/config 文件后，启用 ike 服务。**

```
# svcadm enable svc:/network/ipsec/ike
```

- **更改 /etc/inet/ike/config 文件中的项后，重新启动 ike 服务。**

```
# svcadm restart svc:/network/ipsec/ike
```

- **更改服务属性的值后，查看属性值，然后刷新并重新启动服务。**

```
# svccfg -s ike setprop config/admin_privilege=modkeys
# svcprop -p config/admin_privilege ike
modkeys
# svcadm refresh svc:/network/ipsec/ike
# svcadm restart svc:/network/ipsec/ike
```

- **要停止 ike 服务，请将其禁用。**

```
# svcadm disable svc:/network/ipsec/ike
```

3 **要手动管理密钥，请执行以下操作之一：**

- **将项添加到 /etc/inet/secret/ipseckeys 文件后，启用 manual-key 服务。**

```
# svcadm enable svc:/network/ipsec/manual-key
```

- **更改 ipseckeys 文件后，刷新服务。**

```
# svcadm refresh manual-key
```

- **更改服务属性的值后，查看属性值，然后刷新并重新启动服务。**

```
# svccfg -s manual-key setprop config/config_file=/etc/inet/secret/MyIpseckeyfile
# svcprop -p config/config_file manual-key
/etc/inet/secret/MyIpseckeyfile
# svcadm refresh svc:/network/ipsec/manual-key
# svcadm restart svc:/network/ipsec/manual-key
```

- **要阻止手动密钥管理，请禁用 `manual-key` 服务。**

```
# svcadm disable svc:/network/ipsec/manual-key
```

4 **如果修改 IPsec 协议和算法表，请刷新 `ipsecalgs` 服务。**

```
# svcadm refresh svc:/network/ipsec/ipsecalgs
```

故障排除 使用 `svcs` *service* 命令找到服务的状态。如果服务处于 `maintenance` 模式，请遵循 `svcs -x` *service* 命令输出的调试建议。

使用 IPsec 保护 VPN

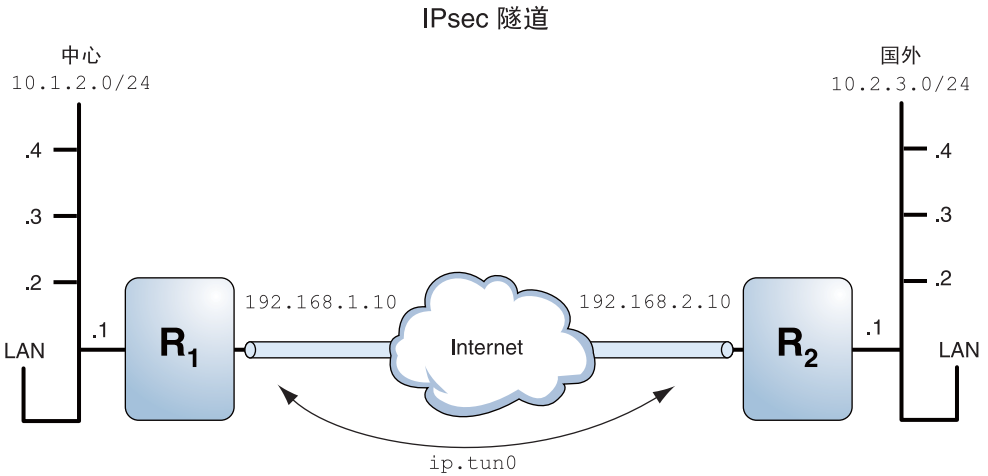
Oracle Solaris 可以配置受 IPsec 保护的 VPN。可在**隧道模式**或**传输模式**下创建隧道。**隧道模式**可与其他供应商的 IPsec 实现交互使用。**传输模式**可与 Solaris OS 的早期版本交互使用。有关隧道模式的讨论，请参见第 436 页中的 “IPsec 中的传输模式和隧道模式”。

处于隧道模式的 IPsec 可提供对通信的更精细控制。在隧道模式中，对于内部 IP 地址，可以为单个端口指定所需的特定保护。

- 有关处于隧道模式的隧道的 IPsec 策略的示例，请参见第 461 页中的 “在隧道模式下使用 IPsec 保护 VPN 的示例”。
- 有关保护 VPN 的过程，请参见第 463 页中的 “使用 IPsec 保护 VPN（任务列表）”。

在隧道模式下使用 IPsec 保护 VPN 的示例

图 20–1 IPsec隧道示意图



以下示例假设这些 LAN 的所有子网都配置了隧道：

```
## Tunnel configuration ##
# Tunnel name is ip.tun0
# Intranet point for the source is 10.1.2.1
# Intranet point for the destination is 10.2.3.1
# Tunnel source is 192.168.1.10
# Tunnel destination is 192.168.2.10
```

示例 20–6 创建一个所有子网都可以使用的隧道

在此示例中，来自图 20–1 中的中心 LAN 的本地 LAN 的所有通信都可以通过隧道从路由器 1 传送到路由器 2，然后再传送到国外 LAN 的所有本地 LAN。通信使用 AES 进行加密。

```
## IPsec policy ##
{tunnel ip.tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

示例 20–7 创建一个仅连接两个子网的隧道

在此示例中，仅为中心 LAN 的子网 10.1.2.0/24 和国外 LAN 的子网 10.2.3.0/24 之间的通信建立了隧道并对通信进行了加密。在中心 LAN 没有其他 IPsec 策略的情况下，如果中心 LAN 尝试通过此隧道路由其他 LAN 的任何通信，则通信会在路由器 1 处被丢弃。

```
## IPsec policy ##
{tunnel ip.tun0 negotiate tunnel laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs aes encr_auth_algs sha1 shared}
```

示例 20–8 仅为两个子网之间的 sendmail 通信创建隧道

在此示例中，仅为 sendmail 通信创建一个隧道。通信从中心 LAN 的子网 10.1.2.0/24 传送到国外 LAN 的子网 10.2.3.0/24 上的电子邮件服务器。电子邮件使用 Blowfish 进行加密。这些策略可应用于远程电子邮件端口和本地电子邮件端口。rport 策略可保护从中心 LAN 发送到国外 LAN 远程电子邮件端口的电子邮件。lport 策略可以保护中心 LAN 在本地端口 25 上接收的来自国外 LAN 的电子邮件。

```
## IPsec policy for email from Central to Overseas ##
{tunnel ip.tun0 negotiate tunnel ulp tcp rport 25
 laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}

## IPsec policy for email from Overseas to Central ##
{tunnel ip.tun0 negotiate tunnel ulp tcp lport 25
 laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}
```

示例 20–9 为所有子网的 FTP 通信创建隧道

在此示例中，IPsec 策略使用 AES 保护图 20–1 中用于从中心 LAN 的所有子网向国外 LAN 的所有子网传输数据的 FTP 端口。此配置适用于 FTP 的主动模式。

```
## IPsec policy for outbound FTP from Central to Overseas ##
{tunnel ip.tun0 negotiate tunnel ulp tcp rport 21}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
{tunnel ip.tun0 negotiate tunnel ulp tcp lport 20}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## IPsec policy for inbound FTP from Central to Overseas ##
{tunnel ip.tun0 negotiate tunnel ulp tcp lport 21}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
{tunnel ip.tun0 negotiate tunnel ulp tcp rport 20}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

使用 IPsec 保护 VPN（任务列表）

以下任务列表列出了配置 IPsec 以保护 Internet 上的通信的相关过程的参考链接。这些过程会在 Internet 上的两个系统之间设置安全的虚拟专用网络 (Virtual Private Network, VPN)。该技术的一种常见用途是通过 Internet 将远程办公室安全连接到公司网络。

任务	说明	参考
保护通过 IPv4 进行的处于隧道模式的隧道通信。	保护两个 Solaris 10 系统之间或者 Solaris 10 系统和 Oracle Solaris 11 系统之间在隧道模式下的通信。Solaris 10 系统必须在 Solaris 10 7/07 发行版或更高版本上运行。 此外，还保护 Solaris 10 系统或 Oracle Solaris 11 系统与在其他平台上运行的系统之间在隧道模式下的通信。Solaris 10 系统必须在 Solaris 10 7/07 发行版或更高版本上运行。	第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”
保护通过 Ipv6 进行的处于隧道模式的隧道通信。	保护两个使用 IPv6 协议的 Oracle Solaris 系统之间在隧道模式下的通信。	第 474 页中的 “如何使用 IPv6 在隧道模式下通过 IPsec 隧道保护 VPN”
保护通过 IPv4 进行的处于传输模式的隧道通信。	保护两个 Solaris 10 系统之间或者 Solaris 10 系统和 Oracle Solaris 11 系统之间在传输模式下的通信。Solaris 10 系统必须在 Solaris 10 7/07 发行版或更高版本上运行。 此外，还保护运行 Solaris OS 早期版本的系统与 Solaris 10 或 Oracle Solaris 系统之间在传输模式下的通信。Solaris 10 系统必须在 Solaris 10 7/07 发行版或更高版本上运行。	第 480 页中的 “如何使用 IPv4 在传输模式下通过 IPsec 隧道保护 VPN”
	通过使用旧的、过时的语法来保护通信。在与运行 Solaris OS 的早期版本的系统进行通信时，此方法很有用。此方法简化了两个系统上的配置文件之间的对比。	示例 20–11 示例 20–16
保护通过 Ipv6 进行的处于传输模式的隧道通信。	保护两个使用 IPv6 协议的 Oracle Solaris 系统之间在传输模式下的通信。	第 486 页中的 “如何使用 IPv6 在传输模式下通过 IPsec 隧道保护 VPN”
防止 IP 电子欺骗。	创建 SMF 服务，以防止系统在没有对包进行解密的情况下通过 VPN 转发包。	第 492 页中的 “如何防止 IP 电子欺骗”

用于保护 VPN 的 IPsec 任务的网络拓扑说明

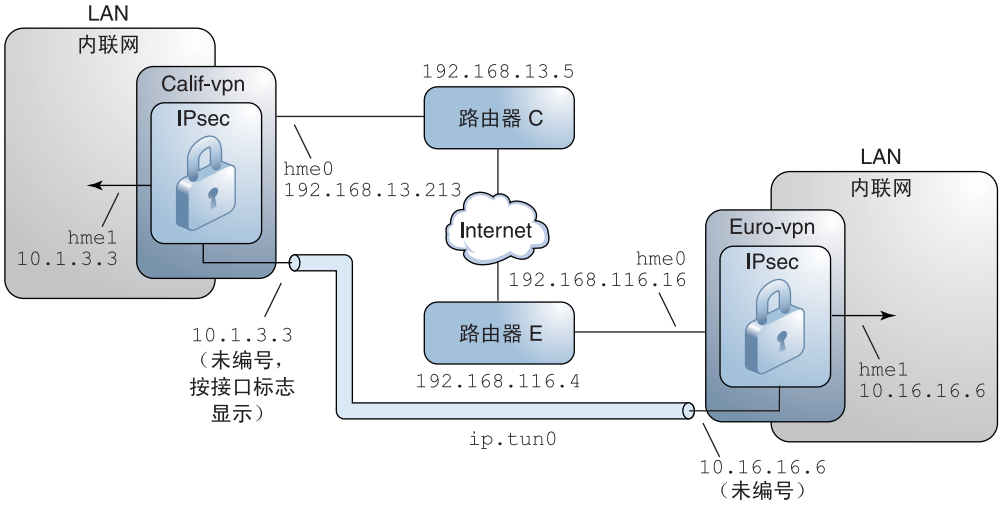
本节中的过程假设您已经进行了以下设置。有关此网络的描述，请参见图 20–2。

- 每个系统都使用 IPv4 地址空间。

  有关使用 IPv6 地址的类似示例，请参见第 474 页中的 “如何使用 IPv6 在隧道模式下通过 IPsec 隧道保护 VPN”。
- 每个系统都有两个接口。`hme0` 接口连接到 Internet。在此示例中，Internet IP 地址以 `192.168` 开始。`hme1` 接口连接到公司的 LAN（即公司的内联网）。在此示例中，内联网 IP 地址以数字 `10` 开始。
- 每个系统都需要采用 SHA–1 算法的 ESP 验证。SHA–1 算法需要 160 位的密钥。

- 每个系统都需要采用 AES 算法的 ESP 加密。AES 算法使用 128 位或 256 位的密钥。
- 每个系统都可以连接到能直接访问 Internet 的路由器。
- 每个系统都使用共享安全关联。

图 20–2 由 Internet 分隔的办公室之间的 VPN 样例



如前面的说明所示，IPv4 网络过程使用以下配置参数。

参数	欧洲	加利福尼亚
系统名称	`enigma`	`partym`
系统内联网接口	`hme1`	`hme1`
系统内联网地址，也是步骤 7 中的 *-point* 地址	`10.16.16.6`	`10.1.3.3`
系统 Internet 接口	`hme0`	`hme0`
系统 Internet 地址，也是步骤 7 中的 *tsrc* 地址	`192.168.116.16`	`192.168.13.213`
Internet 路由器名称	`router-E`	`router-C`
Internet 路由器地址	`192.168.116.4`	`192.168.13.5`
隧道名称	`ip.tun0`	`ip.tun0`

下面的 IPv6 地址在过程中使用。隧道名称相同。

参数	欧洲	加利福尼亚
系统内联网地址	6000:6666::aaaa:1116	6000:3333::eeee:1113
系统 Internet 地址	2001::aaaa:6666:6666	2001::eeee:3333:3333
Internet 路由器地址	2001::aaaa:0:4	2001::eeee:0:1

▼ 如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN

在隧道模式下，内部 IP 包决定保护其内容的 IPsec 策略。

此过程扩展了第 445 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”过程。第 464 页中的 “用于保护 VPN 的 IPsec 任务的网络拓扑说明”介绍了具体设置。

注 – 在两个系统中执行此过程中的步骤。

除了连接两个系统之外，还要连接两个连接到这两个系统的内联网。此过程中的系统作为网关使用。

开始之前 必须位于全局区域中才能为系统或共享 IP 区域配置 IPsec 策略。对于专用 IP 区域，请在非全局区域中配置 IPsec 策略。

1. **在系统控制台上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

   注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

2. **在配置 IPsec 之前控制包流。**

   a. **确保 IP 转发和 IP 动态路由功能均处于禁用状态。**

```
 # routeadm
Configuration   Current             Current
       Option   Configuration       System State
---------------------------------------------------------------
IPv4 routing      default (enabled)    enabled
IPv4 forwarding   disabled             disabled
...
```

如果已启用 IP 转发和 IP 动态路由功能，则禁用它们。

```
# routeadm -d ipv4-routing -d ipv4-forwarding
# routeadm -u
```

关闭 IP 转发功能可阻止包通过此系统从一个网络转发到另一个网络。有关 `routeadm` 命令的说明，请参见 `routeadm(1M)` 手册页。

b. 打开 IP 严格目标多宿主。

```
# ndd -set /dev/ip ip_strict_dst_multihoming 1
```

要打开 IP 严格目标多宿主功能，要求发往系统目标地址之一的包到达正确的目标地址。

启用严格目标多宿主时，到达特定接口的包必须传送到此接口的本地 IP 地址之一。所有其他包，甚至是传送到系统其他本地地址的包，均被丢弃。

注意 – 系统启动时多宿主值会恢复为缺省值。要使更改的值具有持久性，请参见第 492 页中的“如何防止 IP 电子欺骗”。

---

c. 禁用多数网络服务，也可能会禁用所有网络服务。

注 – 如果系统中安装了“受限制的”SMF 配置文件，则可以跳过此步骤。将禁用网络服务（Oracle Solaris 的安全 Shell 功能除外）。

禁用网络服务可防止 IP 包危害系统。例如，可以采用 SNMP 守护进程、`telnet` 连接或 `rlogin` 连接。

选择以下选项之一：

- 如果运行的是 Solaris 10 11/06 发行版或更高版本，请运行“受限制的”SMF 配置文件。

  ```
  # netservices limited
  ```

- 否则，单独禁用网络服务。

  ```
  # svcadm disable network/ftp:default
  # svcadm disable network/finger:default
  # svcadm disable network/login:rlogin
  # svcadm disable network/nfs/server:default
  # svcadm disable network/rpc/rstat:default
  # svcadm disable network/smtp:sendmail
  # svcadm disable network/telnet:default
  ```

d. 检验是否已禁用大多数网络服务。

检验回送挂载和 `ssh` 服务是否正在运行。

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
```

```
...
online         Aug_09   svc:/network/ssh:default
```

3 **在两个系统间添加一对 SA。**

选择以下选项之一：

- 将 IKE 配置为管理 SA 的密钥。请使用第 509 页中的 “配置 IKE（任务列表）”中的过程之一来为 VPN 配置 IKE。
- 如果您有充分的理由来手动管理密钥，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。

4 **添加 IPsec 策略。**

编辑 `/etc/inet/ipsecinit.conf` 文件来为 VPN 添加 IPsec 策略。要增强策略，请参见示例 20–12。有关其他示例，请参见第 461 页中的 “在隧道模式下使用 IPsec 保护 VPN 的示例”。

在此策略中，本地 LAN 上的系统与网关的内部 IP 地址之间不需要 IPsec 保护，因此将添加 `bypass` 语句。

a. **在 `enigma` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：**

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.16.16.6 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. **在 `partym` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：**

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

5 **可选检验 IPsec 策略文件的语法。**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 **要配置隧道并使用 IPsec 对其加以保护，请根据 Oracle Solaris 发行版执行相应步骤：**

- **从 Solaris 10 4/09 发行版开始，请执行步骤 7 至步骤 13，然后运行步骤 22 中的路由协议。**
- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请执行步骤 14 至步骤 22。**

7 **在 /etc/hostname.ip.tun0 文件中配置隧道 ip.tun0。**

此文件的语法如下所示：

system1-point system2-point `tsrc` *system1-taddr* `tdst` *system2-taddr* `router up`

a. **在 enigma 系统上，向 hostname.ip.tun0 文件添加以下项：**

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 tdst 192.168.13.213 router up
```

b. **在 partym 系统上，向 hostname.ip.tun0 文件添加以下项：**

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 tdst 192.168.116.16 router up
```

8 **使用创建的 IPsec 策略保护隧道。**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 **要将隧道配置文件的内容读入内核，请重新启动网络服务。**

```
# svcadm restart svc:/network/initial:default
```

10 **为 hme1 接口打开 IP 转发功能。**

a. **在 enigma 系统上，将路由器项添加到 /etc/hostname.hme1 文件。**

```
192.168.116.16 router
```

b. **在 partym 系统上，将路由器项添加到 /etc/hostname.hme1 文件。**

```
192.168.13.213 router
```

IP 转发指可以转发来自其他位置的包。IP 转发也指由此接口发出的包可能源于其他位置。要成功转发包，必须打开接收接口和传送接口的 IP 转发功能。

因为 hme1 接口在内联网**内部**，所以必须打开 hme1 的 IP 转发功能。因为 ip.tun0 通过 Internet 连接两个系统，所以必须打开 ip.tun0 的 IP 转发功能。

hme0 接口已关闭其 IP 转发功能以阻止**外部**入侵者向受保护的内联网中注入包。**外部**是指 Internet。

11 **确保路由协议不在内联网内通告缺省的路由。**

a. **在 enigma 系统上，将 private 标志添加到 /etc/hostname.hme0 文件。**

```
10.16.16.6 private
```

b. **在 partym 系统上，将 private 标志添加到 /etc/hostname.hme0 文件。**

```
10.1.3.3 private
```

即使 hme0 关闭 IP 转发功能，路由协议实现仍会通告接口。例如，in.routed 协议仍会通告 hme0 可将包转发到内联网中的对等接口。可以通过设置接口的**专用**标志，阻止这些通告。

12 **手动添加通过 hme0 接口实现的缺省路由。**

缺省路由必须是可以直接访问 Internet 的路由器。

a. **在 enigma 系统上，添加以下路由：**

```
# route add default 192.168.116.4
```

b. **在 partym 系统上，添加以下路由：**

```
# route add default 192.168.13.5
```

即使 hme0 接口不是内联网的一部分，hme0 也需要通过 Internet 访问其同级系统。要找到其同级系统，hme0 需要有关 Internet 路由的信息。对于 Internet 的其他部分来说，VPN 系统像是一台主机，而不是路由器。因此，您可以使用缺省的路由器或运行路由器搜索协议来查找同级系统。有关更多信息，请参见 route(1M) 和 in.routed(1M) 手册页。

13 **要完成该过程，请转至步骤 22 运行路由协议。**

14 **配置隧道 ip.tun0。**

注 – 以下步骤用于在运行 Solaris 10 4/09 发行版之前的发行版的系统中配置隧道。

使用 ifconfig 命令创建点对点接口：

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 system1-point system2-point \
tsrc system1-taddr tdst system2-taddr
```

a. **在 enigma 系统上键入以下命令：**

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213
```

b. **在 partym 系统上键入以下命令：**

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.1.3.3 10.16.16.6  \
tsrc 192.168.13.213 tdst 192.168.116.16
```

15 **使用创建的 IPsec 策略保护隧道。**

```
# ipsecconf
```

16 **初启隧道的路由器。**

```
# ifconfig ip.tun0 router up
```

17 为 hme1 接口打开 IP 转发功能。

```
# ifconfig hme1 router
```

IP 转发指可以转发来自其他位置的包。IP 转发也指由此接口发出的包可能源于其他位置。要成功转发包，必须打开接收接口和传送接口的 IP 转发功能。

因为 hme1 接口在内联网**内部**，所以必须打开 hme1 的 IP 转发功能。因为 ip.tun0 通过 Internet 连接两个系统，所以必须打开 ip.tun0 的 IP 转发功能。

hme0 接口已关闭其 IP 转发功能以阻止**外部**入侵者向受保护的内联网中注入包。**外部**是指 Internet。

18 确保路由协议不在内联网内通告缺省的路由。

```
# ifconfig hme0 private
```

即使 hme0 关闭 IP 转发功能，路由协议实现仍会通告接口。例如，in.routed 协议仍会通告 hme0 可将包转发到内联网中的对等接口。可以通过设置接口的**专用**标志，阻止这些通告。

19 手动添加通过 hme0 实现的缺省路由。

缺省路由必须是可以直接访问 Internet 的路由器。

a. 在 enigma 系统上，添加以下路由：

```
# route add default 192.168.116.4
```

b. 在 partym 系统上，添加以下路由：

```
# route add default 192.168.13.5
```

即使 hme0 接口不是内联网的一部分，hme0 也需要通过 Internet 访问其同级系统。要找到其同级系统，hme0 需要有关 Internet 路由的信息。对于 Internet 的其他部分来说，VPN 系统像是一台主机，而不是路由器。因此，您可以使用缺省的路由器或运行路由器搜索协议来查找同级系统。有关更多信息，请参见 route(1M) 和 in.routed(1M) 手册页。

20 通过向 /etc/hostname.ip.tun0 文件中添加项来确保 VPN 在系统重新引导后启动。

system1-point system2-point tsrc *system1-taddr* tdst *system2-taddr* router up

a. 在 enigma 系统上，向 hostname.ip.tun0 文件添加以下项：

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 tdst 192.168.13.213 router up
```

b. 在 partym 系统上，向 hostname.ip.tun0 文件添加以下项：

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 tdst 192.168.116.16 router up
```

21 **将接口文件配置为将正确的参数传送到路由选择守护进程。**

a. **在 enigma 系统上，修改 /etc/hostname.*interface* 文件。**

```
# cat /etc/hostname.hme0
## enigma
10.16.16.6 private

# cat /etc/hostname.hme1
## enigma
192.168.116.16 router
```

b. **在 partym 系统上，修改 /etc/hostname.*interface* 文件。**

```
# cat /etc/hostname.hme0
## partym
10.1.3.3 private

# cat /etc/hostname.hme1
## partym
192.168.13.213 router
```

22 **运行路由协议。**

```
# routeadm -e ipv4-routing
# routeadm -u
```

您可能需要在运行路由协议之前先配置路由协议。有关更多信息，请参见第 222 页中的“Oracle Solaris 中的路由协议”。有关过程，请参见第 106 页中的“如何配置 IPv4 路由器”。

示例 20–10 测试时创建临时隧道

在此示例中，管理员在 Solaris 10 4/09 系统中测试隧道的创建情况。稍后，管理员将使用第 466 页中的“如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”过程使隧道成为永久隧道。在测试过程中，管理员在系统 system1 和 system2 上执行以下一系列步骤：

- 在两个系统上，管理员完成第 466 页中的“如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”的前五步。
- 管理员使用 ifconfig 命令检测并配置临时隧道。

```
system1 # ifconfig ip.tun0 plumb
system1 # ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
          tsrc 192.168.116.16 tdst 192.168.13.213

# ssh system2
Password:      admin-password-on-system2
system2 # ifconfig ip.tun0 plumb
system2 # ifconfig ip.tun0  10.1.3.3 10.16.16.6 \
          tsrc 192.168.13.213 tdst 192.168.116.16
```

- 管理员对隧道启用 IPsec 策略。该策略是在第 466 页中的“如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”的步骤 4 中创建的。

```
system1 # svcadm refresh svc:/network/ipsec/policy:default
system2 # svcadm refresh svc:/network/ipsec/policy:default
```

- 管理员使 Internet 接口成为路由器，并防止路由协议通过内联网接口传送。

```
system1 # ifconfig hme1 router ; ifconfig hme0 private

system2 # ifconfig hme1 router ; ifconfig hme0 private
```

- 管理员在两个系统上通过完成第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”的步骤 12 和步骤 22，手动添加路由并运行路由协议。

示例 20–11 使用命令行为早期版本的 Solaris 系统创建通道

在 Solaris 10 7/07 发行版中，简化了 `ifconfig` 命令的语法。在此示例中，管理员将针对运行 Solaris 10 7/07 发行版之前的 Solaris 版本的系统测试隧道创建情况。通过使用 `ifconfig` 命令的原始语法，管理员可以在两个通信系统中使用完全相同的命令。稍后，管理员将使用第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”使隧道成为永久隧道。

在测试过程中，管理员在系统 `system1` 和 `system2` 上执行以下步骤：

- 在两个系统上，管理员完成第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”的前五步。
- 管理员检测并配置隧道。

```
system1 # ifconfig ip.tun0 plumb
system1 # ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
          tsrc 192.168.116.16 tdst 192.168.13.213 \
          encr_algs aes encr_auth_algs sha1
system1 # ifconfig ip.tun0 router up

# ssh system2
Password:      admin-password-on-system2
system2 # ifconfig ip.tun0 plumb
system2 # ifconfig ip.tun0 10.1.3.3  10.16.16.6 \
          tsrc 192.168.13.213 tdst 192.168.116.16 \
          encr_algs aes encr_auth_algs sha1
system2 # ifconfig ip.tun0 router up
```

- 管理员对隧道启用 IPsec 策略。该策略是在第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”的步骤 4 中创建的。

```
system1 # svcadm refresh svc:/network/ipsec/policy:default
system2 # svcadm refresh svc:/network/ipsec/policy:default
```

- 管理员使 Internet 接口成为路由器，并防止路由协议通过内联网接口传送。

```
system1 # ifconfig hme1 router ; ifconfig hme0 private
system2 # ifconfig hme1 router ; ifconfig hme0 private
```

- 管理员在两个系统上通过完成第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”的步骤 12 和步骤 22 来添加路由。

示例 20–12 在 LAN 中所有系统中要求 IPsec 策略

在此示例中，管理员注释掉了在步骤 4 中配置的 bypass 策略，从而加强了保护。通过此策略配置，LAN 中的每个系统都必须激活 IPsec 以便与路由器通信。

```
# LAN traffic must implement IPsec.
# {laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate tunnel} ipsec {encr_algs aes encr_auth_algs sha1}
```

示例 20–13 使用 IPsec 保护 Telnet 通信以及使用 IPsec 保护 SMTP 通信（二者不同）

在此示例中，第一条规则使用 Blowfish 和 SHA-1 保护端口 23 上的 telnet 通信。第二条规则使用 AES 和 MD5 保护端口 25 上的 SMTP 通信。

```
{laddr 10.1.3.3 ulp tcp dport 23 dir both}
  ipsec {encr_algs blowfish encr_auth_algs sha1 sa unique}
{laddr 10.1.3.3 ulp tcp dport 25 dir both}
 ipsec {encr_algs aes encr_auth_algs md5 sa unique}
```

示例 20–14 使用处于隧道模式的 IPsec 隧道保护子网通信以及使用处于隧道模式的 IPsec 隧道保护其他网络通信（二者不同）

以下隧道配置保护子网 10.1.3.0/24 在隧道上的所有通信：

```
{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

以下隧道配置保护子网 10.1.3.0/24 在隧道上与不同子网的通信。以 10.2.*x.x* 开头的子网要经由隧道。

```
{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24 raddr 10.2.1.0/24}
  ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}

{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24 raddr 10.2.2.0/24}
  ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}

{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24 raddr 10.2.3.0/24}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

▼ 如何使用 IPv6 在隧道模式下通过 IPsec 隧道保护 VPN

要在 IPv6 网络中设置 VPN，您需要执行与针对 IPv4 网络执行的步骤相同的步骤。但是，命令的语法稍有不同。有关运行特定命令的更详尽的原因说明，请参见第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”中的相应步骤。

注 – 在两个系统中执行此过程中的步骤。

此过程使用以下配置参数。

参数	欧洲	加利福尼亚
系统名称	`enigma`	`partym`
系统内联网接口	`hme1`	`hme1`
系统 Internet 接口	`hme0`	`hme0`
系统内联网地址	`6000:6666::aaaa:1116`	`6000:3333::eeee:1113`
系统 Internet 地址	`2001::aaaa:6666:6666`	`2001::eeee:3333:3333`
Internet 路由器名称	`router-E`	`router-C`
Internet 路由器地址	`2001::aaaa:0:4`	`2001::eeee:0:1`
隧道名称	`ip6.tun0`	`ip6.tun0`

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2 在配置 IPsec 之前控制包流。

有关这些命令的作用，请参见步骤 2 in 第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”。

a. 确保 IP 转发和 IP 动态路由功能均处于禁用状态。

```
# routeadm
Configuration       Current         Current
       Option       Configuration  System State
---------------------------------------------------
...
IPv6 forwarding     disabled          disabled
   IPv6 routing     disabled          disabled
```

如果已启用 IP 转发和 IP 动态路由功能，您可以通过键入以下内容来禁用它们：

```
# routeadm -d ipv6-forwarding -d ipv6-routing
# routeadm -u
```

b. 打开 IP 严格目标多宿主。

```
# ndd -set /dev/ip ip6_strict_dst_multihoming 1
```

注意 – 系统引导时，会将 `ip6_strict_dst_multihoming` 的值恢复为缺省值。要使更改的值具有持久性，请参见第 492 页中的 “如何防止 IP 电子欺骗”。

c. 禁用多数网络服务，也可能会禁用所有网络服务。

注 – 如果系统中安装了“受限制的”SMF 配置文件，则可以跳过此步骤。将禁用网络服务（安全 Shell 除外）。

禁用网络服务可防止 IP 包危害系统。例如，可以采用 SNMP 守护进程、`telnet` 连接或 `rlogin` 连接。

选择以下选项之一：

- 如果运行的是 Solaris 10 11/06 发行版或更高版本，请运行“受限制的”SMF 配置文件。

  ```
  # netservices limited
  ```

- 否则，单独禁用网络服务。

  ```
  # svcadm disable network/ftp:default
  # svcadm disable network/finger:default
  # svcadm disable network/login:rlogin
  # svcadm disable network/nfs/server:default
  # svcadm disable network/rpc/rstat:default
  # svcadm disable network/smtp:sendmail
  # svcadm disable network/telnet:default
  ```

d. 检验是否已禁用大多数网络服务。

检验回送挂载和 `ssh` 服务是否正在运行。

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

3 在两个系统间添加一对 SA。

选择以下选项之一：

- 将 IKE 配置为管理 SA 的密钥。请使用第 509 页中的 “配置 IKE（任务列表）”中的过程之一来为 VPN 配置 IKE。
- 如果您有充分的理由来手动管理密钥，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。

4 **为 VPN 添加 IPsec 策略。**

编辑 `/etc/inet/ipsecinit.conf` 文件来为 VPN 添加 IPsec 策略。

a. **在 `enigma` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：**

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic to and from this host can bypass IPsec.
{laddr 6000:6666::aaaa:1116 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate tunnel}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. **在 `partym` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：**

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic to and from this host can bypass IPsec.
{laddr 6000:3333::eeee:1113 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate tunnel}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

5 **可选检验 IPsec 策略文件的语法。**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 **要配置隧道并使用 IPsec 对其加以保护，请根据 Oracle Solaris 发行版执行相应步骤：**

- **从 Solaris 10 4/09 发行版开始，请执行步骤 7 至步骤 13，然后运行步骤 22 中的路由协议。**
- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请执行步骤 14 至步骤 22。**

7 **在 `/etc/hostname.ip6.tun0` 文件中配置隧道 `ip6.tun0`。**

a. **在 `enigma` 系统上，向 `hostname.ip6.tun0` 文件添加以下项：**

```
6000:6666::aaaa:1116 6000:3333::eeee:1113 tsrc 2001::aaaa:6666:6666 tdst 2001::eeee:3333:3333 router up
```

b. **在 `partym` 系统上，向 `hostname.ip6.tun0` 文件添加以下项：**

```
6000:3333::eeee:1113  6000:6666::aaaa:1116 tsrc 2001::eeee:3333:3333 tdst 2001::aaaa:6666:6666 router up
```

8 使用创建的 IPsec 策略保护隧道。

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 要将隧道配置文件的内容读入内核，请重新启动网络服务。

```
# svcadm restart svc:/network/initial:default
```

10 为 hme1 接口打开 IP 转发功能。

a. 在 enigma 系统上，将路由器项添加到 /etc/hostname6.hme1 文件。

```
2001::aaaa:6666:6666 inet6 router
```

b. 在 partym 系统上，将路由器项添加到 /etc/hostname6.hme1 文件。

```
2001::eeee:3333:3333 inet6 router
```

11 确保路由协议不在内联网内通告缺省的路由。

a. 在 enigma 系统上，将 private 标志添加到 /etc/hostname6.hme0 文件。

```
6000:6666::aaaa:1116 inet6 private
```

b. 在 partym 系统上，将 private 标志添加到 /etc/hostname6.hme0 文件。

```
6000:3333::eeee:1113 inet6 private
```

12 手动添加通过 hme0 实现的缺省路由。

a. 在 enigma 系统上，添加以下路由：

```
# route add -inet6 default 2001::aaaa:0:4
```

b. 在 partym 系统上，添加以下路由：

```
# route add -inet6 default 2001::eeee:0:1
```

13 要完成该过程，请转至步骤 22 运行路由协议。

14 配置安全隧道 ip6.tun0。

注 – 以下步骤用于在运行 Solaris 10 4/09 发行版之前的发行版的系统中配置隧道。

a. 在 enigma 系统上键入以下命令：

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333
```

b. 在 partym 系统上键入以下命令：

```
# ifconfig ip6.tun0 inet6 plumb
```

```
# ifconfig ip6.tun0 inet6 6000:3333::eeee:1113 6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666
```

15 使用创建的 IPsec 策略保护隧道。

```
# ipsecconf
```

16 初启隧道的路由器。

```
# ifconfig ip6.tun0 router up
```

17 在每个系统上，为 hme1 接口打开 IP 转发功能。

```
# ifconfig hme1 router
```

18 确保路由协议不在内联网内通告缺省的路由。

```
# ifconfig hme0 private
```

19 手动添加通过 hme0 实现的缺省路由。

缺省路由必须是可以直接访问 Internet 的路由器。

a. 在 enigma 系统上，添加以下路由：

```
# route add -inet6 default 2001::aaaa:0:4
```

b. 在 partym 系统上，添加以下路由：

```
# route add -inet6 default 2001::eeee:0:1
```

20 通过向 /etc/hostname6.ip6.tun0 文件中添加项来确保 VPN 在系统重新引导后启动。

该项复制在步骤 14 中传递到 ifconfig 命令的参数。

a. 在 enigma 系统上，向 hostname6.ip6.tun0 文件添加以下项：

```
6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666  tdst 2001::eeee:3333:3333 router up
```

b. 在 partym 系统上，向 hostname6.ip6.tun0 文件添加以下项：

```
6000:3333::eeee:1113 6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666  router up
```

21 在每个系统上，将接口文件配置为将正确的参数传送到路由选择守护进程。

a. 在 enigma 系统上，修改 /etc/hostname6. *interface* 文件。

```
# cat /etc/hostname6.hme0
## enigma
6000:6666::aaaa:1116 inet6 private

#  cat /etc/hostname6.hme1
## enigma
2001::aaaa:6666:6666 inet6 router
```

b. 在 partym 系统上，修改 /etc/hostname6.*interface* 文件。

```
# cat /etc/hostname6.hme0
## partym
6000:3333::eeee:1113 inet6 private

# cat /etc/hostname6.hme1
## partym
2001::eeee:3333:3333 inet6 router
```

22 运行路由协议。

```
# routeadm -e ipv6-routing
# routeadm -u
```

您可能需要在运行路由协议之前先配置路由协议。有关更多信息，请参见第 222 页中的“Oracle Solaris 中的路由协议”。有关过程，请参见第 154 页中的“配置 IPv6 路由器”。

▼ 如何使用 IPv4 在传输模式下通过 IPsec 隧道保护 VPN

在传输模式下，外部头决定保护内部 IP 包的 IPsec 策略。

此过程扩展了第 445 页中的“如何使用 IPsec 保证两个系统之间的通信安全”过程。除了连接两个系统之外，还要连接两个连接到这两个系统的内联网。此过程中的系统作为网关使用。

此过程使用第 464 页中的“用于保护 VPN 的 IPsec 任务的网络拓扑说明”中介绍的设置。有关运行特定命令的更详尽的原因说明，请参见第 466 页中的“如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”中的相应步骤。

注 – 在两个系统中执行此过程中的步骤。

---

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

---

2 在配置 IPsec 之前控制包流。

a. 确保 IP 转发和 IP 动态路由功能均处于禁用状态。

```
# routeadm
Configuration       Current         Current
       Option       Configuration  System State
```

```
--------------------------------------------------
IPv4 forwarding     disabled           disabled
   IPv4 routing     default (enabled)   enabled
...
```

如果已启用 IP 转发和 IP 动态路由功能，您可以通过键入以下内容来禁用它们：

```
# routeadm -d ipv4-routing -d ipv4-forwarding
# routeadm -u
```

b. 打开 IP 严格目标多宿主。

```
# ndd -set /dev/ip ip_strict_dst_multihoming 1
```

注意 – 系统引导时，会将 `ip_strict_dst_multihoming` 的值恢复为缺省值。要使更改的值具有持久性，请参见第 492 页中的 “如何防止 IP 电子欺骗”。

c. 禁用多数网络服务，也可能会禁用所有网络服务。

注 – 如果系统中安装了“受限制的”SMF 配置文件，则可以跳过此步骤。将禁用网络服务（安全 Shell 除外）。

禁用网络服务可防止 IP 包危害系统。例如，可以采用 SNMP 守护进程、`telnet` 连接或 `rlogin` 连接。

选择以下选项之一：

- 如果运行的是 Solaris 10 11/06 发行版或更高版本，请运行“受限制的”SMF 配置文件。

  ```
  # netservices limited
  ```

- 否则，单独禁用网络服务。

  ```
  # svcadm disable network/ftp:default
  # svcadm disable network/finger:default
  # svcadm disable network/login:rlogin
  # svcadm disable network/nfs/server:default
  # svcadm disable network/rpc/rstat:default
  # svcadm disable network/smtp:sendmail
  # svcadm disable network/telnet:default
  ```

d. 检验是否已禁用大多数网络服务。

检验回送挂载和 `ssh` 服务是否正在运行。

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

3 在两个系统间添加一对 SA。

选择以下选项之一：

- 将 IKE 配置为管理 SA 的密钥。请使用第 509 页中的 “配置 IKE（任务列表）”中的过程之一来为 VPN 配置 IKE。
- 如果您有充分的理由来手动管理密钥，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。

4 添加 IPsec 策略。

编辑 `/etc/inet/ipsecinit.conf` 文件来为 VPN 添加 IPsec 策略。要增强策略，请参见示例 20–15。

a. 在 `enigma` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.16.16.6 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. 在 `partym` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

5 可选检验 IPsec 策略文件的语法。

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 要配置隧道并使用 IPsec 对其加以保护，请根据 Oracle Solaris 发行版执行相应步骤：

- **从 Solaris 10 4/09 发行版开始，请执行步骤 7 至步骤 13，然后运行步骤 22 中的路由协议。**
- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请执行步骤 14 至步骤 22。**

7 在 `/etc/hostname.ip.tun0`文件中配置隧道 `ip.tun0`。

a. 在 `enigma` 系统上，向 `hostname.ip.tun0` 文件添加以下项：

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 tdst 192.168.13.213 router up
```

b. 在 `partym` 系统上，向 `hostname.ip.tun0` 文件添加以下项：

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 tdst 192.168.116.16 router up
```

8 使用创建的 IPsec 策略保护隧道。

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 要将 **hostname.ip.tun0** 文件的内容读入内核，请重新启动网络服务。

```
# svcadm restart svc:/network/initial:default
```

10 为 **hme1** 接口打开 IP 转发功能。

a. 在 **enigma** 系统上，将路由器项添加到 **/etc/hostname.hme1** 文件。

```
192.168.116.16 router
```

b. 在 **partym** 系统上，将路由器项添加到 **/etc/hostname.hme1** 文件。

```
192.168.13.213 router
```

11 确保路由协议不在内联网内通告缺省的路由。

a. 在 **enigma** 系统上，将 **private** 标志添加到 **/etc/hostname.hme0** 文件。

```
10.16.16.6 private
```

b. 在 **partym** 系统上，将 **private** 标志添加到 **/etc/hostname.hme0** 文件。

```
10.1.3.3 private
```

12 手动添加通过 **hme0** 实现的缺省路由。

a. 在 **enigma** 系统上，添加以下路由：

```
# route add default 192.168.116.4
```

b. 在 **partym** 系统上，添加以下路由：

```
# route add default 192.168.13.5
```

13 要完成该过程，请转至步骤 22 运行路由协议。

14 配置隧道 **ip.tun0**。

注 – 以下步骤用于在运行 Solaris 10 4/09 发行版之前的发行版的系统中配置隧道。

使用 `ifconfig` 命令创建点对点接口：

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 system1-point system2-point \
tsrc system1-taddr tdst system2-taddr
```

a. 在 **enigma** 系统上键入以下命令：

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213
```

b. 在 **partym** 系统上键入以下命令：

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.1.3.3 10.16.16.6  \
tsrc 192.168.13.213 tdst 192.168.116.16
```

15 使用创建的 IPsec 策略保护隧道。

```
# ipsecconf
```

16 初启隧道的路由器。

```
# ifconfig ip.tun0 router up
```

17 为 **hme1** 接口打开 IP 转发功能。

```
# ifconfig hme1 router
```

18 确保路由协议不在内联网内通告缺省的路由。

```
# ifconfig hme0 private
```

19 手动添加通过 **hme0** 实现的缺省路由。

缺省路由必须是可以直接访问 Internet 的路由器。

```
# route add default router-on-hme0-subnet
```

a. 在 **enigma** 系统上，添加以下路由：

```
# route add default 192.168.116.4
```

b. 在 **partym** 系统上，添加以下路由：

```
# route add default 192.168.13.5
```

20 通过向 **/etc/hostname.ip.tun0** 文件中添加项来确保 VPN 在系统重新引导后启动。

```
system1-point system2-point tsrc system1-taddr \
tdst system2-taddr encr_algs aes encr_auth_algs sha1 router up
```

a. 在 **enigma** 系统上，向 **hostname.ip.tun0** 文件添加以下项：

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 \
tdst 192.168.13.213 router up
```

b. 在 **partym** 系统上，向 **hostname.ip.tun0** 文件添加以下项：

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 \
tdst 192.168.116.16 router up
```

21 将接口文件配置为将正确的参数传送到路由选择守护进程。

a. 在 **enigma** 系统上，修改 **/etc/hostname.***interface* 文件。

```
# cat /etc/hostname.hme0
## enigma
10.16.16.6 private

# cat /etc/hostname.hme1
## enigma
192.168.116.16 router
```

b. 在 **partym** 系统上，修改 **/etc/hostname.***interface* 文件。

```
# cat /etc/hostname.hme0
## partym
10.1.3.3 private

# cat /etc/hostname.hme1
## partym
192.168.13.213 router
```

22 运行路由协议。

```
# routeadm -e ipv4-routing
# routeadm -u
```

示例 20–15 在所有处于传输模式的系统上要求 IPsec 策略

在此示例中，管理员注释掉了在步骤 4 中配置的 bypass 策略，从而加强了保护。通过此策略配置，LAN 中的每个系统都必须激活 IPsec 以便与路由器通信。

```
# LAN traffic must implement IPsec.
# {laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate transport} ipsec {encr_algs aes encr_auth_algs sha1}
```

示例 20–16 使用过时的语法配置处于传输模式的 IPsec 隧道

在此示例中，管理员将 Solaris 10 7/07 系统与运行 Oracle Solaris 10 发行版的系统连接在一起。因此，管理员在配置文件中使用 Solaris 10 语法，并在 `ifconfig` 命令中包含 IPsec 算法。

管理员按照第 480 页中的“如何使用 IPv4 在传输模式下通过 IPsec 隧道保护 VPN”过程进行操作，但在语法上进行了以下更改。

- 对于步骤 4，`ipsecinit.conf` 文件的语法如下所示：

```
# LAN traffic to and from this address can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{} ipsec {encr_algs aes encr_auth_algs sha1}
```

- 对于步骤 14 至步骤 16，配置安全隧道的语法如下所示：

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213 \
encr_algs aes encr_auth_algs sha1

# ifconfig ip.tun0 router up

# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213 \
encr_algs aes encr_auth_algs sha1
```

  传送到 `ifconfig` 命令的 IPsec 策略必须与 `ipsecinit.conf` 文件中的 IPsec 策略相同。在重新引导时，每个系统都会读取 `ipsecinit.conf` 文件来获取其策略。

- 对于步骤 20，`hostname.ip.tun0` 文件的语法如下所示：

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 \
tdst 192.168.13.213 encr_algs aes encr_auth_algs sha1 router up
```

▼ 如何使用 IPv6 在传输模式下通过 IPsec 隧道保护 VPN

要在 IPv6 网络中设置 VPN，您需要执行与针对 IPv4 网络执行的步骤相同的步骤。但是，命令的语法稍有不同。有关运行特定命令的更详尽的原因说明，请参见第 466 页中的 “如何使用 IPv4 在隧道模式下通过 IPsec 隧道保护 VPN”中的相应步骤。

注 – 在两个系统中执行此过程中的步骤。

此过程使用以下配置参数。

参数	欧洲	加利福尼亚
系统名称	`enigma`	`partym`
系统内联网接口	`hme1`	`hme1`
系统 Internet 接口	`hme0`	`hme0`
系统内联网地址	`6000:6666::aaaa:1116`	`6000:3333::eeee:1113`
系统 Internet 地址	`2001::aaaa:6666:6666`	`2001::eeee:3333:3333`
Internet 路由器名称	`router-E`	`router-C`
Internet 路由器地址	`2001::aaaa:0:4`	`2001::eeee:0:1`

参数	欧洲	加利福尼亚
隧道名称	ip6.tun0	ip6.tun0

1 **在系统控制台上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

2 **在配置 IPsec 之前控制包流。**

a. **确保 IP 转发和 IP 动态路由功能均处于禁用状态。**

```
# routeadm
Configuration       Current         Current
       Option       Configuration  System State
-------------------------------------------------
...
IPv6 forwarding     disabled          disabled
   IPv6 routing     disabled          disabled
```

如果已启用 IP 转发和 IP 动态路由功能，您可以通过键入以下内容来禁用它们：

```
# routeadm -d ipv6-forwarding -d ipv6-routing
# routeadm -u
```

b. **打开 IP 严格目标多宿主。**

```
# ndd -set /dev/ip ip6_strict_dst_multihoming 1
```

注意 – 系统引导时，会将 ip6_strict_dst_multihoming 的值恢复为缺省值。要使更改的值具有持久性，请参见第 492 页中的 “如何防止 IP 电子欺骗”。

c. **检验是否已禁用大多数网络服务。**

检验回送挂载和 ssh 服务是否正在运行。

```
# svcs | grep network
online          Aug_02   svc:/network/loopback:default
...
online          Aug_09   svc:/network/ssh:default
```

3 **在两个系统间添加一对 SA。**

选择以下选项之一：

- 将 IKE 配置为管理 SA 的密钥。请使用第 509 页中的 “配置 IKE（任务列表）”中的过程之一来为 VPN 配置 IKE。
- 如果您有充分的理由来手动管理密钥，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。

4 添加 IPsec 策略。

编辑 `/etc/inet/ipsecinit.conf` 文件来为 VPN 添加 IPsec 策略。

a. 在 `enigma` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic can bypass IPsec.
{laddr 6000:6666::aaaa:1116 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1}
```

b. 在 `partym` 系统上，将以下项键入到 `ipsecinit.conf` 文件中：

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic can bypass IPsec.
{laddr 6000:3333::eeee:1113 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1}
```

5 可选检验 IPsec 策略文件的语法。

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 要配置隧道并使用 IPsec 对其加以保护，请根据 Oracle Solaris 发行版执行相应步骤：

- **从 Solaris 10 4/09 发行版开始，请执行步骤 7 至步骤 13，然后运行步骤 22 中的路由协议。**
- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请执行步骤 14 至步骤 22。**

7 在 `/etc/hostname.ip6.tun0` 文件中配置隧道 `ip6.tun0`。

a. 在 `enigma` 系统上，向 `hostname.ip6.tun0` 文件添加以下项：

```
6000:6666::aaaa:1116 6000:3333::eeee:1113 tsrc 2001::aaaa:6666:6666 tdst 2001::eeee:3333:3333 router up
```

b. 在 `partym` 系统上，向 `hostname.ip6.tun0` 文件添加以下项：

```
6000:3333::eeee:1113  6000:6666::aaaa:1116 tsrc 2001::eeee:3333:3333 tdst 2001::aaaa:6666:6666 router up
```

8 使用创建的 IPsec 策略保护隧道。

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 要将 hostname.ip6.tun0 文件的内容读入内核，请重新启动网络服务。

```
# svcadm restart svc:/network/initial:default
```

10 为 hme1 接口打开 IP 转发功能。

a. 在 enigma 系统上，将路由器项添加到 /etc/hostname6.hme1 文件。

```
2001::aaaa:6666:6666 inet6 router
```

b. 在 partym 系统上，将路由器项添加到 /etc/hostname6.hme1 文件。

```
2001::eeee:3333:3333 inet6 router
```

11 确保路由协议不在内联网内通告缺省的路由。

a. 在 enigma 系统上，将 private 标志添加到 /etc/hostname6.hme0 文件。

```
6000:6666::aaaa:1116 inet6 private
```

b. 在 partym 系统上，将 private 标志添加到 /etc/hostname6.hme0 文件。

```
6000:3333::eeee:1113 inet6 private
```

12 手动添加通过 hme0 实现的缺省路由。

a. 在 enigma 系统上，添加以下路由：

```
# route add -inet6 default 2001::aaaa:0:4
```

b. 在 partym 系统上，添加以下路由：

```
# route add -inet6 default 2001::eeee:0:1
```

13 要完成该过程，请转至步骤 22 运行路由协议。

14 配置安全隧道 ip6.tun0。

注 – 以下步骤用于在运行 Solaris 10 4/09 发行版之前的发行版的系统中配置隧道。

a. 在 enigma 系统上键入以下命令：

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333
```

b. 在 partym 系统上键入以下命令：

```
# ifconfig ip6.tun0 inet6 plumb
```

```
# ifconfig ip6.tun0 inet6  6000:3333::eeee:1113  6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666
```

15 使用创建的 IPsec 策略保护隧道。

```
# ipsecconf
```

16 初启隧道的路由器。

```
# ifconfig ip6.tun0 router up
```

17 为 hme1 接口打开 IP 转发功能。

```
# ifconfig hme1 router
```

18 确保路由协议不在内联网内通告缺省的路由。

```
# ifconfig hme0 private
```

19 在每个系统上手动添加通过 hme0 实现的缺省路由。

缺省路由必须是可以直接访问 Internet 的路由器。

a. 在 enigma 系统上，添加以下路由：

```
# route add -inet6 default 2001::aaaa:0:4
```

b. 在 partym 系统上，添加以下路由：

```
# route add -inet6 default 2001::eeee:0:1
```

20 在每个系统上，通过向 /etc/hostname6.ip6.tun0 文件中添加项来确保 VPN 在系统重新引导后启动。

该项复制在步骤 14 中传递到 ifconfig 命令的参数。

a. 在 enigma 系统上，向 hostname6.ip6.tun0 文件添加以下项：

```
6000:6666::aaaa:1116  6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333  router up
```

b. 在 partym 系统上，向 hostname6.ip6.tun0 文件添加以下项：

```
6000:3333::eeee:1113  6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666  router up
```

21 将接口文件配置为将正确的参数传送到路由选择守护进程。

a. 在 enigma 系统上，修改 /etc/hostname6.*interface* 文件。

```
# cat /etc/hostname6.hme0
## enigma
6000:6666::aaaa:1116 inet6 private

#  cat /etc/hostname6.hme1
## enigma
2001::aaaa:6666:6666 inet6 router
```

b. 在 **partym** 系统上，修改 **/etc/hostname6.** *interface* 文件。

```
# cat /etc/hostname6.hme0
## partym
6000:3333::eeee:1113 inet6 private

# cat /etc/hostname6.hme1
##
partym2001::eeee:3333:3333 inet6 router
```

22 运行路由协议。

```
# routeadm -e ipv6-routing
# routeadm -u
```

示例 20–17 使用过时的语法在使用 IPv6 的传输模式下配置 IPsec

在此示例中，管理员将 Solaris 10 7/07 系统与运行 Oracle Solaris 10 发行版的系统连接在一起。因此，管理员在配置文件中使用 Solaris 10 语法，并在 `ifconfig` 命令中包含 IPsec 算法。

管理员按照第 486 页中的 “如何使用 IPv6 在传输模式下通过 IPsec 隧道保护 VPN”过程进行操作，但在语法上进行了以下更改。

- 对于步骤 4，`ipsecinit.conf` 文件的语法如下所示：

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic can bypass IPsec.
{laddr 6000:3333::eeee:1113 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{} ipsec {encr_algs aes encr_auth_algs sha1}
```

- 对于步骤 14 至步骤 17，配置安全隧道的语法如下所示：

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666 tdst 2001::eeee:3333:3333 \
encr_algs aes encr_auth_algs sha1

# ifconfig ip6.tun0 inet6 router up
```

传送到 `ifconfig` 命令的 IPsec 策略必须与 `ipsecinit.conf` 文件中的 IPsec 策略相同。在重新引导时，每个系统都会读取 `ipsecinit.conf` 文件来获取其策略。

- 对于步骤 20，`hostname6.ip6.tun0` 文件的语法如下所示：

```
6000:6666::aaaa:1116  6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333 \
encr_algs aes encr_auth_algs sha1 router up
```

▼ 如何防止 IP 电子欺骗

要防止系统在未尝试对包进行解密的情况下将包转发至另一个接口，系统需要检查 IP 电子欺骗。一种预防方法是使用 ndd 命令设置 IP 严格目标多宿主参数。在 SMF 清单中设置了此参数时，系统重新引导时就会设置此参数。

注 – 在两个系统中执行此过程中的步骤。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 创建用于检查 IP 电子欺骗的站点专用 SMF 清单。

使用以下样例脚本 /var/svc/manifest/site/spoof_check.xml。

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type='manifest' name='Custom:ip_spoof_checking'>

<!--    This is a custom smf(5) manifest for this system. Place this
        file in /var/svc/manifest/site, the directory for local
        system customizations. The exec method uses an unstable
        interface to provide a degree of protection against IP
        spoofing attacks when this system is acting as a router.

        IP spoof protection can also be achieved by using ipfilter(5).
        If ipfilter is configured, this service can be disabled.

        Note: Unstable interfaces might be removed in later
        releases.  See attributes(5).
-->

<service
        name='site/ip_spoofcheck'
        type='service'
        version='1'>

        <create_default_instance enabled='false' />
        <single_instance />

        <!--    Don't enable spoof protection until the
                network is up.
        -->
        <dependency
                name='basic_network'
                grouping='require_all'
                restart_on='none'
                type='service'>
        <service_fmri value='svc:/milestone/network' />
        </dependency>
```

```
        <exec_method
                type='method'
                name='start'
                exec='/usr/sbin/ndd -set /dev/ip ip_strict_dst_multihoming 1'
<!--
     For an IPv6 network, use the IPv6 version of this command, as in:
                exec='/usr/sbin/ndd -set /dev/ip ip6_strict_dst_multihoming 1
-->
                timeout_seconds='60'
        />

        <exec_method
                type='method'
                name='stop'
                exec=':true'
                timeout_seconds='3'
        />

        <property_group name='startd' type='framework'>
                <propval
                        name='duration'
                        type='astring'
                        value='transient'
                />
        </property_group>

        <stability value='Unstable' />

</service>
</service_bundle>
```

3 **将此清单导入到 SMF 系统信息库。**

```
# svccfg import /var/svc/manifest/site/spoof_check.xml
```

4 **启用 `ip_spoofcheck` 服务。**

使用在清单中定义的名称 `/site/ip_spoofcheck`。

```
# svcadm enable /site/ip_spoofcheck
```

5 **检验 `ip_spoofcheck` 服务是否联机。**

```
# svcs /site/ip_spoofcheck
```

◆ ◆ ◆ 第 21 章

IP 安全体系结构（参考信息）

本章包含以下参考信息：

有关如何在网络中实现 IPsec 的说明，请参见第 20 章，配置 IPsec（任务）。有关 IPsec 的概述，请参见第 19 章，IP 安全体系结构（概述）。

IPsec 服务

服务管理工具 (Service Management Facility, SMF) 为 IPsec 提供以下服务：

- svc:/network/ipsec/policy 服务－管理 IPsec 策略。缺省情况下，此服务处于启用状态。config_file 属性的值确定了 ipsecinit.conf 文件的位置。初始值为 /etc/inet/ipsecinit.conf。
- svc:/network/ipsec/ipsecalgs 服务－管理可用于 IPsec 的算法。缺省情况下，此服务处于启用状态。
- svc:/network/ipsec/manual-key 服务－激活手动密钥管理。缺省情况下，此服务处于禁用状态。config_file 属性的值确定了 ipseckeys 配置文件的位置。初始值为 /etc/inet/secret/ipseckeys。
- svc:/network/ipsec/ike 服务－管理 IKE。缺省情况下，此服务处于禁用状态。有关可配置的属性，请参见第 551 页中的 “IKE 服务”。

有关 SMF 的信息，请参见《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。另请参见 smf(5)、svcadm(1M) 和 svccfg(1M) 手册页。

ipsecconf 命令

您可以使用 ipsecconf 命令为主机配置 IPsec 策略。当运行此命令来配置策略时，系统会在内核中创建 IPsec 策略项。系统使用这些项来检查所有传入和外发 IP 数据报的策略。转发的数据报不受使用此命令添加的策略检查的约束。ipsecconf 命令也可配置安全策略数据库 (Security Policy Database, SPD)。

- 有关如何保护转发包的信息，请参见 ifconfig(1M) 和 tun(7M) 手册页。
- 有关 IPsec 策略选项，请参见 ipsecconf(1M) 手册页。

您必须成为超级用户或承担等效角色，才能调用 ipsecconf 命令。此命令接受保护双向通信的项，同时也接受仅保护单向通信的项。

具有本地地址和远程地址格式的策略项可以借助单个策略项保护双向通信。例如，如果没有为指定的主机指定方向，则包含模式 laddr host1 和 raddr host2 的项会保护双向通信。因此，对于每台主机，只需一个策略项。

具有源地址到目标地址格式的策略项仅保护单向通信。例如，模式为 saddr host1 daddr host2 的策略项只保护传入通信或外发通信，不同时保护这两个方向的通信。因此，要保护双向通信，需要向 ipsecconf 命令传递另一个项，即 saddr host2 daddr host1。

要确保 IPsec 策略在引导计算机时处于活动状态，可以创建一个 IPsec 策略文件 /etc/inet/ipsecinit.conf。此文件在网络服务启动时读取。有关如何创建 IPsec 策略文件的说明，请参见第 443 页中的 “使用 IPsec 保护通信（任务列表）”。

从 Solaris 10 4/09发行版开始，借助 -c 选项，ipsecconf命令可检查作为参数提供的 IPsec 策略文件的语法。

由 ipsecconf 命令添加的策略项在系统重新引导后不会保留。要确保 IPsec 策略在系统引导时保持活动状态，请将策略项添加到 /etc/inet/ipsecinit.conf 文件。在当前发行版中，请刷新或启用 policy 服务。在 Solaris 10 4/09 发行版之前的发行版中，请重新引导或使用 ipsecconf 命令。例如，请参见第 443 页中的 “使用 IPsec 保护通信（任务列表）”。

ipsecinit.conf 文件

要在启动 Oracle Solaris 时启用 IPsec 安全策略，请创建一个配置文件以通过特定的 IPsec 策略项来初始化 IPsec。此文件的缺省名称为 /etc/inet/ipsecinit.conf。有关策略项及其格式的详细信息，请参见 ipsecconf(1M) 手册页。配置策略之后，您可以使用 ipsecconf 命令来查看或修改现有配置。从 Solaris 10 4/09 发行版开始，可刷新 policy 服务来修改现有配置。

ipsecinit.conf 文件样例

Oracle Solaris 软件中包括样例 IPsec 策略文件 ipsecinit.sample。您可以使用此文件作为模板来创建自己的 ipsecinit.conf 文件。ipsecinit.sample 文件包含以下示例：

```
#
# For example,
#
#     {rport 23} ipsec {encr_algs des encr_auth_algs md5}
#
# will protect the telnet traffic originating from the host with ESP using
# DES and MD5. Also:
#
#     {raddr 10.5.5.0/24} ipsec {auth_algs any}
#
# will protect traffic to or from the 10.5.5.0 subnet with AH
# using any available algorithm.
#
#
# To do basic filtering, a drop rule may be used. For example:
#
#     {lport 23 dir in} drop {}
#     {lport 23 dir out} drop {}
# will disallow any remote system from telnetting in.
#
# If you are using IPv6, it may be useful to bypass neighbor discovery
# to allow in.iked to work properly with on-link neighbors. To do that,
# add the following lines:
#
#        {ulp ipv6-icmp type 133-137 dir both } pass { }
#
# This will allow neighbor discovery to work normally.
```

ipsecinit.conf 和 ipsecconf 的安全注意事项

在网络中传输 ipsecinit.conf 文件副本时请格外小心。入侵者可能会在系统读取网络挂载的文件时也读取此文件。例如，在从 NFS 挂载的文件系统中访问或复制 /etc/inet/ipsecinit.conf 文件时，入侵者可能会更改此文件中包含的策略。

无法更改已建立连接的 IPsec 策略。其策略不能更改的套接字称为**锁定的套接字**。新策略项不保护已锁定的套接字。有关更多信息，请参见 connect(3SOCKET) 和 accept(3SOCKET) 手册页。如果有疑虑，请重新启动连接。

保护您的名称系统。如果发生以下两种情况，则您的主机名不再值得信任：

- 您的源地址是可以在网络中查找到的主机。
- 您的名称系统受到威胁。

安全漏洞通常是由工具使用不当造成的，而并非由工具本身引起。应慎用 ipsecconf 命令。请将控制台或其他硬连接的 TTY 用作最安全的操作模式。

ipsecalgs 命令

加密框架为 IPsec 提供验证和加密算法。ipsecalgs 命令可以列出每个 IPsec 协议支持的算法。ipsecalgs 配置存储在 /etc/inet/ipsecalgs 文件中。通常，不需要修改此文件。但是，如果需要修改此文件，请使用 ipsecalgs 命令。决不能直接编辑此文件。在当前发行版中，系统引导时会通过 svc:/network/ipsec/ipsecalgs:default 服务使支持的算法与内核同步。

有效的 IPsec 协议和算法由 RFC 2407 中介绍的 ISAKMP domain of interpretation, DOI（系统解释域）进行说明。通常，DOI（解释域）定义数据格式、网络通信交换类型，以及命名安全相关信息的约定。安全策略、加密算法和加密模式都属于安全相关信息。

具体而言，ISAKMP DOI 为有效的 IPsec 算法及其协议（PROTO_IPSEC_AH 和 PROTO_IPSEC_ESP）定义命名约定和编号约定。每个算法都仅与一项协议相关联。这些 ISAKMP DOI 定义位于 /etc/inet/ipsecalgs 文件中。算法和协议编号由 Internet 编号分配机构 (Internet Assigned Numbers Authority, IANA) 定义。使用 ipsecalgs 命令，可以针对 IPsec 扩展算法列表。

有关算法的更多信息，请参阅 ipsecalgs(1M) 手册页。有关加密框架的更多信息，请参见《System Administration Guide: Security Services》中的第 13 章 “Oracle Solaris Cryptographic Framework (Overview)”。

IPsec 的安全关联数据库

有关 IPsec 安全服务加密材料的信息保留在安全关联数据库 (SADB) 中。安全关联 (Security Association, SA) 保护传入包和外发包。SADB 可能由某个用户进程维护，也可能由多个协作进程（以特定类的套接字发送消息）维护。这种维护 SADB 的方法类似于 route(7P) 手册页中介绍的方法。只有超级用户或承担等效角色的用户才能访问此数据库。

in.iked 守护进程和 ipseckey 命令使用 PF_KEY 套接字接口维护 SADB。有关 SADB 如何处理请求和消息的更多信息，请参见 pf_key(7P) 手册页。

IPsec 中用于生成 SA 的实用程序

IKE（Internet 密钥交换）协议提供自动执行的 IPv4 和 IPv6 地址密钥管理。有关如何设置 IKE 的说明，请参见第 23 章，配置 IKE（任务） 。手动加密实用程序是 ipseckey 命令，在 ipseckey(1M) 手册页中对其进行了介绍。

可使用 ipseckey 命令手动填充安全关联数据库 (Security Associations Database, SADB)。通常，由于某种原因而无法使用 IKE 时，会使用手动 SA 生成。但是，如果 SPI 值是唯一的，可以同时使用手动 SA 生成和 IKE。

ipseckey 命令可用于查看系统识别的所有 SA，无论密钥是手动添加的还是由 IKE 添加的。从 Solaris 10 4/09 发行版开始，借助 -c 选项，ipseckey 命令可检查作为参数提供的密钥文件的语法。

由 ipseckey 命令添加的 IPsec SA 在系统重新引导后不会保留。在当前发行版中，要在系统引导时启用手动添加的 SA，请将相应的项添加到 /etc/inet/secret/ipseckeys 文件，然后启用 svc:/network/ipsec/manual-key:default 服务。有关过程，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。

虽然 ipseckey 命令只有有限的常规选项，但是此命令支持丰富的命令语言。您可以指定使用专用于手动加密的程序接口发送请求。有关其他信息，请参见 pf_key(7P) 手册页。

ipseckey 的安全注意事项

超级用户或拥有网络安全或网络 IPsec 管理权限配置文件的角色可以使用 ipseckey 命令输入敏感的密钥加密信息。如果入侵者获得对此文件的访问权，便会威胁 IPsec 通信的安全。

注 – 如果可能，使用 IKE 而不是通过 ipseckey 进行密钥管理。

当处理加密材料和使用 ipseckey 命令时，您应该考虑以下问题：

- 是否已更新加密材料？定期更新密钥是一项基本的安全措施。密钥更新可以防止遭到潜在的算法和密钥漏洞攻击，并且限制对已公开的密钥的破坏。
- TTY 是否联网？ipseckey 命令是否处于交互模式？
  - 在交互模式下，加密材料的安全即为此 TTY 通信的网络路径的安全。应该避免在明文 telnet 或 rlogin 会话中使用 ipseckey 命令。
  - 甚至本地窗口也可能受到读取窗口事件的隐藏程序的攻击。
- 是否已使用 -f 选项？是否通过网络访问文件？此文件是否完全公开？
  - 入侵者可能会在系统读取网络挂载的文件时也读取此文件。您应该避免使用包含加密材料的完全公开文件。
  - 保护您的名称系统。如果发生以下两种情况，则您的主机名不再值得信任：
    - 您的源地址是可以在网络中查找到的主机。
    - 您的名称系统受到威胁。

安全漏洞通常是由工具使用不当造成的，而并非由工具本身引起。应慎用 ipseckey 命令。请将控制台或其他硬连接的 TTY 用作最安全的操作模式。

其他实用程序的 IPsec 扩展

ifconfig 命令具有用来管理隧道接口上的 IPsec 策略的选项。snoop 命令可以解析 AH 头和 ESP 头。

ifconfig 命令和 IPsec

在 Solaris 10、Solaris 10 7/05、Solaris 10 1/06 和 Solaris 10 11/06 发行版中： 为了支持 IPsec，ifconfig 命令提供以下安全选项。在 Solaris 10 7/07 发行版中，这些安全选项由 ipsecconf 命令处理。

- auth_algs
- encr_auth_algs
- encr_algs

必须在一次调用中为隧道指定所有 IPsec 安全选项。例如，如果只使用 ESP 保护通信，则一次使用两个安全选项配置隧道 ip.tun0，如下所示：

```
# ifconfig ip.tun0 encr_algs aes encr_auth_algs md5
```

同样，ipsecinit.conf 项将使用两个安全选项一次性配置隧道，如下所示：

```
# WAN traffic uses ESP with AES and MD5.
   {} ipsec {encr_algs aes encr_auth_algs md5}
```

auth_algs 安全选项

此选项使用指定的验证算法为隧道启用 IPsec AH。auth_algs 选项的格式如下所示：

auth_algs *authentication-algorithm*

对于算法，您可以通过指定一个数字或一个算法名称（包括参数 *any*）来表示没有特定的算法首选项。要禁用隧道安全性，请指定以下选项：

```
auth_algs none
```

要获得可用验证算法的列表，请运行 ipsecalgs 命令。

注 – auth_algs 选项不能用于 NAT 遍历。有关更多信息，请参见第 438 页中的 “IPsec 和 NAT 遍历”。

---

encr_auth_algs 安全选项

此选项使用指定的验证算法为隧道启用 IPsec ESP。encr_auth_algs 选项的格式如下所示：

encr_auth_algs *authentication-algorithm*

对于算法，您可以通过指定一个数字或一个算法名称（包括参数 *any*）来表示没有特定的算法首选项。如果指定一个 ESP 加密算法，但是未指定验证算法，则 ESP 验证算法的缺省值为参数 *any*。

要获得可用验证算法的列表，请运行 `ipsecalgs` 命令。

encr_algs 安全选项

此选项使用指定的加密算法为隧道启用 IPsec ESP。`encr_algs` 选项的格式如下所示：

`encr_algs` *encryption-algorithm*

对于算法，您可以指定一个数字或一个算法名称。要禁用隧道安全性，请指定以下选项：

```
encr_algs none
```

如果指定一个 ESP 验证算法，但是未指定加密算法，则 ESP 加密算法的缺省值为参数 *null*。

要获得可用加密算法的列表，请运行 `ipsecalgs` 命令。

snoop 命令和 IPsec

`snoop` 命令可以解析 AH 头和 ESP 头。由于 ESP 对自己的数据进行加密，因此，`snoop` 命令不能查看受 ESP 保护的加密头。AH 不会对数据进行加密。因此，可以使用 `snoop` 命令来检查受 AH 保护的通信。此命令的 `-V` 选项显示何时对包使用 AH。有关更多详细信息，请参见 `snoop`(1M) 手册页。

有关受保护包中的详细 `snoop` 输出样例，请参见第 457 页中的 “如何检验包是否受 IPsec 保护”。

◆ ◆ ◆ 第 22 章

22 Internet 密钥交换（概述）

Internet 密钥交换 (Internet Key Exchange, IKE) 自动进行 IPsec 的密钥管理。Oracle Solaris 实现 IKEv1。本章包含有关 IKE 的以下信息：

有关实现 IKE 的说明，请参见第 23 章，配置 IKE（任务）。有关参考信息，请参见第 24 章，Internet 密钥交换（参考信息）。有关 IPsec 的概述信息，请参见第 19 章，IP 安全体系结构（概述）。

IKE 中的新增功能

Solaris 10 4/09：从此发行版开始，服务管理工具 (Service Management Facility, SMF) 将 IKE 做为一种服务来管理。缺省情况下，会禁用 `svc:/network/ipsec/ike:default` 服务。而且，在此发行版中，还提供了网络 IPsec 管理权限配置文件以用于管理 IPsec 和 IKE。

Solaris 10 7/07：从此发行版开始，IKE 可使用 AES 算法，并可在全局区域中进行配置以便在非全局区域中使用。

- 通过使用 `SO_ALLZONES` 套接字选项，IKE 可以处理非全局区域中的通信流量。
- 有关 Oracle Solaris 新增功能的完整列表以及 Solaris 发行版的说明，请参见《Oracle Solaris 10 1/13 新增功能》。

使用 IKE 进行密钥管理

对 IPsec 安全关联 (Security Association, SA) 的加密材料进行的管理称为**密钥管理**。自动密钥管理需要用于创建、验证和交换密钥的安全信道。Oracle Solaris 使用 Internet 密钥交换 (Internet Key Exchange, IKE) 版本 1 自动进行密钥管理。IKE 可轻松扩展以便为大量通信提供安全信道。IPv4 和 IPv6 包中的 IPsec SA 可以利用 IKE。

IKE 可以利用可用的硬件加速和硬件存储。通过硬件加速器，可以将密集的密钥操作转移到系统外处理。硬件上的密钥存储提供了额外的一层保护。

IKE 密钥协商

IKE 守护进程 `in.iked` 以安全方式为 IPsec SA 协商和验证加密材料。该守护进程使用来自 OS 提供的内部函数的随机密钥种子。IKE 提供完全正向保密 (Perfect Forward Secrecy, PFS)。在 PFS 中，不能使用保护数据传输的密钥派生其他密钥。此外，不重新使用用于创建数据传输密钥的种子。请参见 `in.iked(1M)` 手册页。

IKE 密钥术语

下表列出在密钥协商中使用的术语，提供其常用的首字母缩略词，并给出每个术语的定义和用法。

表 22–1 密钥协商术语及其首字母缩略词和用法

密钥协商术语	首字母缩略词	定义和用法
Key exchange（密钥交换）		生成非对称加密算法的密钥的过程。两种主要方法是 RSA 和 Diffie-Hellman 协议。
Diffie-Hellman 算法	DH	提供密钥生成和密钥验证的密钥交换算法。通常称为**经过验证的密钥交换**。
RSA 算法	RSA	提供密钥生成和密钥传输的密钥交换算法。此协议以其三个创建者 Rivest、Shamir 和 Adleman 命名。
Perfect forward secrecy（完全正向保密）	PFS	仅适用于经过验证的密钥交换。在 PFS 中，不能使用保护数据传输的密钥派生其他密钥。此外，也不能使用保护数据传输的密钥的源派生其他密钥。
Oakley group（Oakley 组）		一种以安全方式为阶段 2 建立密钥的方法。Oakley 组用于协商 PFS。请参见 The Internet Key Exchange (IKE) (`http://www.faqs.org/rfcs/rfc2409.html`)（Internet 密钥交换）的第 6 节。

IKE 阶段 1 交换

阶段 1 交换称为**主模式**。在阶段 1 交换中，IKE 使用公钥加密方法向对等 IKE 实体进行自我验证。结果是 Internet 安全关联和密钥管理协议 (Internet Security Association and Key Management Protocol, ISAKMP) 安全关联 (Security Association, SA)。ISAKMP SA 是 IKE 用于协商 IP 数据报的加密材料的安全信道。与 IPsec SA 不同，ISAKMP SA 是双向的，因此只需要一个安全关联。

IKE 在阶段 1 交换中协商加密材料的方式是可配置的。IKE 从 `/etc/inet/ike/config` 文件读取配置信息。配置信息包括：

- 全局参数，如公钥证书的名称
- 是否使用完全正向保密 (Perfect Forward Secrecy, PFS)
- 受影响的接口
- 安全协议及其算法
- 验证方法

两种验证方法是预先共享的密钥和公钥证书。公钥证书可以自签名。或者，证书可以由来自公钥基础结构 (Public Key Infrastructure, PKI) 组织的 certificate authority, CA（证书颁发机构）颁发。

IKE 阶段 2 交换

阶段 2 交换称为**快速模式**。在阶段 2 交换中，IKE 在运行 IKE 守护进程的系统之间创建和管理 IPsec SA。IKE 使用在阶段 1 交换中创建的安全通道保护加密材料的传输。IKE 守护进程使用 `/dev/random` 设备从随机数生成器创建密钥。该守护进程按可配置的速率刷新密钥。加密材料可供在 IPsec 策略的配置文件 `ipsecinit.conf` 中指定的算法使用。

IKE 配置选择

`/etc/inet/ike/config` 配置文件包含 IKE 策略项。为了使两个 IKE 守护进程相互验证，这些项必须是有效的。此外，加密材料必须可用。配置文件中的项确定使用加密材料验证阶段 1 交换的方法。可以选择预先共享的密钥或公钥证书。

项 `auth_method preshared` 指示使用预先共享的密钥。除 `preshared` 之外的 `auth_method` 值指示要使用公钥证书。公钥证书可以自签名，也可以从 PKI 组织安装。有关更多信息，请参见 `ike.config(4)` 手册页。

使用预先共享的密钥验证的 IKE

预先共享的密钥用于验证两个或多个对等方系统。预先共享的密钥为由一个系统上的管理员创建的十六位数字或 ASCII 字符串。然后与对等方系统的管理员在带外共享此密钥。如果预先共享的密钥被入侵者拦截，则该入侵者可以假扮某个对等方系统。

使用此验证方法的对等方中预先共享的密钥必须相同。这些密钥与特定的 IP 地址相关联。这些密钥放置在每个系统上的 /etc/inet/secret/ike.preshared 文件中。ike.preshared 文件用于 IKE，而 ipseckeys 文件用于 IPsec。一旦 ike.preshared 文件中的密钥泄露，将危及所有传输。由一个管理员控制通信系统时，密钥是最安全的。有关更多信息，请参见 ike.preshared(4) 手册页。

IKE，使用公钥证书

使用公钥证书，通信系统就无需在带外共享秘密的加密材料。公钥使用 Diffie-Hellman algorithm（Diffie-Hellman 算法）(DH) 来验证和协商密钥。公钥证书有两种类型。这些证书可以自签名，也可以由 certificate authority, CA（证书颁发机构）认证。

自签名的公钥证书由管理员创建。ikecert certlocal -ks 命令为系统创建公钥/私钥对的私钥部分。然后，从远程系统获取 X.509 格式的自签名证书输出。远程系统的证书是用于创建密钥对的公钥部分的 ikecert certdb 命令的输入。在通信系统上，自签名的证书驻留在 /etc/inet/ike/publickeys 目录中。使用 -T 选项时，证书驻留在连接的硬件上。

自签名的证书介于预先共享的密钥和 CA 中间。与预先共享的密钥不同，自签名的证书可以在移动计算机或可能重新编号的系统上使用。要为没有固定编号的系统对证书自行签名，请使用 DNS (www.example.org) 或 email (root@domain.org) 替换名称。

公钥可以由 PKI 或 CA 组织提供。在 /etc/inet/ike/publickeys 目录中安装公钥及其相应的 CA。使用 -T 选项时，证书驻留在连接的硬件上。供应商还发布证书撤销列表 (Certificate Revocation List, CRL)。除安装密钥和 CA 以外，您还负责在 /etc/inet/ike/crls 目录中安装 CRL。

CA 的优势在于由外部组织而不是由站点管理员认证。在某种意义上，CA 是经过确认的证书。与自签名的证书一样，CA 可以在移动计算机或可能重新编号的系统上使用。与自签名的证书不同的是，CA 可以非常容易地扩展以保护大量的通信系统。

IKE 和硬件加速

IKE 算法的计算开销很大，尤其是在阶段 1 交换中。处理大量交换的系统可以使用 Sun Crypto Accelerator 1000 或 Sun Crypto Accelerator 6000 板处理公钥操作。Sun Crypto Accelerator 6000 和 Sun Crypto Accelerator 4000 板也可用于处理开销很大的阶段 1 计算。

有关如何配置 IKE 以将其计算转移到加速器板的信息，请参见第 544 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 4000 板”。有关如何存储密钥的信息，请参见第 544 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 4000 板”和 cryptoadm(1M) 手册页。

IKE 和硬件存储

公钥证书、私钥和公钥可以存储在 Sun Crypto Accelerator 6000 或 Sun Crypto Accelerator 4000 板上。对于 RSA 加密，这些板支持最多 2048 位的密钥。对于 DSA 加密，这些板支持最多 1024 位的密钥。Sun Crypto Accelerator 6000 板支持 SHA-512 和 ECC 算法。

有关如何配置 IKE 以访问该板的信息，请参见第 544 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 4000 板”。有关如何向该板添加证书和公钥的信息，请参见第 530 页中的 “如何在硬件中生成和存储公钥证书”。

IKE 实用程序和文件

下表概述了 IKE 策略的配置文件、IKE 密钥的存储位置以及实现 IKE 的各种命令和服务。有关服务的更多信息，请参见《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。

表 22–2 IKE 配置文件、密钥存储位置、命令和服务

文件、位置、命令或服务	说明	手册页
`svc:/network/ipsec/ike`	在当前发行版中，为管理 IKE 的 SMF 服务。	`smf(5)`
`/usr/lib/inet/in.iked`	Internet 密钥交换 (Internet Key Exchange, IKE) 守护进程。激活自动密钥管理。在当前发行版中，`ike` 服务会启用此守护进程。在早期发行版中，会使用 `in.iked` 命令。	`in.iked(1M)`
`/usr/sbin/ikeadm`	用于查看和修改 IKE 策略的 IKE 管理命令。	`ikeadm(1M)`
`/usr/sbin/ikecert`	用于处理包含公钥证书的本地数据库的证书数据库管理命令。这些数据库也可以存储在连接的硬件上。	`ikecert(1M)`
`/etc/inet/ike/config`	IKE 策略的缺省配置文件。包含用于匹配传入 IKE 请求和准备外发 IKE 请求的站点规则。 在当前发行版中，如果此文件存在，在启用 `ike` 服务时，`in.iked` 守护进程会启动。可以使用 `svccfg` 命令更改此文件的位置。	`ike.config(4)`
`ike.preshared`	`/etc/inet/secret` 目录中的预先共享密钥文件。包含用于阶段 1 交换中验证的秘密加密材料。在用预先共享的密钥配置 IKE 时使用。	`ike.preshared(4)`
`ike.privatekeys`	`/etc/inet/secret` 目录中的私钥目录。包含公钥/私钥对的私钥部分。	`ikecert(1M)`
`publickeys` 目录	`/etc/inet/ike` 目录中包含公钥和证书文件的目录。包含公钥/私钥对的公钥部分。	`ikecert(1M)`
`crls` 目录	`/etc/inet/ike` 目录中包含公钥和证书文件的撤销列表的目录。	`ikecert(1M)`

表 22–2 IKE 配置文件、密钥存储位置、命令和服务 （续）

文件、位置、命令或服务	说明	手册页
Sun Crypto Accelerator 1000 板	通过从操作系统转移操作来加速公钥操作的硬件。	ikecert(1M)
Sun Crypto Accelerator 4000 板	通过从操作系统转移操作来加速公钥操作的硬件。该板还存储公钥、私钥和公钥证书。Sun Crypto Accelerator 6000 板是第 3 级别的 FIPS 140-2 认证设备。	ikecert(1M)

Oracle Solaris 10 发行版对 IKE 的更改

从 Solaris 9 发行版开始，IKE 包括以下功能：

- 可以使用 IKE 在 IPv6 网络中自动进行 IPsec 的密钥交换。有关更多信息，请参见第 504 页中的 “使用 IKE 进行密钥管理”。

  注 – 不能使用 IKE 在非全局区域中管理 IPsec 的密钥。

- IKE 中的公钥操作可以由 Sun Crypto Accelerator 1000 板或 Sun Crypto Accelerator 4000 板加速。有关操作都被转移到该板中。这样将加快加密过程，从而降低对操作系统资源的需求。有关更多信息，请参见第 506 页中的 “IKE 和硬件加速”。有关过程，请参见第 543 页中的 “将 IKE 配置为查找连接的硬件”。
- 公钥证书、私钥和公钥可以存储在 Sun Crypto Accelerator 4000 板上。有关密钥存储的更多信息，请参见第 507 页中的 “IKE 和硬件存储”。
- 可以使用 IKE 从 NAT 盒 (NAT box) 之后自动进行 IPsec 的密钥交换。但是，遍历 NAT 的 IPsec ESP 密钥不能由硬件加速。有关更多信息，请参见第 438 页中的 “IPsec 和 NAT 遍历”。有关过程，请参见第 536 页中的 “为移动系统配置 IKE（任务列表）”。
- 重新传输参数和包超时参数已添加到 `/etc/inet/ike/config` 文件中。这些参数调整 IKE 阶段 1（主模式）协商，以处理网络干扰、网络通信流量过大以及与具有 IKE 协议的不同实现的平台的交互操作。有关这些参数的详细信息，请参见 `ike.config(4)` 手册页。有关过程，请参见第 547 页中的 “更改 IKE 传输参数（任务列表）”。

◆ ◆ ◆ 第 23 章

23 配置 IKE（任务）

本章介绍如何为系统配置 Internet 密钥交换 (Internet Key Exchange, IKE)。配置 IKE 后，它将自动为网络上的 IPsec 生成加密材料。

本章包含以下信息：

有关 IKE 的概述信息，请参见第 22 章，Internet 密钥交换（概述）。有关 IKE 的参考信息，请参见第 24 章，Internet 密钥交换（参考信息）。有关更多过程，请参见 `ikeadm(1M)`、`ikecert(1M)` 和 `ike.config(4)` 手册页的示例部分。

配置 IKE（任务列表）

可以使用预先共享的密钥、自签名证书和证书颁发机构 (Certificate Authority, CA) 所颁发的证书来验证 IKE。规则将特定的 IKE 验证方法与受保护的端点相关联。因此，可以在系统上使用一种或所有 IKE 验证方法。利用指向 PKCS #11 库的指针，证书可以使用连接的硬件加速器。

配置 IKE 后，完成使用 IKE 配置的 IPsec 任务。下表提供了着重说明特定 IKE 配置的任务列表。

任务	说明	参考
使用预先共享的密钥配置 IKE	保护共享一个密钥的系统之间的通信。	第 510 页中的 “使用预先共享的密钥配置 IKE（任务列表）”

任务	说明	参考
使用公钥证书配置 IKE	使用公钥证书保护通信。这些证书可以是自签名的，也可以由 PKI 组织认证。	第 519 页中的 “使用公钥证书配置 IKE（任务列表）”
跨 NAT 边界	将 IPsec 和 IKE 配置为与移动系统进行通信。	第 536 页中的 “为移动系统配置 IKE（任务列表）”
将 IKE 配置为在连接的硬件上生成和存储公钥证书	使 Sun Crypto Accelerator 1000 板或 Sun Crypto Accelerator 4000 板可以加快 IKE 操作。此外，使 Sun Crypto Accelerator 4000 板可以存储公钥证书。	第 543 页中的 “将 IKE 配置为查找连接的硬件”
调整阶段 1 密钥协商参数	更改 IKE 密钥协商的时间安排。	第 547 页中的 “更改 IKE 传输参数（任务列表）”

使用预先共享的密钥配置 IKE（任务列表）

下表包含使用预先共享的密钥配置和维护 IKE 的过程的链接。

任务	说明	参考
使用预先共享的密钥配置 IKE	创建 IKE 策略文件和要共享的一个密钥。	第 510 页中的 “如何使用预先共享的密钥配置 IKE”
在正运行的 IKE 系统上刷新预先共享的密钥	在通信系统上为 IKE 添加新的加密材料。	第 513 页中的 “如何刷新 IKE 预先共享密钥”
将预先共享的密钥添加到正运行的 IKE 系统	将新的 IKE 策略项和新的加密材料添加到当前实施 IKE 策略的系统。	第 516 页中的 “如何为 `ipsecinit.conf` 中的新策略项添加 IKE 预先共享密钥”
检查预先共享的密钥是否完全相同	在两个系统上显示预先共享的密钥，以查看它们是否完全相同。	第 518 页中的 “如何检验 IKE 预先共享密钥是否完全相同”

使用预先共享的密钥配置 IKE

使用预先共享的密钥是验证 IKE 的最简单方法。如果要将两个系统配置为使用 IKE，而且您是这两个系统的管理员，则使用预先共享的密钥是一个良好的选择。但是，与公钥证书不同，预先共享的密钥与特定的 IP 地址相关联。预先共享的密钥不能用于移动系统或可能重新编号的系统。

▼ 如何使用预先共享的密钥配置 IKE

IKE 实现提供了采用可变密钥长度的算法。所选的密钥长度是由站点安全性确定的。通常，密钥越长，提供的安全性就越高。

以下过程使用系统名称 enigma 和 partym。请用您的系统名称替换名称 enigma 和 partym。

1 **在系统控制台上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

---

2 **在每个系统上，将文件 /etc/inet/ike/config.sample 复制到文件 /etc/inet/ike/config。**

3 **在每个系统上的 ike/config 文件中输入规则和全局参数。**

此文件中的规则和全局参数应该允许系统的 ipsecinit.conf 文件中的 IPsec 策略可以成功实施。以下是与第 445 页中的 “如何使用 IPsec 保证两个系统之间的通信安全”中的 ipsecinit.conf 示例配合使用的 ike/config 示例。

a. **例如，在 enigma 系统上修改 /etc/inet/ike/config 文件：**

```
### ike/config file on enigma, 192.168.116.16

## Global parameters
#
## Phase 1 transform defaults
p1_lifetime_secs 14400
p1_nonce_len 40
#
## Defaults that individual rules can override.
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2
#
## The rule to communicate with partym
#  Label must be unique
{ label "enigma-partym"
  local_addr 192.168.116.16
  remote_addr 192.168.13.213
  p1_xform
    { auth_method preshared oakley_group 5 auth_alg sha1 encr_alg aes }
  p2_pfs 5
}
```

b. **在 partym 系统上修改 /etc/inet/ike/config 文件：**

```
### ike/config file on partym, 192.168.13.213
## Global Parameters
#
p1_lifetime_secs 14400
p1_nonce_len 40
```

```
#
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2

## The rule to communicate with enigma
#  Label must be unique
{ label "partym-enigma"
  local_addr 192.168.13.213
  remote_addr 192.168.116.16
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha1 encr_alg aes }
p2_pfs 5
}
```

4 **在每个系统上，检验该文件的语法。**

```
# /usr/lib/inet/in.iked -c -f /etc/inet/ike/config
```

5 **生成随机数以用作加密材料。**

如果站点具有随机数生成器，请使用该生成器。在 Oracle Solaris 10 系统上，可以使用 od 命令。例如，以下命令列显两行十六进制数：

```
% od -X -A n /dev/random | head -2
         f47cb0f4 32e14480 951095f8 2b735ba8
         0a9467d0 8f92c880 68b6a40e 0efe067d
```

有关 od 命令的说明，请参见第 451 页中的 “如何在 Oracle Solaris 系统上生成随机数”和 od(1) 手册页。

注 – 其他操作系统可能需要 ASCII 加密材料。要以十六进制格式和 ASCII 格式生成相同的密钥，请参见示例 23–1。

---

6 **利用步骤 5 的输出构造一个密钥。**

```
f47cb0f432e14480951095f82b735ba80a9467d08f92c88068b6a40e
```

此过程中的验证算法是 SHA–1，如步骤 3 所示。散列的大小（即验证算法输出的大小）确定预先共享密钥的最小建议大小。SHA–1 算法的输出是 160 位或 40 个字符。示例密钥的长度是 56 个字符，这将提供其他加密材料供 IKE 使用。

7 **在每个系统上创建文件 /etc/inet/secret/ike.preshared。**

在每个文件中放置预先共享的密钥。

a. **例如，在 enigma 系统上，ike.preshared 文件的显示与以下信息类似：**

```
# ike.preshared on enigma, 192.168.116.16
#...
{ localidtype IP
    localid 192.168.116.16
    remoteidtype IP
    remoteid 192.168.13.213
    # enigma and partym's shared key in hex (192 bits)
```

```
        key f47cb0f432e14480951095f82b735ba80a9467d08f92c88068b6a40e
        }
```

b. 在 partym 系统上，ike.preshared 文件的显示与以下信息类似：

```
# ike.preshared on partym, 192.168.13.213
#...
{ localidtype IP
    localid 192.168.13.213
    remoteidtype IP
    remoteid 192.168.116.16
    # partym and enigma's shared key in hex (192 bits)
    key f47cb0f432e14480951095f82b735ba80a9467d08f92c88068b6a40e
    }
```

注 – 每个系统上的预先共享密钥必须完全相同。

示例 23–1 **为运行不同操作系统的两个系统生成相同的加密材料**

Oracle Solaris 的 IPsec 功能可与其他操作系统上的 IPsec 交互操作。如果您的系统与需要 ASCII 预先共享密钥的系统通信，则需要以两种格式（十六进制格式和 ASCII 格式）生成一个密钥。

在此示例中，Oracle Solaris 系统管理员需要 56 个字符的加密材料。该管理员使用以下命令从 ASCII 口令短语中生成十六进制的密钥。在所有 Oracle Solaris 系统中，`-tx1` 选项每次输出一个字节。

```
# /bin/echo "papiermache with cashews and\c" | od -tx1 | cut -c 8-55 | \
tr -d '\n' | tr -d ' ' | awk '{print}'
7061706965726d6163686520776974682063617368657773206 16e64
```

通过删除偏移和串联十六进制输出，Oracle Solaris 系统的十六进制密钥为 `7061706965726d6163686520776974682063617368657773206 16e64` 。管理员可以将该值放入 Oracle Solaris 系统上的 `ike.preshared` 文件中。

```
# Shared key in hex (192 bits)
key 7061706965726d6163686520776974682063617368657773206 16e64
```

在需要 ASCII 预先共享密钥的系统中，口令短语就是预先共享的密钥。Oracle Solaris 系统管理员使用口令短语 `papiermache with cashews and` 给其他管理员打电话。

▼ 如何刷新 IKE 预先共享密钥

此过程假定您希望按固定的时间间隔替换现有的预先共享密钥。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2. **生成随机数，并构造适当长度的密钥。**

   有关详细信息，请参见第 451 页中的 “如何在 Oracle Solaris 系统上生成随机数”。如果要生成预先共享密钥的 Oracle Solaris 系统与需要 ASCII 的操作系统通信，请参见示例 23–1。

3. **将当前密钥替换为新密钥。**

   例如，在主机 `enigma` 和 `partym` 上，将 `/etc/inet/secret/ike.preshared` 文件中 `key` 的值替换为一个相同长度的新数值。

4. **将新密钥读入内核。**

   - **从 Solaris 10 4/09 发行版开始，请重新启动 `ike` 服务。**

     ```
     # svcadm enable ike
     ```

   - **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请中止并重新启动 `in.iked` 守护进程。**

     a. **检查 `in.iked` 守护进程的特权级别。**

        ```
        # /usr/sbin/ikeadm get priv
        Current privilege level is 0x0, base privileges enabled
        ```

        如果该命令返回特权级别 `0x1` 或 `0x2`，则您可以更改加密材料。级别 `0x0` 不允许执行修改或查看加密材料的操作。缺省情况下，`in.iked` 守护进程按特权级别 `0x0` 运行。

     b. **如果特权级别是 `0x0`，请先中止再重新启动该守护进程。**

        守护进程重新启动时，它将读取新版本的 `ike.preshared` 文件。

        ```
        # pkill in.iked
        # /usr/lib/inet/in.iked
        ```

     c. **如果特权级别是 `0x1` 或 `0x2`，则读入新版本的 `ike.preshared` 文件。**

        ```
        # ikeadm read preshared
        ```

▼ 如何查看 IKE 预先共享密钥

缺省情况下，`ikeadm` 命令会阻止您在阶段 1 SA 的转储中查看实际密钥。在调试期间查看密钥很有用。

要查看实际密钥，您必须提高守护进程的特权级别。有关特权级别的说明，请参见第 553 页中的 “`ikeadm` 命令”。

注 – 要在 Solaris 10 4/09 发行版之前的发行版中执行此过程，请参见示例 23–2。

开始之前 已配置 IKE，并且 ike 服务正在运行。

1. **查看 IKE 预先共享密钥。**

```
# ikeadm
ikeadm> dump preshared
```

2. **如果出现错误，请提高 in.iked 守护进程的特权级别。**

   a. **提高 SMF 系统信息库中 in.iked 守护进程的特权级别。**

```
# svcprop -p config/admin_privilege ike
base
# svccfg -s ike setprop config/admin_privilege=keymat
```

   b. **提高正在运行的 in.iked 守护进程的特权级别。**

```
# svcadm refresh ike ; svcadm restart ike
```

   c. **可选确认特权级别为 keymat。**

```
# svcprop -p config/admin_privilege ike
keymat
```

   d. **通过再次运行步骤 1 查看密钥。**

3. **将 IKE 守护进程恢复为 base 特权级别。**

   a. **查看密钥后，将特权级别恢复为缺省级别。**

```
# svccfg -s ike setprop config/admin_privilege=base
```

   b. **刷新 IKE，然后再重新启动 IKE。**

```
# svcadm refresh ike ; svcadm restart ike
```

示例 23–2 在 Solaris 10 4/09 发行版之前的发行版中检验 IKE 预先共享密钥

在以下示例中，管理员要在运行非当前 Oracle Solaris 10 发行版的 Solaris 系统中查看密钥。管理员想要检验该系统中的密钥是否与通信系统中的密钥完全相同。检验两个系统中的密钥是否完全相同后，管理员将特权级别恢复为 0。

- 首先，管理员确定 in.iked 守护进程的特权级别。

```
adm1 # /usr/sbin/ikeadm get priv
Current privilege level is 0x0, base privileges enabled
```

- 由于特权级别不是 0x1 或 0x2，所以管理员会停止 in.iked 守护进程，然后将特权级别提高到 2。

```
adm1 # pkill in.iked
adm1 # /usr/lib/inet/in.iked -p 2
Setting privilege level to 2
```

- 管理员显示密钥。

```
adm1 # ikeadm dump preshared
PSKEY: Preshared key (24 bytes): f47cb.../192
LOCIP: AF_INET: port 0, 192.168.116.16 (adm1).
REMIP: AF_INET: port 0, 192.168.13.213 (com1).
```

- 管理员远程登录到通信系统，并确定密钥是否完全相同。
- 然后，管理员恢复 base 特权级别。

```
# ikeadm set priv base
```

▼ 如何为 ipsecinit.conf 中的新策略项添加 IKE 预先共享密钥

如果将 IPsec 策略项添加到相同对等方之间的工作配置，则需要刷新 IPsec 策略服务。无需重新配置或重新启动 IKE。

如果将新的对等方添加到 IPsec 策略，则除了进行 IPsec 更改之外，还必须修改 IKE 配置。

注 – 要在 Solaris 10 4/09 发行版之前的发行版中执行此过程，请参见示例 23–3。

开始之前 您已更新了对等方系统的 `ipsecinit.conf` 文件并刷新了 IPsec 策略。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

---

2 在此系统中，生成随机数，并构造一个 64 位到 448 位的密钥。

有关详细信息，请参见第 451 页中的 “如何在 Oracle Solaris 系统上生成随机数”。如果要生成预先共享密钥的 Oracle Solaris 系统与需要 ASCII 的操作系统通信，请参见示例 23–1。

3 **以某种方法将密钥发送给远程系统的管理员。**

需要同时添加相同的预先共享密钥。密钥的安全性仅与传输机制的安全性相同。带外机制（如已注册的邮件或受保护的传真机）是最佳的。您也可以使用 `ssh` 会话管理这两个系统。

4 **为 IKE 创建一个规则以管理 `enigma` 和新的对等方 `ada` 的密钥。**

a. **在 `enigma` 系统上，将以下规则添加到 `/etc/inet/ike/config` 文件：**

```
### ike/config file on enigma, 192.168.116.16

## The rule to communicate with ada

{label "enigma-to-ada"
 local_addr 192.168.116.16
 remote_addr 192.168.15.7
 p1_xform
 {auth_method preshared oakley_group 5 auth_alg sha1 encr_alg blowfish}
 p2_pfs 5
    }
```

b. **在 `ada` 系统上，添加以下规则：**

```
### ike/config file on ada, 192.168.15.7

## The rule to communicate with enigma

{label "ada-to-enigma"
 local_addr 192.168.15.7
 remote_addr 192.168.116.16
 p1_xform
 {auth_method preshared oakley_group 5 auth_alg sha1 encr_alg blowfish}
 p2_pfs 5
}
```

5 **确保 IKE 预先共享密钥在重新引导时是可用的。**

a. **在 `enigma` 系统上，将以下信息添加到 `/etc/inet/secret/ike.preshared` 文件：**

```
# ike.preshared on enigma for the ada interface
#
{ localidtype IP
  localid 192.168.116.16
  remoteidtype IP
  remoteid 192.168.15.7
  # enigma and ada's shared key in hex (32 - 448 bits required)
  key 8d1fb4ee500e2bea071deb2e781cb48374411af5a9671714672bb1749ad9364d
}
```

b. **在 `ada` 系统上，将以下信息添加到 `ike.preshared` 文件：**

```
# ike.preshared on ada for the enigma interface
#
{ localidtype IP
  localid 192.168.15.7
  remoteidtype IP
```

```
    remoteid 192.168.116.16
    # ada and enigma's shared key in hex (32 - 448 bits required)
    key 8d1fb4ee500e2bea071deb2e781cb48374411af5a9671714672bb1749ad9364d
  }
```

6 **在每个系统上，刷新 ike 服务。**

```
# svcadm refresh ike
```

7 **检验系统是否可以进行通信。**

请参见第 518 页中的 “如何检验 IKE 预先共享密钥是否完全相同”。

示例 23–3 为新的 IPsec 策略项添加 IKE 预先共享密钥

在以下示例中，管理员要将预先共享的密钥添加到运行非当前 Oracle Solaris 10 发行版的 Solaris 系统。管理员遵循前面的过程来修改 `ike/config` 和 `ike.preshared` 文件，然后生成密钥并联系远程系统。

- 在生成新密钥之前，管理员将 `in.iked` 守护进程的特权级别设置为 2。

```
# pkill in.iked
# /usr/lib/inet/in.iked -p 2
Setting privilege level to 2
```

- 将密钥发送给另一个系统并将新密钥添加到系统后，管理员降低了特权级别。

```
# ikeadm set priv base
```

- 最后，管理员将新的 IKE 规则读入内核。

```
# ikeadm read rules
```

接下来的步骤 如果建立 IPsec 策略未完成，请返回到 IPsec 过程以启用或刷新 IPsec 策略。

▼ 如何检验 IKE 预先共享密钥是否完全相同

如果通信系统上的预先共享密钥不是完全相同的，则系统无法进行验证。

开始之前 在要测试的两个系统之间已配置并启用 IPsec。您运行的是当前 Oracle Solaris 10 发行版。

注 – 要在 Solaris 10 4/09 发行版之前的发行版中执行此过程，请参见示例 23–2。

---

1 **在系统控制台上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2 **在每个系统上，检查 `in.iked` 守护进程的权限级别。**

```
# svcprop -p config/admin_privilege ike
base
```

- **如果特权级别为 `keymat`，请继续执行步骤 3。**
- **如果权限级别为 `base` 或 `modkeys`，请提高权限级别。**

  然后，刷新并重新启动 `ike` 服务。

  ```
  # svccfg -s ike setprop config/admin_privilege=keymat
  # svcadm refresh ike ; svcadm restart ike
  # svcprop -p config/admin_privilege ike
  keymat
  ```

3 **在每个系统上，查看预先共享密钥的信息。**

```
# ikeadm dump preshared
PSKEY: Preshared key (24 bytes): f47cb.../192
LOCIP: AF_INET: port 0, 192.168.116.16 (enigma).
REMIP: AF_INET: port 0, 192.168.13.213 (partym).
```

4 **比较两个转储。**

如果预先共享的密钥不是完全相同的，则在 `/etc/inet/secret/ike.preshared` 文件中将一个密钥替换为另一个密钥。

5 **检验完成后，在每个系统中将特权级别恢复为缺省级别。**

```
# svccfg -s ike setprop config/admin_privilege=base
# svcadm restart ike
```

使用公钥证书配置 IKE（任务列表）

下表提供了为 IKE 创建公钥证书的过程的链接。这些过程包括如何在连接的硬件上加速和存储证书。

公共证书必须是唯一的，因此公钥证书的创建者应为该证书生成任意的唯一名称。通常，将使用 X.509 标识名。此外，也可使用替代名称进行标识。这些名称的格式为 *tag=value*。这些值是任意的，但值的格式必须与其标记类型对应。例如，`email` 标记的格式为 *name@domain.suffix*。

任务	说明	参考
使用自签名的公钥证书配置 IKE	在每个系统上创建并放置两个证书： ■ 自签名证书 ■ 来自远程系统的公钥证书	第 520 页中的 “如何使用自签名的公钥证书配置 IKE”
通过 PKI 证书颁发机构配置 IKE	创建证书请求，然后在每个系统上放置三个证书： ■ 证书颁发机构 (Certificate Authority, CA) 根据您的请求创建的证书 ■ 来自 CA 的公钥证书 ■ 来自 CA 的 CRL	第 525 页中的 “如何使用 CA 签名的证书配置 IKE”
在本地硬件上配置公钥证书	涉及以下操作之一： ■ 在本地硬件上生成自签名证书，然后将公钥从远程系统添加到硬件 ■ 在本地硬件上生成证书请求，然后将公钥证书从 CA 添加到硬件	第 530 页中的 “如何在硬件中生成和存储公钥证书”
更新来自 PKI 的证书撤销列表 (Certificate Revocation List, CRL)	从中心分发点访问 CRL。	第 534 页中的 “如何处理证书撤销列表”

使用公钥证书配置 IKE

使用公钥证书，通信系统就无需在带外共享秘密的加密材料。与预先共享的密钥不同，公钥证书可以在移动计算机或可能重新编号的系统上使用。

公钥证书也可以存储在连接的硬件上。有关过程，请参见第 543 页中的 “将 IKE 配置为查找连接的硬件”。

▼ 如何使用自签名的公钥证书配置 IKE

在此过程中，将创建证书对。私钥存储于本地证书数据库中的磁盘上，可以使用 `certlocal` 子命令进行引用。证书对的公共部分存储于公共证书数据库中，可以使用 `certdb` 子命令进行引用。您将与对等方系统交换该公共部分。两个证书的组合用于验证 IKE 传输。

自签名证书比 CA 颁发的公共证书所需的开销少，但不太易于扩展。与 CA 颁发的证书不同，自签名证书必须在带外进行验证。

1 **在系统控制台上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2 在 `ike.privatekeys` 数据库中创建自签名证书。

```
# ikecert certlocal -ks|-kc -m keysize -t keytype \
-D dname -A altname \
[-S validity-start-time] [-F validity-end-time] [-T token-ID]
```

`-ks`	创建自签名证书。
`-kc`	创建证书请求。有关过程，请参见第 525 页中的 “如何使用 CA 签名的证书配置 IKE”。
`-m` *keysize*	是密钥的大小。*keysize* 可以是 512、1024、2048、3072 或 4096。
`-t` *keytype*	指定要使用的算法类型。*keytype* 可以是 `rsa-sha1`、`rsa-md5` 或 `dsa-sha1`。
`-D` *dname*	是证书主题的 X.509 标识名。*dname* 通常具有以下格式：C=country, O=organization, OU=organizational unit, CN=common name。有效标记是 C、O、OU 和 CN。
`-A` *altname*	是证书的替代名称。*altname* 的形式为 `tag=value`。有效标记是 `IP`、`DNS`、`email` 和 `DN`。
`-S` *validity-start-time*	为证书提供绝对或相对有效开始时间。
`-F` *validity-end-time*	为证书提供绝对或相对有效结束时间。
`-T` *token-ID*	启用 PKCS #11 硬件标记来生成密钥。然后证书将被存储在硬件中。

a. 例如，`partym` 系统上命令的显示与以下信息类似：

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=US, O=PartyCo, OU=US-Partym, CN=Partym" \
-A IP=192.168.13.213
Creating software private keys.
  Writing private key to file /etc/inet/secret/ike.privatekeys/0.
Enabling external key providers - done.
Acquiring private keys for signing - done.
Certificate:
 Proceeding with the signing operation.
 Certificate generated successfully (.../publickeys/0)
Finished successfully.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIICLTCCAZagAwIBAgIBATANBgkqhkiG9w0BAQQFADBNMQswCQYDVQQGEwJVUzEX
...
6sKTxpg4GP3GkQGcd0r1rhW/3yaWBkDwOdFCqEUyffzU
-----END X509 CERTIFICATE-----
```

注 – -D 和 -A 选项的值是任意的。这些值仅用于标识证书。它们不用于标识系统，例如 192.168.13.213。事实上，由于这些值具有特异性，必须在带外验证对等方系统上是否安装了正确的证书。

b. enigma 系统上命令的显示与以下信息类似：

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax" \
-A IP=192.168.116.16
Creating software private keys.
  ...
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIICKDCCAZGgAwIBAgIBATANBgkqhkiG9w0BAQQFADBJMQswCQYDVQQGEwJVUzEV
...
jpxfLM98xyFVyLCbkr3dZ3Tvxvi732BXePKF2A==
-----END X509 CERTIFICATE-----
```

3 保存证书并将它发送到远程系统。

可以将证书粘贴到电子邮件中。

输出为证书的公共部分的编码版本。您可以安全地将此证书粘贴到电子邮件中。接收方必须在带外验证其是否安装了正确的证书，如步骤 5 中所述。

a. 例如，将以下 partym 证书的公共部分发送给 enigma 管理员：

```
To: admin@ja.enigmaexample.com
From: admin@us.partyexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MIICLTCCAZagAwIBAgIBATANBgkqhkiG9w0BAQQFADBNMQswCQYDVQQGEwJVUzEX
...
6sKTxpg4GP3GkQGcd0r1rhW/3yaWBkDwOdFCqEUyffzU
-----END X509 CERTIFICATE-----
```

b. enigma 管理员将向您发送 enigma 证书的公共部分：

```
To: admin@us.partyexample.com
From: admin@ja.enigmaexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MIICKDCCAZGgAwIBAgIBATANBgkqhkiG9w0BAQQFADBJMQswCQYDVQQGEwJVUzEV
...
jpxfLM98xyFVyLCbkr3dZ3Tvxvi732BXePKF2A==
-----END X509 CERTIFICATE-----
```

4 在每个系统上，添加收到的证书。

a. 从管理员的电子邮件中复制公钥。

b. 键入 ikecert certdb -a 命令，然后按回车键。

按回车键时，不显示任何提示。

```
# ikecert certdb -a       Press the Return key
```

c. 粘贴公钥。然后按回车键。要结束输入，请按 Ctrl-D 组合键。

```
-----BEGIN X509 CERTIFICATE-----
MIIC...
...
----END X509 CERTIFICATE-----       Press the Return key
<Control>-D
```

5 向其他管理员核实证书是否来自该管理员。

例如，可以致电其他管理员，以验证您拥有的其公共证书的散列是否与仅他们拥有的其私钥证书的散列匹配。

a. 列出 partym 中存储的证书。

在以下示例中，Note 1 指示了 slot 0 中证书的标识名 (Distinguished Name, DN)。slot 0 中的私钥具有相同的散列（请参见 Note 3），因此这些证书具有相同的证书对。要使用公共证书，必须具有匹对的证书对。certdb 子命令可列出公共部分，而 certlocal 子命令可列出私密部分。

```
partym # ikecert certdb -l
Certificate Slot Name: 0   Type: rsa-sha1
    Subject Name: <C=US, O=PartyCo, OU=US-Partym, CN=Partym>     Note 1
    Key Size: 1024
    Public key hash: 2239A6A127F88EE0CB40F7C24A65B818

Certificate Slot Name: 1   Type: rsa-sha1
    (Private key in certlocal slot 0)
    Subject Name: <C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax>
    Key Size: 1024
    Public key hash: B2BD13FCE95FD27ECE6D2DCD0DE760E2
partym # ikecert certlocal -l
Local ID Slot Name: 0   Key Type: rsa-sha1
    Key Size: 1024
    Public key hash: 2239A6A127F88EE0CB40F7C24A65B818     Note 3

Local ID Slot Name: 1   Key Type: rsa-sha1
        Key Size: 1024
        Public key hash: FEA65C5387BBF3B2C8F16C019FEBC388
...
```

此检查操作已验证 partym 系统具有有效的证书对。

b. 验证 enigma 系统是否具有 partym 的公共证书。

您可通过电话读取公钥散列。

将上一步骤中 partym 上 Note 3 的散列与 enigma 上 Note 4 的散列进行比较。

```
enigma # ikecert certdb -l
Certificate Slot Name: 4   Type: rsa-sha1
    Subject Name: <C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax>
    Key Size: 1024
    Public key hash: DF3F108F6AC669C88C6BD026B0FCE3A0

Certificate Slot Name: 5   Type: rsa-sha1
    Subject Name: <C=US, O=PartyCo, OU=US-Partym, CN=Partym>
    Key Size: 1024
    Public key hash: 2239A6A127F88EE0CB40F7C24A65B818     Note 4
```

存储于 enigma 的公共证书数据库中的最后一个证书的公钥散列和主题名称与上一步骤中 partym 的私密证书匹配。

6 在每个系统上，同时信任这两个证书。

编辑 /etc/inet/ike/config 文件以识别证书。

远程系统的管理员提供 cert_trust、remote_addr 和 remote_id 参数的值。

a. 例如，在 partym 系统上，ike/config 文件的显示与以下信息类似：

```
# Explicitly trust the self-signed certs
# that we verified out of band. The local certificate
# is implicitly trusted because we have access to the private key.
cert_trust "192.168.116.16"      Remote system's certificate Subject Alt Name

## Parameters that may also show up in rules.

p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 5

{
 label "US-partym to JA-enigmax"
 local_id_type dn
 local_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"
 remote_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"

 local_addr  192.168.13.213

# We could explicitly enter the peerâs IP address here, but we donât need
# to do this with certificates, so use a wildcard address. The wildcard
# allows the remote device to be mobile or behind a NAT box.
# remote_addr 192.168.116.16
 remote_addr 0.0.0.0/0

 p1_xform
   { auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes }
}
```

b. 在 enigma 系统上，在 ike/config 文件中添加本地参数的 enigma 值。

对于远程参数，请使用 partym 值。确保本地系统上 label 关键字的值是唯一的。

```
...
{
 label "JA-enigmax to US-partym"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 0.0.0.0/0
...
```

示例 23–4 指定证书的开始时间和结束时间

在此示例中，由 partym 系统上的管理员建立证书有效日期。该证书可回溯 2 1/2 天，自创建之日起 4 年零 6 个月内有效。

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
-A IP=192.168.13.213 \
-S -2d12h -F +4y6m
```

由 enigma 系统上的管理员建立证书有效日期。该证书可回溯两天，在 2010 年 12 月 31 日午夜前有效。

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax" \
-A IP=192.168.116.16 \
-S -2d -F "12/31/2010 12:00 AM"
```

▼ 如何使用 CA 签名的证书配置 IKE

证书颁发机构 (Certificate Authority, CA) 颁发的公共证书需要与外部组织进行协商。证书很容易扩展为保护大量通信系统。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

---

2 **使用 `ikecert certlocal -kc` 命令创建证书请求。**

有关该命令的参数的说明，请参见第 520 页中的 “如何使用自签名的公钥证书配置 IKE”中的步骤 2。

```
# ikecert certlocal -kc -m keysize -t keytype \
-D dname -A altname
```

a. **例如，以下命令在 `partym` 系统上创建证书请求：**

```
# ikecert certlocal -kc -m 1024 -t rsa-sha1 \
> -D "C=US, O=PartyCompany\, Inc., OU=US-Partym, CN=Partym" \
> -A "DN=C=US, O=PartyCompany\, Inc., OU=US-Partym"
Creating software private keys.
  Writing private key to file /etc/inet/secret/ike.privatekeys/2.
Enabling external key providers - done.
Certificate Request:
  Proceeding with the signing operation.
  Certificate request generated successfully (.../publickeys/0)
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIByjCCATMCAQAwUzELMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGVDb21w
...
lcM+tw0ThRrfuJX9t/Qa1R/KxRlMA3zckO80mO9X
-----END CERTIFICATE REQUEST-----
```

b. **以下命令在 `enigma` 系统上创建证书请求：**

```
# ikecert certlocal -kc -m 1024 -t rsa-sha1 \
> -D "C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax, CN=Enigmax" \
> -A "DN=C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax"
Creating software private keys.
...
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
...
8qlqdjaStLGfhDOO
-----END CERTIFICATE REQUEST-----
```

3 **将证书请求提交到 PKI 组织。**

PKI 组织可以告诉您如何提交证书请求。大多数组织具有包含提交表单的 Web 站点。该表单要求证明提交是合法的。通常，将证书请求粘贴到表单中。组织在检查您的请求后，将向您发出以下两个证书对象和已撤销证书的列表：

- 公钥证书－此证书基于您提交给组织的请求。所提交的请求是此公钥证书的一部分。证书可对您进行唯一标识。
- 证书颁发机构－组织的签名。CA 检验公钥证书是否合法。
- 证书撤销列表 (Certificate Revocation List, CRL)－组织已撤销的证书的最新列表。如果在公钥证书中嵌入对 CRL 的访问，则不会将 CRL 作为证书对象单独发送。

  在公钥证书中嵌入 CRL 的 URI 时，IKE 可以自动检索 CRL。同样，在公钥证书中嵌入 DN（LDAP 服务器上的目录名称）项时，IKE 可以从指定的 LDAP 服务器检索并高速缓存 CRL。

有关公钥证书中的嵌入式 URI 和嵌入式 DN 项的示例，请参见第 534 页中的 “如何处理证书撤销列表”。

4 **将每个证书添加到系统。**

`ikecert certdb -a` 的 `-a` 选项将已粘贴的对象添加到系统上的适当证书数据库。有关更多信息，请参见第 506 页中的 “IKE，使用公钥证书”。

a. **在系统控制台上，承担主管理员角色或成为超级用户。**

b. **添加从 PKI 组织收到的公钥证书。**

```
# ikecert certdb -a
      Press the Return key
      Paste the certificate:
-----BEGIN X509 CERTIFICATE-----
...
-----END X509 CERTIFICATE----
      Press the Return key
<Control>-D
```

c. **添加来自 PKI 组织的 CA。**

```
# ikecert certdb -a
      Press the Return key
      Paste the CA:
-----BEGIN X509 CERTIFICATE-----
...
-----END X509 CERTIFICATE----
      Press the Return key
<Control>-D
```

d. **如果 PKI 组织已发送撤销证书列表，则将 CRL 添加到 `certrldb` 数据库：**

```
# ikecert certrldb -a
      Press the Return key
      Paste the CRL:
-----BEGIN CRL-----
...
-----END CRL----
      Press the Return key
<Control>-D
```

5 **在 `/etc/inet/ike/config` 文件中使用 `cert_root` 关键字标识 PKI 组织。**

使用 PKI 组织提供的名称。

a. **例如，`partym` 系统上 `ike/config` 文件的显示可能与以下信息类似：**

```
# Trusted root cert
# This certificate is from Example PKI
```

```
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

## Parameters that may also show up in rules.

p1_xform
  { auth_method rsa_sig oakley_group 1 auth_alg sha1 encr_alg 3des }
p2_pfs 2

{
 label "US-partym to JA-enigmax - Example PKI"
 local_id_type dn
 local_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"
 remote_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"

 local_addr  192.168.13.213
 remote_addr 192.168.116.16

 p1_xform
   { auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes }
}
```

注 – auth_method 参数的所有变量都必须在同一行上。

b. 在 enigma 系统上，创建一个类似的文件。

具体而言，enigma 的 ike/config 文件应该满足以下要求：

- 包括相同的 cert_root 值。
- 对于本地参数，使用 enigma 值。
- 对于远程参数，使用 partym 值。
- 为 label 关键字创建唯一值。此值必须与远程系统的 label 值不同。

```
...
cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"
...
{
 label "JA-enigmax to US-partym - Example PKI"
 local_id_type dn
 local_id   "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213
...
```

6 **通知 IKE 如何处理 CRL。**

选择适当的选项：

- **未提供 CRL**

  如果 PKI 组织未提供 CRL，则将关键字 `ignore_crls` 添加到 `ike/config` 文件。

```
# Trusted root cert
...
cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example,...
ignore_crls
...
```

  `ignore_crls` 关键字指示 IKE 不搜索 CRL。

- **提供了 CRL**

  如果 PKI 组织提供了 CRL 的中心分发点，则可以修改 `ike/config` 文件以指向该位置。

  有关示例，请参见第 534 页中的 “如何处理证书撤销列表”。

示例 23–5 配置 IKE 时使用 rsa_encrypt

在 `ike/config` 文件中使用 `auth_method rsa_encrypt` 时，必须将对等方的证书添加到 `publickeys` 数据库。

1. 将证书发送给远程系统的管理员。

   可以将证书粘贴到电子邮件中。

   例如，`partym` 管理员将发送以下电子邮件：

```
To: admin@ja.enigmaexample.com
From: admin@us.partyexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MII...
----END X509 CERTIFICATE-----
```

   `enigma` 管理员将发送以下电子邮件：

```
To: admin@us.partyexample.com
From: admin@ja.enigmaexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MII
...
-----END X509 CERTIFICATE-----
```

2. 在每个系统上，将通过电子邮件发送的证书添加到本地 `publickeys` 数据库。

```
# ikecert certdb -a
     Press the Return key
-----BEGIN X509 CERTIFICATE-----
MII...
-----END X509 CERTIFICATE-----
```

Press the Return key
<Control>-D

RSA 加密的验证方法可防止窃听者知道 IKE 中的标识。由于 rsa_encrypt 方法隐藏对等方的标识，IKE 无法检索对等方的证书。因此，rsa_encrypt 方法要求 IKE 对等方知道彼此的公钥。

所以，在 /etc/inet/ike/config 文件中使用 rsa_encrypt 的 auth_method 时，必须将对等方的证书添加到 publickeys 数据库。添加证书后，publickeys 数据库包含每对通信系统的三个证书：

- 您的公钥证书
- CA 证书
- 对等方的公钥证书

故障排除－IKE 有效负荷（它包括这三个证书）可能变得过大而无法由 rsa_encrypt 加密。诸如 "authorization failed"（授权失败）和 "malformed payload"（有效负荷格式错误）之类的错误，可以指明 rsa_encrypt 方法无法对总有效负荷进行加密。使用仅需要两个证书的方法（如 rsa_sig）来减小有效负荷的大小。

▼ 如何在硬件中生成和存储公钥证书

在硬件上生成和存储公钥证书，与在系统上生成和存储公钥证书类似。在硬件上，ikecert certlocal 和 ikecert certdb 命令必须标识硬件。带有标记 ID 的 -T 选项向命令标识硬件。

开始之前
- 必须配置硬件。
- 该硬件使用 /usr/lib/libpkcs11.so 库，除非 /etc/inet/ike/config 文件中的 pkcs11_path 关键字指向其他库。该库必须按照以下标准实现：RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)，即 PKCS #11 库。

  有关设置说明，请参见第 544 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 4000 板”。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

---

2 **生成自签名证书或证书请求，并指定标记 ID。**

选择以下选项之一：

注 – 对于 RSA，Sun Crypto Accelerator 4000 和 Sun Crypto Accelerator 6000 板最多支持 2048 位的密钥。对于 DSA，它们最多支持 1024 位的密钥。

- **对于自签名证书，请使用此语法。**

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

`-T` 选项的参数是来自已连接板的标记 ID。

- **对于证书请求，请使用此语法。**

```
# ikecert certlocal -kc -m 1024 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

有关 `ikecert` 命令的参数的说明，请参见 `ikecert(1M)` 手册页。

3 **在系统提示输入 PIN 时，键入板用户、冒号和该用户的口令。**

如果板具有口令为 `rgm4tigt` 的用户 `ikemgr`，应键入以下内容：

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
```

注 – PIN 响应**以明文形式**存储在磁盘上。

键入口令后，将输出证书内容：

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
-----BEGIN X509 CERTIFICATE-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
...
oKUDBbZ9O/pLWYGr
-----END X509 CERTIFICATE-----
```

4 **发送您的证书以供对方使用。**

选择以下选项之一：

- **将自签名证书发送到远程系统。**

  可以将证书粘贴到电子邮件中。

- **将证书请求发送到处理 PKI 的组织。**

  按照 PKI 组织的说明提交证书请求。有关更详细的论述，请参见第 525 页中的 “如何使用 CA 签名的证书配置 IKE”中的步骤 3。

5 **在系统上，编辑 `/etc/inet/ike/config` 文件以识别这些证书。**

选择以下选项之一。

- **自签名证书**

  使用远程系统管理员为 `cert_trust`、`remote_id` 和 `remote_addr` 参数提供的值。例如，在 `enigma` 系统上，`ike/config` 文件的显示与以下信息类似：

```
# Explicitly trust the following self-signed certs
# Use the Subject Alternate Name to identify the cert

cert_trust "192.168.116.16"       Local system's certificate Subject Alt Name
cert_trust "192.168.13.213"       Remote system's certificate Subject Alt name


# Solaris 10 1/06 release: default path does not have to be typed in #pkcs11_path
"/usr/lib/libpkcs11.so"      Hardware connection

# Solaris 10 release: use this path
#pkcs11_path "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
...
{
 label "JA-enigmax to US-partym"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213

 p1_xform
  {auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes}
}
```

- **证书请求**

  将 PKI 组织提供的名称作为 `cert_root` 关键字的值键入。例如，`enigma` 系统上 `ike/config` 文件的显示可能与以下信息类似：

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

# Solaris 10 1/06 release: default path does not have to be typed in #pkcs11_path
```

```
"/usr/lib/libpkcs11.so"       Hardware connection

# Solaris 10 release: use this path
#pkcs11_path "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
...
{
 label "JA-enigmax to US-partym - Example PKI"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213

 p1_xform
  {auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes}
}
```

6 **在硬件中存放来自对方的证书。**

按照在步骤 3 中作出的响应，响应 PIN 请求。

注 – 必须将公钥证书添加到生成私钥的那个连接硬件上。

- **自签名证书。**

  添加远程系统的自签名证书。在此示例中，证书存储在 `DCA.ACCEL.STOR.CERT` 文件中。

  ```
  # ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
  Enter PIN for PKCS#11 token:      Type user:password
  ```

  如果自签名证书将 `rsa_encrypt` 用作 `auth_method` 参数的值，则将对等方的证书添加到硬件存储。

- **来自 PKI 组织的证书。**

  添加组织从证书请求生成的证书，然后添加证书颁发机构 (Certificate Authority, CA)。

  ```
  # ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
  Enter PIN for PKCS#11 token:      Type user:password

  # ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CA.CERT
  Enter PIN for PKCS#11 token:      Type user:password
  ```

  要添加来自 PKI 组织的证书撤销列表 (Certificate Revocation List, CRL)，请参见第 534 页中的 “如何处理证书撤销列表”。

▼ 如何处理证书撤销列表

证书撤销列表 (Certificate Revocation List, CRL) 包含来自证书颁发机构的过时证书或已损坏证书。CRL 的处理方式有四种。

- 如果 CA 组织未发出 CRL，则您必须指示 IKE 忽略 CRL。步骤 6 in 第 525 页中的 “如何使用 CA 签名的证书配置 IKE”中介绍了此选项。
- 可以指示 IKE 从一个 URI（Uniform Resource Indicator，统一资源指示符）访问 CRL，该 URI 的地址嵌入到来自 CA 的公钥证书中。
- 可以指示 IKE 从 LDAP 服务器访问 CRL，该服务器的 DN（Directory Name，目录名称）项嵌入到来自 CA 的公钥证书中。
- 可以将 CRL 作为 `ikecert certrldb` 命令的参数提供。有关示例，请参见示例 23–6。

以下过程介绍如何指示 IKE 从中心分发点使用 CRL。

1 显示从 CA 收到的证书。

```
# ikecert certdb -lv certspec
```

`-l` 列出 IKE 证书数据库中的证书。

`-v` 以详细模式列出证书。应谨慎使用此选项。

certspec 是一种与 IKE 证书数据库中的证书匹配的模式。

例如，以下证书由 Oracle 颁发。详细信息已更改。

```
# ikecert certdb -lv example-protect.oracle.com
Certificate Slot Name: 0   Type: dsa-sha1
   (Private key in certlocal slot 0)
 Subject Name: <O=Oracle, CN=example-protect.oracle.com>
 Issuer Name: <CN=Oracle CA (Cl B), O=Oracle>
 SerialNumber: 14000D93
   Validity:
      Not Valid Before: 2002 Jul 19th, 21:11:11 GMT
      Not Valid After:  2005 Jul 18th, 21:11:11 GMT
   Public Key Info:
      Public Modulus  (n) (2048 bits): C575A...A5
      Public Exponent (e) (  24 bits): 010001
   Extensions:
      Subject Alternative Names:
              DNS = example-protect.oracle.com
      Key Usage: DigitalSignature KeyEncipherment
      [CRITICAL]
   CRL Distribution Points:
      Full Name:
         URI = #Ihttp://www.oracle.com/pki/pkismica.crl#i
         DN = <CN= Oracle CA (Cl B), O=Oracle>
      CRL Issuer:
      Authority Key ID:
      Key ID:              4F ... 6B
      SubjectKeyID:        A5 ... FD
```

```
    Certificate Policies
    Authority Information Access
```

请注意 CRL Distribution Points 项。URI 项指示此组织的 CRL 在 Web 上是可用的。DN 项指示 CRL 在 LDAP 服务器上是可用的。在 IKE 访问 CRL 后，将高速缓存该 CRL 以供将来使用。

要访问 CRL，您需要到达分发点。

2 选择以下方法之一从中心分发点访问 CRL。

- **使用 URI。**

  将关键字 use_http 添加到主机的 /etc/inet/ike/config 文件。例如，ike/config 文件的显示与以下信息类似：

  ```
  # Use CRL from organization's URI
  use_http
  ...
  ```

- **使用 Web 代理。**

  将关键字 proxy 添加到 ike/config 文件。proxy 关键字将 URL 用作参数，如下所示：

  ```
  # Use own web proxy
  proxy "http://proxy1:8080"
  ```

- **使用 LDAP 服务器。**

  在主机的 /etc/inet/ike/config 文件中，将 LDAP 服务器指定为 ldap-list 关键字的参数。您的组织提供 LDAP 服务器的名称。ike/config 文件中项的显示与以下信息类似：

  ```
  # Use CRL from organization's LDAP
  ldap-list "ldap1.oracle.com:389,ldap2.oracle.com"
  ...
  ```

在证书到期之前，IKE 检索并高速缓存 CRL。

示例 23–6 将 CRL 粘贴到本地 certrldb 数据库中

如果无法从中心分发点获取 PKI 组织的 CRL，则可以将该 CRL 手动添加到本地 certrldb 数据库。按照 PKI 组织的说明将 CRL 提取到文件中，然后使用 ikecert certrldb -a 命令将此 CRL 添加到数据库。

```
# ikecert certrldb -a < Oracle.Cert.CRL
```

为移动系统配置 IKE（任务列表）

下表包含将 IKE 配置为处理远程登录到中心站点的系统的过程的链接。

任务	说明	参考
从站点外与中心站点进行通信	允许站点外系统与中心站点进行通信。站点外系统可能是移动系统。	第 536 页中的 “如何为站点外系统配置 IKE”
在接受来自移动系统的通信流量的中心系统上使用 CA 的公共证书和 IKE	将网关系统配置为接受来自没有固定 IP 地址的系统的 IPsec 流量。	示例 23–7
在没有固定 IP 地址的系统上使用 CA 的公共证书和 IKE	将移动系统配置为保护它传输到中心站点（如公司总部）的流量。	示例 23–8
在接受来自移动系统的流量的中心系统上使用自签名证书和 IKE	使用自签名证书配置网关系统，以接受来自移动系统的 IPsec 流量。	示例 23–9
在没有固定 IP 地址的系统上使用自签名证书和 IKE	使用自签名证书配置移动系统，以保护它传输到中心站点的流量。	示例 23–10

为移动系统配置 IKE

在进行适当配置后，家庭办公室和手提电脑可以使用 IPsec 和 IKE 与其公司的中央计算机进行通信。利用与公钥证书验证方法组合的综合 IPsec 策略，离站系统可以保护它们传输到中心系统的流量。

▼ 如何为站点外系统配置 IKE

IPsec 和 IKE 要求用唯一 ID 标识源和目标。对于没有唯一 IP 地址的站点外系统或移动系统，必须使用其他 ID 类型。可以使用诸如 `DNS`、`DN` 或 `email` 之类的 ID 类型唯一地标识系统。

对于具有唯一 IP 地址的站点外系统或移动系统，最好也应使用其他 ID 类型进行配置。例如，如果系统尝试从 NAT 盒 (NAT box) 之后连接到中心站点，则不会使用它们的唯一地址。NAT 盒 (NAT box) 指定一个中心系统无法识别的任意 IP 地址。

预先共享的密钥也不太适合用作移动系统的验证机制，因为预先共享的密钥需要固定的 IP 地址。使用自签名证书或来自 PKI 的证书，移动系统可以与中心站点进行通信。

1. **在系统控制台上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2 将中心系统配置为识别移动系统。

a. 设置 `ipsecinit.conf` 文件。

中心系统需要一个允许很宽的 IP 地址范围的策略。随后，IKE 策略中的证书确保进行连接的系统是合法的。

```
# /etc/inet/ipsecinit.conf on central
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. 设置 `ike.config` 文件。

DNS 标识中心系统。证书用于验证该系统。

```
## /etc/inet/ike/ike.config on central
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://somecache.domain:port/"
#
# Use LDAP server
ldap_server    "ldap-server1.domain.org,ldap2.domain.org:port"
#
# List CA-signed certificates
cert_root     "C=US, O=Domain  Org, CN=Domain STATE"
#
# List self-signed certificates - trust server and enumerated others
#cert_trust     "DNS=central.domain.org"
#cert_trust     "DNS=mobile.domain.org"
#cert_trust     "DN=CN=Domain Org STATE (CLASS), O=Domain Org
#cert_trust     "email=root@central.domain.org"
#cert_trust     "email=user1@mobile.domain.org"
#

# Rule for mobile systems with certificate
{
  label "Mobile systems with certificate"
  local_id_type DNS

# CA's public certificate ensures trust,
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

3 登录到每个移动系统，然后将该系统配置为查找中心系统。

a. 设置 /etc/hosts 文件。

/etc/hosts 文件不需要移动系统的地址，但是可以提供一个地址。该文件必须包含中心系统的公共 IP 地址。

```
# /etc/hosts on mobile
central 192.xxx.xxx.x
```

b. 设置 ipsecinit.conf 文件。

移动系统需要按照中心系统的公共 IP 地址来查找中心系统。这些系统必须配置相同的 IPsec 策略。

```
# /etc/inet/ipsecinit.conf on mobile
# Find central
{raddr 192.xxx.xxx.x} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

c. 设置 ike.config 文件。

标识符不能是 IP 地址。以下标识符对移动系统有效：

- DN=*ldap-directory-name*
- DNS=*domain-name-server-address*
- email=*email-address*

证书用于验证移动系统。

```
## /etc/inet/ike/ike.config on mobile
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://somecache.domain:port/"
#
# Use LDAP server
ldap_server   "ldap-server1.domain.org,ldap2.domain.org:port"
#
# List CA-signed certificates
cert_root    "C=US, O=Domain Org, CN=Domain STATE"
#
# Self-signed certificates - trust me and enumerated others
#cert_trust    "DNS=mobile.domain.org"
#cert_trust    "DNS=central.domain.org"
#cert_trust    "DN=CN=Domain Org STATE (CLASS), O=Domain Org
#cert_trust    "email=user1@domain.org"
#cert_trust    "email=root@central.domain.org"
#
# Rule for off-site systems with root certificate
{
    label "Off-site mobile with certificate"
    local_id_type DNS
```

```
# NAT-T can translate local_addr into any public IP address
# central knows me by my DNS

    local_id "mobile.domain.org"
    local_addr 0.0.0.0/0

# Find central and trust the root certificate
    remote_id "central.domain.org"
    remote_addr 192.xxx.xxx.x

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

4 **将 IKE 配置读入内核。**

- **从 Solaris 10 4/09 发行版开始，请启用 `ike` 服务。**

  ```
  # svcadm enable svc:/network/ipsec/ike
  ```

- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请重新引导系统。**

  ```
  # init 6
  ```

  或者，先停止然后再启动 `in.iked` 守护进程。

示例 23–7 将中心计算机配置为接受来自移动系统的 IPsec 流量

IKE 可以从 NAT 盒 (NAT box) 之后启动协商。但是，IKE 的理想设置是在 NAT 盒 (NAT box) 没有介入的情况下进行的。在以下示例中，CA 的公共证书放置在移动系统和中心系统上。中心系统接受来自 NAT 盒 (NAT box) 之后的系统的 IPsec 协商。`main1` 是可以接受来自站点外系统的连接的公司系统。有关如何设置站点外系统，请参见示例 23–8。

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
```

```
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
  label "Off-site system with root certificate"
  local_id_type DNS
  local_id "main1.domain.org"
  local_addr 192.168.0.100

# CA's public certificate ensures trust,
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha1}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha1}
}
```

示例 23–8 使用 IPsec 配置 NAT 之后的系统

在以下示例中，CA 的公共证书放置在移动系统和中心系统上。`mobile1` 将从本部连接到公司总部。Internet 服务提供商 (Internet Service Provider, ISP) 网络使用 NAT 盒 (NAT box)，以允许 ISP 为 `mobile1` 指定专用地址。然后，NAT 盒 (NAT box) 将专用地址转换为与其他 ISP 网络节点共享的公共 IP 地址。公司总部不在 NAT 之后。有关如何在公司总部设置计算机，请参见示例 23–7。

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
```

```
# Use LDAP server
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
  label "Off-site mobile1 with root certificate"
  local_id_type DNS
  local_id "mobile1.domain.org"
  local_addr 0.0.0.0/0

# Find main1 and trust the root certificate
  remote_id "main1.domain.org"
  remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

示例 23–9 接受来自移动系统的自签名证书

在以下示例中，自签名证书已经颁发，并存放在移动系统和中心系统上。`main1` 是可以接受来自站点外系统的连接的公司系统。有关如何设置站点外系统，请参见示例 23–10。

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Self-signed certificates - trust me and enumerated others
cert_trust    "DNS=main1.domain.org"
cert_trust    "jdoe@domain.org"
cert_trust    "user2@domain.org"
cert_trust    "user3@domain.org"
#
# Rule for off-site systems with trusted certificate
{
  label "Off-site systems with trusted certificates"
  local_id_type DNS
  local_id "main1.domain.org"
  local_addr 192.168.0.100

# Trust the self-signed certificates
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0
```

```
p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

示例 23–10 使用自签名证书联系中心系统

在以下示例中，mobile1 将从本部连接到公司总部。证书已经颁发，并放置在移动系统和中心系统上。ISP 网络使用 NAT 盒 (NAT box)，以允许 ISP 为 mobile1 指定专用地址。然后，NAT 盒 (NAT box) 将专用地址转换为与其他 ISP 网络节点共享的公共 IP 地址。公司总部不在 NAT 之后。有关如何在公司总部设置计算机，请参见示例 23–9。

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters

# Self-signed certificates - trust me and the central system
cert_trust    "jdoe@domain.org"
cert_trust    "DNS=main1.domain.org"
#
# Rule for off-site systems with trusted certificate
{
  label "Off-site mobile1 with trusted certificate"
  local_id_type email
  local_id "jdoe@domain.org"
  local_addr 0.0.0.0/0

# Find main1 and trust the certificate
  remote_id "main1.domain.org"
  remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

将 IKE 配置为查找连接的硬件（任务列表）

下表包含将已连接硬件的相关信息通知给 IKE 的过程的链接。只有将已连接硬件的相关信息通知 IKE，IKE 才能使用该硬件。要使用硬件，请按照第 520 页中的 “使用公钥证书配置 IKE”中的硬件过程操作。

注 – 您不必通知 IKE 有关片内 (on-chip) 硬件的信息。例如，UltraSPARC T2 处理器提供加密加速。您不需要配置 IKE 来查找片内加速器。

任务	说明	参考
将 IKE 密钥操作转移到 Sun Crypto Accelerator 1000 板	将 IKE 链接到 PKCS #11 库。	第 543 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 1000 板”
将 IKE 密钥操作转移到 Sun Crypto Accelerator 4000 板并在该板上存储密钥	将 IKE 链接到 PKCS #11 库，并列出已连接硬件的名称。	第 544 页中的 “如何将 IKE 配置为查找 Sun Crypto Accelerator 4000 板”

将 IKE 配置为查找连接的硬件

公钥证书也可以存储在连接的硬件上。Sun Crypto Accelerator 1000 板仅提供存储。Sun Crypto Accelerator 4000 和 Sun Crypto Accelerator 6000 板提供存储空间，并允许将公钥操作从系统转移到板上。

▼ 如何将 IKE 配置为查找 Sun Crypto Accelerator 1000 板

开始之前 以下过程假定 Sun Crypto Accelerator 1000 板已连接到系统。此过程还假定已安装板的软件，而且已配置该软件。有关说明，请参见《Sun Crypto Accelerator 1000 Board Version 2.0 Installation and User's Guide》（《Sun Crypto Accelerator 1000 板 2.0 版安装和用户指南》）(`http://download.oracle.com/docs/cd/E19412-01/819-0425-11/819-0425-11.pdf`)。

1. **在系统控制台上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

2 **检查是否已链接 PKCS #11 库。**

键入以下命令，以确定 PKCS #11 库是否已链接：

```
# ikeadm get stats
Phase 1 SA counts:
Current:   initiator:          0   responder:          0
Total:     initiator:          0   responder:          0
Attempted: initiator:          0   responder:          0
Failed:    initiator:          0   responder:          0
           initiator fails include 0 time-out(s)
PKCS#11 library linked in from /usr/lib/libpkcs11.so
#
```

3 **Solaris 10 1/06： 从此发行版开始，可以在 softtoken 密钥库中存储密钥。**

有关加密框架提供的密钥库的信息，请参见 cryptoadm(1M) 手册页。有关使用密钥库的示例，请参见Example 23–11。

▼ 如何将 IKE 配置为查找 Sun Crypto Accelerator 4000 板

开始之前 以下过程假定 Sun Crypto Accelerator 4000 板已连接到系统。此过程还假定已安装板的软件，而且已配置该软件。有关说明，请参见《Sun Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide》（《Sun Crypto Accelerator 4000 板 1.1 版安装和用户指南》）(http://download.oracle.com/docs/cd/E19877-01/817-3693-10/817-3693-10.pdf)。

1 **在系统控制台上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 ssh 命令进行安全的远程登录。

2 **检查是否已链接 PKCS #11 库。**

IKE 使用该库的例程在 Sun Crypto Accelerator 4000 板上处理密钥生成和密钥存储。键入以下命令，以确定 PKCS #11 库是否已链接：

```
$ ikeadm get stats
...
PKCS#11 library linked in from /usr/lib/libpkcs11.so
$
```

注 – 对于 RSA，Sun Crypto Accelerator 4000 板最多支持 2048 位的密钥。对于 DSA，此板最多支持 1024 位的密钥。

3 查找已连接的 Sun Crypto Accelerator 4000 板的标记 ID。

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                          "
```

该库返回一个包含 32 个字符的标记 ID（也称为 keystore name（密钥库名称））。在此示例中，可以将 `Sun Metaslot` 标记与 `ikecert` 命令一起使用来存储和加速 IKE 密钥。

有关如何使用标记的说明，请参见第 530 页中的 “如何在硬件中生成和存储公钥证书”。

结尾空格是由 `ikecert` 命令自动填充的。

示例 23–11 查找和使用 metaslot 标记

标记可以存储在磁盘上、连接的板上或加密框架提供的 softtoken 密钥库中。softtoken 密钥库标记 ID 可能与以下信息类似。

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                          "
```

有关如何为 softtoken 密钥库创建口令短语，请参见 `pktool(1)` 手册页。

如下所示的命令可向 softtoken 密钥库添加证书。`Sun.Metaslot.cert` 是一个包含 CA 证书的文件。

```
# ikecert certdb -a -T "Sun Metaslot" < Sun.Metaslot.cert
Enter PIN for PKCS#11 token:      Type user:passphrase
```

▼ 如何将 IKE 配置为查找 Sun Crypto Accelerator 6000 板

开始之前 以下过程假定 Sun Crypto Accelerator 6000 板已连接到系统。此过程还假定已安装板的软件，而且已配置该软件。有关说明，请参见《Sun Crypto Accelerator 6000 Board Version 1.1 User's Guide》（《Sun Crypto Accelerator 6000 板 1.1 版用户指南》）(`http://download.oracle.com/docs/cd/E19321-01/820-4144-12/820-4144-12.pdf`)。

1 在系统控制台上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

2 **检查是否已链接 PKCS #11 库。**

IKE 使用该库的例程在 Sun Crypto Accelerator 6000 板上处理密钥生成和密钥存储。键入以下命令，以确定 PKCS #11 库是否已链接：

```
$ ikeadm get stats
...
PKCS#11 library linked in from /usr/lib/libpkcs11.so
$
```

3 **查找已连接的 Sun Crypto Accelerator 6000 板的标记 ID。**

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                    "
```

该库返回一个包含 32 个字符的标记 ID（也称为 keystore name（密钥库名称））。在此示例中，可以将 `Sun Metaslot` 标记与 `ikecert` 命令一起使用来存储和加速 IKE 密钥。

有关如何使用标记的说明，请参见第 530 页中的 “如何在硬件中生成和存储公钥证书”。

结尾空格是由 `ikecert` 命令自动填充的。

示例 23–12 查找和使用 metaslot 标记

标记可以存储在磁盘上、连接的板上或加密框架提供的 softtoken 密钥库中。softtoken 密钥库标记 ID 可能与以下信息类似。

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                    "
```

有关如何为 softtoken 密钥库创建口令短语，请参见 `pktool(1)` 手册页。

如下所示的命令可向 softtoken 密钥库添加证书。`Sun.Metaslot.cert` 是一个包含 CA 证书的文件。

```
# ikecert certdb -a -T "Sun Metaslot" < Sun.Metaslot.cert
Enter PIN for PKCS#11 token:      Type user:passphrase
```

更改 IKE 传输参数（任务列表）

下表包含配置 IKE 传输参数的过程的链接。

任务	说明	参考
使密钥协商的效率更高。	更改密钥协商参数。	第 547 页中的 “如何更改阶段 1 IKE 密钥协商的持续时间”
配置密钥协商以允许传输延迟。	增大密钥协商参数。	示例 23–13
将密钥协商配置为快速成功或快速显示故障。	减小密钥协商参数。	示例 23–14

更改 IKE 传输参数

当 IKE 协商密钥时，传输速度可能会影响协商的成功。通常，无需更改 IKE 传输参数的缺省值。但是，在通过很脏的线路优化密钥协商时，或者再现问题时，您可能希望更改传输值。

在持续时间较长的情况下，IKE 可以通过不可靠的传输线路协商密钥。可以增大某些参数以使初始尝试成功。如果初始尝试未成功，则可以隔开后续尝试以便为成功提供更多时间。

通过缩短持续时间，可以利用可靠的传输线路。这样，可以更快捷地重试已失败的协商，以便加快协商速度。在诊断问题时，您可能还希望加快协商，以便尽早获得失败的结果。缩短持续时间也使阶段 1 SA 可用于其生命周期。

▼ 如何更改阶段 1 IKE 密钥协商的持续时间

1. **在系统控制台上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

   注 – 远程登录会使安全关键型通信易于遭到窃听。即使以某种方式保护远程登录，系统的安全性也会降至远程登录会话的安全性。请使用 `ssh` 命令进行安全的远程登录。

   ---

2. **在每个系统上更改全局传输参数的缺省值。**

   在每个系统上，修改 `/etc/inet/ike/config` 文件中的阶段 1 持续时间参数。

   ```
   ### ike/config file on      system

   ## Global parameters
   ```

```
#
## Phase 1 transform defaults
#
#expire_timer      300
#retry_limit        5
#retry_timer_init    0.5 (integer or float)
#retry_timer_max    30   (integer or float)
```

`expire_timer`	允许尚未完成的 IKE 阶段 1 协商在删除协商尝试之前延迟的秒数。缺省情况下，尝试延迟 30 秒。
`retry_limit`	异常中止任何 IKE 协商之前的重新传输次数。缺省情况下，IKE 尝试五次。
`retry_timer_init`	重新传输之间的初始时间间隔。在达到 `retry_timer_max` 值之前，此时间间隔以双倍递增。初始时间间隔为 0.5 秒。
`retry_timer_max`	重新传输之间的最大时间间隔（以秒为单位）。重新传输时间间隔在达到此限制时停止增加。缺省情况下，该限制为 30 秒。

3 将已更改的配置读入内核。

- **从 Solaris 10 4/09 发行版开始，请刷新 `ike` 服务。**

  ```
  # svcadm refresh svc:/network/ipsec/ike
  ```

- **如果您运行的是 Solaris 10 4/09 发行版之前的发行版，请重新引导系统。**

  ```
  # init 6
  ```

  或者，先停止然后再启动 `in.iked` 守护进程。

示例 23–13 延长 IKE 阶段 1 协商时间

在以下示例中，系统已通过高流量传输线路连接到其 IKE 对等方。原始设置位于文件的注释中。新设置延长了协商时间。

```
### ike/config file on partym
## Global Parameters
#
## Phase 1 transform defaults
#expire_timer   300
#retry_limit      5
#retry_timer_init 0.5 (integer or float)
#retry_timer_max 30   (integer or float)
#
expire_timer  600
retry_limit  10
retry_timer_init  2.5
retry_timer_max  180
```

示例 23–14 缩短 IKE 阶段 1 协商时间

在以下示例中，系统已通过小流量的高速线路连接到其 IKE 对等方。原始设置位于文件的注释中。新设置缩短了协商时间。

```
### ike/config file on partym
## Global Parameters
#
## Phase 1 transform defaults
#expire_timer   300
#retry_limit      5
#retry_timer_init 0.5 (integer or float)
#retry_timer_max 30   (integer or float)
#
expire_timer  120
retry_timer_init  0.20
```

◆ ◆ ◆ 第 24 章

Internet 密钥交换（参考信息）

本章包含有关 IKE 的以下参考信息：

有关实现 IKE 的说明，请参见第 23 章，配置 IKE（任务）。有关概述信息，请参见第 22 章，Internet 密钥交换（概述）。

IKE 服务

svc:/network/ipsec/ike:default **服务** – 服务管理工具 (Service Management Facility, SMF) 提供 ike 服务以管理 IKE。缺省情况下，此服务处于禁用状态。启用此服务之前，必须创建 IKE 配置文件 /etc/inet/ike/config。

以下 ike 服务属性是可配置的：

- config_file **属性** – 为 IKE 配置文件的位置。初始值为 /etc/inet/ike/config。
- debug_level **属性** – 为 in.iked 守护进程的调试级别。初始值为 op 或 operational。有关可能的值，请参见 ikeadm(1M) 手册页中**对象类型**下有关调试级别的表。
- admin_privilege **属性** – 为 in.iked 守护进程的特权级别。初始值为 base。其他值为 modkeys 和 keymat。有关详细信息，请参见第 553 页中的 “ikeadm 命令”。

有关 SMF 的信息，请参见《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。另请参见 smf(5)、svcadm(1M) 和 svccfg(1M) 手册页。

IKE 守护进程

in.iked 守护进程自动管理 Oracle Solaris 系统上 IPsec 的加密密钥。该守护进程与运行相同协议的远程系统协商，以便以受保护方式为安全关联 (Security Association, SA) 提供经过验证的加密材料。必须在计划以安全方式通信的所有系统上运行该守护进程。

缺省情况下，未启用 svc:/network/ipsec/ike:default 服务。配置了 /etc/inet/ike/config 文件并启用 ike 服务后，in.iked 守护进程会在系统引导时运行。

在 IKE 守护进程运行时，系统在阶段 1 交换中向其对等 IKE 实体进行自我验证。与验证方法一样，对等实体也是在 IKE 策略文件中定义的。然后守护进程建立阶段 2 交换的密钥。按照在策略文件中指定的时间间隔，自动刷新 IKE 密钥。in.iked 守护进程通过 PF_KEY 套接字侦听从网络传入的 IKE 请求，并侦听外发通信流量请求。有关更多信息，请参见 pf_key(7P) 手册页。

有两个命令支持 IKE 守护进程。ikeadm 命令可用于查看并临时修改 IKE 策略。要永久修改 IKE 策略，请修改 ike 服务的属性。要修改 IKE 服务的属性，请参见第 460 页中的 “如何管理 IKE 和 IPsec 服务”。

使用 ikecert 命令可以查看和管理公钥数据库。此命令管理本地数据库 ike.privatekeys 和 publickeys。它还管理公钥操作和公钥在硬件上的存储。

IKE 配置文件

IKE 配置文件 /etc/inet/ike/config 管理 IPsec 策略文件 /etc/inet/ipsecinit.conf 中受保护的接口的密钥。

使用 IKE 的密钥管理包括规则和全局参数。IKE 规则标识加密材料保护的系统或网络。该规则还指定验证方法。全局参数包括诸如已连接硬件加速器路径之类的项。有关 IKE 策略文件的示例，请参见第 510 页中的 “使用预先共享的密钥配置 IKE（任务列表）”。有关 IKE 策略项的示例和说明，请参见 ike.config(4) 手册页。

IKE 支持的 IPsec SA 根据 IPsec 配置文件 /etc/inet/ipsecinit.conf 中的策略来保护 IP 数据报。IKE 策略文件确定是否在创建 IPsec SA 时使用完全转发保密 (Perfect Forward Security, PFS)。

/etc/inet/ike/config 文件可以包括按照以下标准实现的库的路径：RSA Security Inc. 推出的 PKCS #11 加密令牌接口 (Cryptographic Token Interface, Cryptoki)。IKE 使用此 PKCS #11 库访问用于密钥加速和密钥存储的硬件。

ike/config 文件的安全注意事项与 ipsecinit.conf 文件的安全注意事项类似。有关详细信息，请参见第 497 页中的 “ipsecinit.conf 和 ipsecconf 的安全注意事项”。

ikeadm 命令

可以使用 ikeadm 命令执行以下操作：

- 查看 IKE 守护进程的各个方面。
- 更改传递到 IKE 守护进程的参数。
- 显示在阶段 1 交换期间有关 SA 创建的统计信息。
- 调试 IKE 进程。
- 查看 IKE 状态的各个方面。
- 更改 IKE 守护进程的属性。
- 显示在阶段 1 交换期间有关 SA 创建的统计信息。
- 调试 IKE 协议交换。

有关此命令的选项的示例和完整说明，请参见 ikeadm(1M) 手册页。

正在运行的 IKE 守护进程的特权级别决定可以查看和修改 IKE 守护进程的哪些方面。可以有三种权限级别。

base 级别　　不能查看或修改加密材料。base 级别是缺省特权级别。

modkeys 级别　　可以删除、更改和添加预先共享的密钥。

keymat 级别　　可以使用 ikeadm 命令查看实际的加密材料。

如果要临时更改特权，可使用 ikeadm 命令。如果要进行永久更改，请更改 ike 服务的 admin_privilege 属性。有关过程，请参见第 460 页中的 “如何管理 IKE 和 IPsec 服务”。

ikeadm 命令的安全注意事项与 ipseckey 命令的安全注意事项类似。有关详细信息，请参见第 499 页中的 “ipseckey 的安全注意事项”。

IKE 预先共享的密钥文件

如果手动创建预先共享的密钥，这些密钥将存储在 /etc/inet/secret 目录下的文件中。ike.preshared 文件包含用于 Internet 安全关联和密钥管理协议 (Internet Security Association and Key Management Protocol, ISAKMP) SA 的预先共享的密钥。ipseckeys 文件包含用于 IPsec SA 的预先共享的密钥。按 0600 保护这些文件。按 0700 保护 secret 目录。

- 将 ike/config 文件配置为需要预先共享的密钥时，您需要创建 ike.preshared 文件。在 ike.preshared 文件中输入 ISAKMP SA 的加密材料（即用于 IKE 验证）。由于预先共享的密钥用于验证阶段 1 交换，因此在 in.iked 守护进程启动之前，该文件必须有效。
- ipseckeys 文件包含 IPsec SA 的加密材料。有关手动管理该文件的示例，请参见第 453 页中的 “如何手动创建 IPsec 安全关联”。IKE 守护进程不使用此文件。IKE 为 IPsec SA 生成的加密材料存储在内核中。

IKE 公钥数据库和命令

ikecert 命令处理本地系统的公钥数据库。在 ike/config 文件需要公钥证书时，可以使用此命令。由于 IKE 使用这些数据库验证阶段 1 交换，因此必须在激活 in.iked 守护进程之前填充这些数据库。以下三个子命令可分别处理三种数据库中的其中一种：certlocal、certdb 和 certrldb。

ikecert 命令还处理密钥存储。密钥可以存储在磁盘上、连接的 Sun Crypto Accelerator 6000 或 Sun Crypto Accelerator 4000 板上或 softtoken 密钥库中。当加密框架中的 metaslot 用于和硬件设备进行通信时，softtoken 密钥库是可用的。ikecert 命令使用 PKCS #11 库定位密钥存储。

- **Solaris 10 1/06**：从此发行版开始，不必指定该库。缺省情况下，PKCS #11 库为 /usr/lib/libpkcs11.so。
- **Solaris 10**：在此发行版中，必须指定 PKCS #11 项。否则，ikecert 命令的 -T 选项无法起作用。该项的显示如下所示：

  ```
  pkcs11_path "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
  ```

有关更多信息，请参见 ikecert(1M) 手册页。有关 metaslot 以及 softtoken 密钥库的信息，请参见 cryptoadm(1M) 手册页。

ikecert tokens 命令

tokens 参数列出可用的标记 ID。使用标记 ID，ikecert certlocal 和 ikecert certdb 命令可以生成公钥证书和证书请求。证书和证书请求也可以由加密框架存储在 softtoken 密钥库中或存储在连接的 Sun Crypto Accelerator 6000 或 Sun Crypto Accelerator 4000 板上。ikecert 命令使用 PKCS #11 库定位证书存储。

ikecert certlocal 命令

certlocal 子命令管理私钥数据库。使用此子命令的选项，可以添加、查看和删除私钥。此子命令还用于创建自签名的证书或证书请求。-ks 选项用于创建自签名的证书。-kc 选项用于创建证书请求。密钥存储在系统的 /etc/inet/secret/ike.privatekeys 目录中，或者通过 -T 选项存储在连接的硬件上。

创建私钥时，ikecert certlocal 命令的选项必须在 ike/config 文件中具有相关项。ikecert 选项和 ike/config 项之间的对应关系如下表所示。

表 24–1 ikecert 选项和 ike/config 项之间的对应关系

ikecert 选项	ike/config 项	说明
-A *subject-alternate-name*	cert_trust *subject-alternate-name*	唯一标识证书的别名。可能的值是 IP 地址、电子邮件地址或域名。
-D *X.509-distinguished-name*	*X.509-distinguished-name*	证书颁发机构的完整名称，包括国家/地区 (C)、组织名称 (ON)、组织单元 (OU) 和公用名称 (CN)。
-t dsa-sha1	auth_method dsa_sig	一种速度比 RSA 稍慢的验证方法。
-t rsa-md5 和 -t rsa-sha1	auth_method rsa_sig	一种速度比 DSA 稍快的验证方法。 RSA 公钥必须大到足以加密最大的 payload（有效负荷）。通常，标识有效负荷（如 X.509 标识名）是最大的有效负荷。
-t rsa-md5 和 -t rsa-sha1	auth_method rsa_encrypt	RSA 加密防止窃听者知道 IKE 中的标识，但是要求 IKE 对等方知道彼此的公钥。
-T	pkcs11_path	PKCS #11 库处理 Sun Crypto Accelerator 1000 板、Sun Crypto Accelerator 6000 板和 Sun Crypto Accelerator 4000 板上的密钥加速。该库还提供标记，用于处理 Sun Crypto Accelerator 6000 和 Sun Crypto Accelerator 4000 板上的密钥存储。

如果使用 ikecert certlocal -kc 命令发出证书请求，则会将该命令的输出发送到 PKI 组织或证书颁发机构 (Certificate Authority, CA)。如果您的公司运行自己的 PKI，则会将输出发送到 PKI 管理员。然后，PKI 组织、CA 或 PKI 管理员将创建证书。PKI 或 CA 返回给您的证书是 certdb 子命令的输入。PKI 返回给您的证书撤销列表 (Certificate Revocation List, CRL) 是 certrldb 子命令的输入。

ikecert certdb 命令

certdb 子命令管理公钥数据库。使用此子命令的选项，可以添加、查看以及删除证书和公钥。该命令将 ikecert certlocal -ks 命令在远程系统上生成的证书作为输入接受。有关过程，请参见第 520 页中的 “如何使用自签名的公钥证书配置 IKE”。此命令还将您从 PKI 或 CA 接收的证书接受为输入。有关过程，请参见第 525 页中的 “如何使用 CA 签名的证书配置 IKE”。

证书和公钥存储在系统的 /etc/inet/ike/publickeys 目录中。-T 选项在连接的硬件上存储证书、私钥和公钥。

ikecert certrldb 命令

certrldb 子命令管理证书撤销列表 (Certificate Revocation List, CRL) 数据库 /etc/inet/ike/crls。CRL 数据库维护公钥的撤销列表。不再有效的证书包含在此列表中。当 PKI 为您提供 CRL 时，您可以使用 ikecert certrldb 命令在 CRL 数据库中安装 CRL。有关过程，请参见第 534 页中的 “如何处理证书撤销列表”。

/etc/inet/ike/publickeys 目录

/etc/inet/ike/publickeys 目录将公钥/私钥对的公钥部分及其证书包含在文件或**插槽**中。按 0755 保护该目录。ikecert certdb 命令填充该目录。-T 选项将密钥存储在 Sun Crypto Accelerator 6000 或 Sun Crypto Accelerator 4000 板上，而不是存储在publickeys 目录中。

插槽以编码形式包含在其他系统上生成的证书的 X.509 标识名。如果使用自签名的证书，则将从远程系统管理员处接收的证书用作该命令的输入。如果使用来自 CA 的证书，则将两个签名证书从 CA 安装到此数据库中。将安装一个基于发送到 CA 的证书签名请求的证书。也安装 CA 的证书。

/etc/inet/secret/ike.privatekeys 目录

/etc/inet/secret/ike.privatekeys 目录中存储属于公钥/私钥对一部分的私钥文件。按 0700 保护该目录。ikecert certlocal 命令填充 ike.privatekeys 目录。在安装其对应公钥、自签名的证书或 CA 后，私钥才生效。对应公钥存储在 /etc/inet/ike/publickeys 目录中，或者存储在 Sun Crypto Accelerator 6000 或 Sun Crypto Accelerator 4000 板上。

/etc/inet/ike/crls 目录

/etc/inet/ike/crls 目录包含证书撤销列表 (Certificate Revocation List, CRL) 文件。每个文件都对应于 /etc/inet/ike/publickeys 目录中的公共证书文件。PKI 组织为其证书提供 CRL。可以使用 ikecert certrldb 命令填充数据库。

◆ ◆ ◆ 第 25 章

Oracle Solaris 中的 IP 过滤器（概述）

本章概述 Oracle Solaris 中的 IP 过滤器功能。有关 IP 过滤器的任务，请参见第 26 章，IP 过滤器（任务）。

本章包含以下信息：

IP 过滤器的新增功能

本节介绍 IP 过滤器的新增功能。

有关 Oracle Solaris 发行版的新增功能完整列表以及说明，请参见《Oracle Solaris 10 1/13 新增功能》

用于包过滤的包过滤器钩子 (hook)

从 Solaris 10 7/07 发行版开始，Oracle Solaris 中使用包过滤器钩子来过滤网络数据包。此功能在系统管理方面具有以下优点：

- 包过滤器钩子简化了 IP 过滤器的配置。
- 现在支持跨区域过滤包。

- 使用过滤器钩子可提高 IP 过滤器的性能。

有关这些钩子进一步的详细信息，请参见第 567 页中的 “包过滤器钩子”。有关与包过滤器钩子相关的任务，请参见第 26 章，IP 过滤器（任务）。

IP 过滤器的 IPv6 包过滤

Solaris 6/06： 对于使用 IPv6 配置其全部或部分网络基础结构的系统管理员， Solaris IP 过滤器已增强为包括 IPv6 包过滤。IPv6 包过滤可以基于源/目标 IPv6 地址、包含 IPv6 地址的池和 IPv6 扩展头进行过滤。

`-6` 选项已添加到 `ipf` 命令和 `ipfstat` 命令中以用于 IPv6。虽然对于 `ipmon` 和 `ippool` 命令，命令行界面没有进行更改，但是这些命令也支持 IPv6。`ipmon` 命令已增强为包含 IPv6 包的日志记录，并且 `ippool` 命令可支持在池中包括 IPv6 地址。

有关更多信息，请参见“用于 IP 过滤器的 IPv6”。有关与 IPv6 包过滤相关的任务，请参见第 26 章，IP 过滤器（任务）。

此外，IP 过滤器的网络地址转换 (Network Address Translation, NAT) 功能支持 IPv6。有关 NAT 的更多信息，请参见第 565 页中的 “使用 IP 过滤器的 NAT 功能”。

IP 过滤器简介

Oracle Solaris 的 IP 过滤器功能替换 OS 中的 SunScreen 防火墙。与 SunScreen 防火墙一样，IP 过滤器也会提供有状态包过滤和网络地址转换 (Network Address Translation, NAT)。IP 过滤器还包括无状态包过滤以及创建和管理地址池的功能。

包过滤可提供基本的保护以防止基于网络的攻击。IP 过滤器可以按 IP 地址、端口、协议、网络接口和流量方向来进行过滤。IP 过滤器还可以按单个源 IP 地址、目标 IP 地址、IP 地址范围或地址池进行过滤。

IP 过滤器是从开源 IP 过滤器软件派生的。要查看开源 IP 过滤器的许可证条款、所有权和版权声明，缺省路径为 `/usr/lib/ipf/IPFILTER.LICENCE`。如果已经将 Oracle Solaris 安装在其他位置而没有安装在缺省位置，请修改指定的路径，以便在安装位置访问该文件。

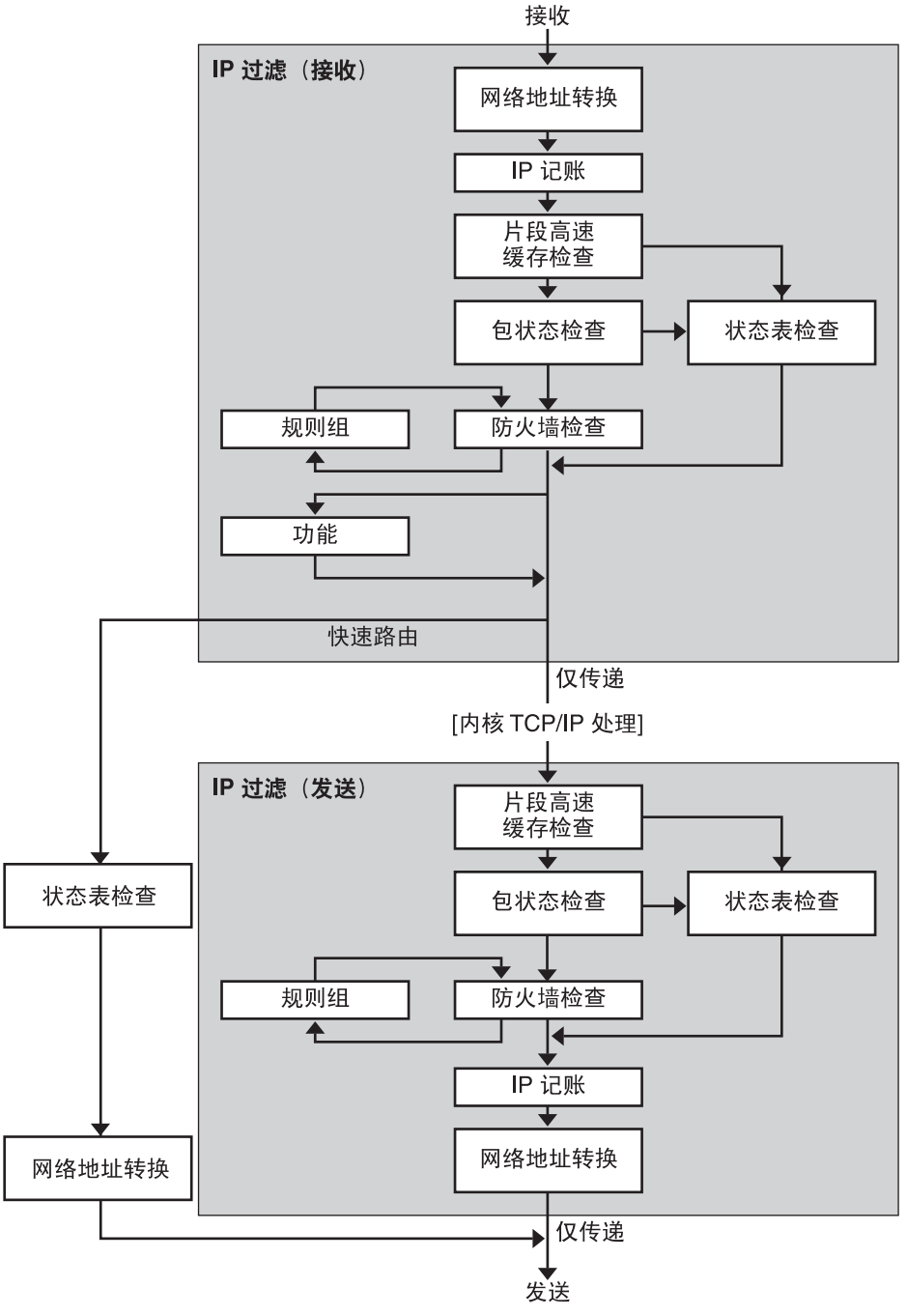
开源 IP 过滤器的信息源

Darren Reed 编写的开源 IP 过滤器软件的主页位于 `http://coombs.anu.edu.au/~avalon/ip-filter.html`。此站点包括有关开源 IP 过滤器的信息，其中包括指向标题为 "IP Filter Based Firewalls HOWTO"（Brendan Conoboy 和 Erik Fichtner，2002）的教程的链接。此教程提供了在 BSD UNIX 环境中构建防火墙的逐步说明。此教程虽然是针对 BSD UNIX 环境编写的，但也与 IP 过滤器的配置相关。

IP 过滤器包处理

在处理包时，IP 过滤器会执行一系列步骤。下图说明处理包的步骤，以及过滤如何与 TCP/IP 协议栈集成在一起。

图 25–1 包处理顺序


接收
IP 过滤（接收）
网络地址转换
IP 记账
片段高速
缓存检查
包状态检查
状态表检查
规则组
防火墙检查
功能
快速路由
仅传递
[内核 TCP/IP 处理]
IP 过滤（发送）
片段高速
缓存检查
状态表检查
包状态检查
状态表检查
规则组
防火墙检查
IP 记账
网络地址转换
网络地址转换
仅传递
发送

包处理顺序包括下列步骤：

- **网络地址转换 (Network Address Translation, NAT)**

  将专用 IP 地址转换为不同的公共地址，或者将多个专用地址的别名指定为单个公共地址。当组织具有现有的网络并需要访问 Internet 时，通过 NAT，该组织可解决 IP 地址用尽的问题。

- **IP 记帐**

  可以分别设置输入规则和输出规则，从而记录所通过的字节数。每次与规则匹配时，都会将包的字节计数添加到该规则中，并允许收集层叠统计信息。

- **片段高速缓存检查**

  如果当前流量中的下一个包是片段，而且允许前一个包通过，则也将允许包片段通过，从而绕过状态表和规则检查。

- **包状态检查**

  如果规则中包括 `keep state`，则会自动传递或阻止指定会话中的所有包，具体取决于规则指明了 `pass` 还是 `block`。

- **防火墙检查**

  可以分别设置输入规则和输出规则，确定是否允许包通过 IP 过滤器传入内核的 TCP/IP 例程或者传出到网络上。

- **组**

  通过分组可以按树的形式编写规则集合。

- **功能**

  功能是指要执行的操作。可能的功能包括 `block`、`pass`、`literal` 和 `send ICMP response`。

- **快速路由**

  快速路由指示 IP 过滤器不将包传入 UNIX IP 栈进行路由，从而导致 TTL 递减。

- **IP 验证**

  已验证的包仅通过防火墙循环一次来防止双重处理。

IP 过滤器使用准则

- IP 过滤器由 SMF 服务 svc:/network/pfil 和 svc:/network/ipfilter 管理。有关 SMF 的完整概述，请参见《Oracle Solaris 管理：基本管理》中的第 18 章 “管理服务（概述）”。有关与 SMF 相关的逐步过程的信息，请参见《Oracle Solaris 管理：基本管理》中的第 19 章 “管理服务（任务）”。
- IP 过滤器要求直接编辑配置文件。

- IP 过滤器作为 Oracle Solaris 的一部分安装。缺省情况下，在初次安装后不会激活 IP 过滤器。要配置过滤，必须编辑配置文件并手动激活 IP 过滤器。可以通过重新引导系统，或者使用 `ifconfig` 命令检测接口来激活过滤。有关更多信息，请参见 `ifconfig(1M)` 手册页。有关与启用 IP 过滤器关联的任务，请参见第 571 页中的 “配置 IP 过滤器”。
- 要管理 IP 过滤器，必须能够承担拥有 IP 过滤器管理权限配置文件的角色或者成为超级用户。可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。
- IP 网络多路径 (IP Network Multipathing, IPMP) 仅支持无状态过滤。

  要使 IP 过滤器对进出 IPMP 组的流量执行无状态过滤，您必须设置 `ipmp_hook_emulation` 参数。缺省情况下，该参数设置为零 (`0`)，这意味着 IP 过滤器无法对属于 IPMP 组的物理接口上的流量执行有状态包检查。要启用 IPMP 包过滤，请发出以下命令：

  ```
  ndd -set /dev/ip ipmp_hook_emulation 1
  ```
- Oracle Solaris Cluster 软件对可伸缩服务不支持使用 IP 过滤器进行过滤，但对故障转移服务支持 IP 过滤器。有关在群集中配置 IP 过滤器的指导和限制，请参见《Oracle Solaris Cluster 软件安装指南》中的“Oracle Solaris OS 功能限制”。
- 如果在充当系统中其他区域的虚拟路由器的区域中实现 IP 过滤器过则，则支持在各区域之间进行过滤。

使用 IP 过滤器配置文件

IP 过滤器可用于提供防火墙服务或网络地址转换 (Network Address Translation, NAT)。可以使用可装入的配置文件实现 IP 过滤器。IP 过滤器包括一个名为 `/etc/ipf` 的目录。可以创建名为 `ipf.conf`、`ipnat.conf` 和 `ippool.conf` 的配置文件，并将其存储在 `/etc/ipf` 目录中。当这些文件驻留在 `/etc/ipf` 目录中时，在引导过程中会将其自动装入。也可以将配置文件存储在其他位置并将其手动装入。有关配置文件示例，请参见第 600 页中的 “创建和编辑 IP 过滤器配置文件”。

使用 IP 过滤器规则集合

要管理防火墙，请使用 IP 过滤器指定用于过滤网络通信流量的规则集合。可以创建以下类型的规则集：

- 包过滤规则集合
- 网络地址转换 (Network Address Translation, NAT) 规则集合

此外，还可以创建地址池以引用 IP 地址组。然后，可以在规则集合中使用这些池。地址池有助于加快规则处理速度，还可使大型地址组更易于管理。

使用 IP 过滤器的包过滤功能

可以使用包过滤规则集合来设置包过滤。使用 ipf 命令可以对包过滤规则集合进行处理。有关 ipf 命令的更多信息，请参见 ipf(1M) 命令。

可以在命令行上使用 ipf 命令或在包过滤配置文件中创建包过滤规则。如果希望在引导时装入包过滤规则，请创建一个名为 /etc/ipf/ipf.conf 的配置文件，在其中放置包过滤规则。如果不希望在引导时装入包过滤规则，请将 ipf.conf 文件放置在所选的位置中，然后使用 ipf 命令手动激活包过滤。

使用 IP 过滤器可以维护两种包过滤规则集合：活动规则集合和非活动规则集合。大多数情况下，会使用活动规则集合。但是，使用 ipf -I 命令可以将命令操作应用于非活动规则列表。除非您选择非活动规则列表，否则 IP 过滤器不会使用该列表。非活动规则列表可提供一个存储规则的位置，而不会影响活动包过滤。

在传递或阻止包之前，IP 过滤器会按照从已配置规则列表开头到其结尾的顺序处理规则列表中的规则。IP 过滤器可维护用于确定它是否将传递包的标志。它会遍历整个规则集合，并基于最后一个匹配规则来确定是传递包还是阻止包。

此过程有两种例外情况。第一种例外情况是当包与包含 quick 关键字的规则匹配时。如果规则包括 quick 关键字，则会针对该规则执行操作，并且不会检查后续规则。第二种例外情况是当包与包含 group 关键字的规则匹配时。如果包与组匹配，则仅会检查标记有该组的规则。

配置包过滤规则

使用以下语法可创建包过滤规则：

action [in|out] *option keyword, keyword...*

1. 每个规则都以操作开头。如果包与规则匹配，则 IP 过滤器将操作应用于该包。以下列表包括应用于包的常用操作。

block	阻止包通过过滤器。
pass	允许包通过过滤器。
log	记录包但不确定是阻止包还是传递包。使用 ipmon 命令可查看日志。
count	将包包括在过滤器统计信息中。使用 ipfstat 命令可查看统计信息。
skip *number*	使过滤器跳过 *number* 个过滤规则。
auth	请求由验证包信息的用户程序执行包验证。该程序会确定是传递包还是阻止包。

2. 操作后面的下一个单词必须是 in 或 out。您的选择将确定是将包过滤规则应用于传入包还是应用于传出包。

3. 接下来，可以从选项列表中进行选择。如果使用多个选项，则这些选项必须采用此处显示的顺序。

`log`	如果规则是最后一个匹配规则，则记录包。使用 `ipmon` 命令可查看日志。
`quick`	如果存在匹配的包，则执行包含 `quick` 选项的规则。所有进一步的规则检查都将停止。
`on` *interface-name*	仅当包移入或移出指定接口时才应用规则。
`dup-to` *interface-name*	复制包并将 *interface-name* 上的副本向外发送到随意指定的 IP 地址。
`to` *interface-name*	将包移动到 *interface-name* 上的外发队列。

4. 指定选项后，可以从确定包是否与规则匹配的各关键字中进行选择。必须按此处显示的顺序使用以下关键字。

注 – 缺省情况下，所有与配置文件中的任何规则都不匹配的包会通过此过滤器。

`tos`	基于表示为十六进制或十进制整数的服务类型值，对包进行过滤。
`ttl`	基于包的生存时间值与包匹配。在包中存储的生存时间值指明了包在被废弃之前可在网络中存在的时间长度。
`proto`	与特定协议匹配。可以使用在 `/etc/protocols` 文件中指定的任何协议名称，或者使用十进制数来表示协议。关键字 `tcp/udp` 可以用于与 TCP 包或 UDP 包匹配。
`from/to/all/ any`	与以下任一项或所有项匹配：源 IP 地址、目标 IP 地址和端口号。`all` 关键字用于接受来自所有源和发往所有目标的包。
`with`	与和包关联的指定属性匹配。在关键字前面插入 `not` 或 `no` 一词，以便仅当选项不存在时才与包匹配。
`flags`	供 TCP 用来基于已设置的 TCP 标志进行过滤。有关 TCP 标志的更多信息，请参见 `ipf(4)` 手册页。
`icmp-type`	根据 ICMP 类型进行过滤。仅当 `proto` 选项设置为 `icmp` 时才使用此关键字；如果使用 `flags` 选项，则不使用此关键字。
`keep` *keep-options*	确定为包保留的信息。可用的 *keep-options* 包括 `state` 选项和 `frags` 选项。`state` 选项会保留有关会话的信息，并可以保留在 TCP、UDP 和 ICMP 包中。`frags` 选项可保留有关包片段的信息，并将该信息应用于后续片段。*keep-options* 允许匹配包通过，而不会查询访问控制列表。

head *number*	为过滤规则创建一个新组，该组由数字 *number* 表示。
group *number*	将规则添加到编号为 *number* 的组而不是缺省组。如果未指定其他组，则将所有过滤规则放置在组 0 中。

以下示例说明如何组织包过滤规则语法以创建规则。要阻止从 IP 地址 192.168.0.0/16 传入的通信流量，需要在规则列表中包括以下规则：

```
block in quick from 192.168.0.0/16 to any
```

有关用于编写包过滤规则的完整语法和句法，请参见 ipf(4) 手册页。有关与包过滤关联的任务，请参见第 584 页中的 “管理 IP 过滤器的包过滤规则集合”。有关示例中所示的 IP 地址方案 (192.168.0.0/16) 的说明，请参见第 2 章，规划 TCP/IP 网络（任务）。

使用 IP 过滤器的 NAT 功能

NAT 可设置映射规则，用于将源 IP 地址和目标 IP 地址转换为其他 Internet 或内联网地址。这些规则可修改传入或传出 IP 包的源地址和目标地址并继续发送包。另外，还可以使用 NAT 将流量从一个端口重定向到另一个端口。在对包进行任何修改或重定向的过程中，NAT 将维护包的完整性。

使用 ipnat 命令可对 NAT 规则列表进行处理。有关 ipnat 命令的更多信息，请参见 ipnat(1M) 命令。

可以在命令行上使用 ipnat 命令或在 NAT 配置文件中创建 NAT 规则。NAT 配置规则驻留在 ipnat.conf 文件中。如果希望在引导时装入 NAT 规则，请创建一个名为 /etc/ipf/ipnat.conf 的文件，在其中放置 NAT 规则。如果不希望在引导时装入 NAT 规则，请将 ipnat.conf 文件放置在所选的位置中，然后使用 ipnat 命令手动激活包过滤。

NAT 规则可以应用到 IPv4 和 IPv6 地址。但是，不能在单个规则中指定两种类型的地址。相反，必须为每种地址类型设置单独的规则。在包含 IPv6 地址的 NAT 规则中，不能同时使用 mapproxy 和 rdrproxy NAT 命令。

配置 NAT 规则

使用以下语法创建 NAT 规则：

command interface-name parameters

1. 每个规则都以以下命令之一开头：

map	在无法控制的循环过程中将一个 IP 地址或网络映射到另一个 IP 地址或网络。
rdr	将包从一个 IP 地址和端口对重定向到另一个 IP 地址和端口对。

`bimap` 在外部 IP 地址和内部 IP 地址之间建立双向 NAT。

`map-block` 建立基于静态 IP 地址的转换。此命令基于将地址强制转换为目标范围的算法。

2. 此命令后面的下一个单词是接口名称，如 `hme0`。
3. 接下来，可以从确定 NAT 配置的各种参数中进行选择。其中一些参数包括：

`ipmask` 指定网络掩码。

`dstipmask` 指定 `ipmask` 要转换成的地址。

`mapport` 指定 `tcp`、`udp` 或 `tcp/udp` 协议以及端口号的范围。

以下示例说明如何组织 NAT 规则语法以创建 NAT 规则。要重新编写从源地址为 `192.168.1.0/24` 的 `de0` 设备上传出的包并在外部将该设备的源地址显示为 `10.1.0.0/16`，需要在 NAT 规则集合中包括以下规则：

```
map de0 192.168.1.0/24 -> 10.1.0.0/16
```

以下规则适用于 IPv6 地址：

```
map ppp0 fec0:1::/64 -> 2000:1:2::/72 portmap tcp/udp 1025:65000
map-block ppp0 fe80:0:0:209::/64 -> 209:1:2::/72 ports auto
rdr ce0 209::ffff:fe13:e43e port 80 -> fec0:1::e,fec0:1::f port 80 tcp round-robin
```

有关用于编写 NAT 规则的完整语法和句法，请参见 `ipnat(4)` 手册页。

使用 IP 过滤器的地址池功能

地址池可建立用于命名一组地址/网络掩码对的单个引用。地址池提供可减少将 IP 地址与规则相匹配的时间的进程，还可使大型地址组更易于管理。

地址池配置规则驻留在 `ippool.conf` 文件中。如果希望在引导时装入地址池规则，请创建一个名为 `/etc/ipf/ippool.conf` 的文件，在其中放置地址池规则。如果不希望在引导时装入地址池规则，请将 `ippool.conf` 文件放置在所选的位置中，然后使用 `ippool` 命令手动激活包过滤。

配置地址池

使用以下语法可创建地址池：

`table role =` *role-name* `type =` *storage-format* `number =` *reference-number*

`table` 定义对多个地址的引用。

`role` 指定 IP 过滤器中池的角色。此时，可以引用的唯一角色是 `ipf`。

`type` 指定池的存储格式。

number　　指定过滤规则所用的引用号。

例如，要将地址组 10.1.1.1 和 10.1.1.2 以及网络 192.16.1.0 作为池编号 13 引用，需要在地址池配置文件中包括以下规则：

```
table role = ipf type = tree number = 13
{ 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24 };
```

然后，要在过滤规则中引用池编号 13，需要构建与以下示例类似的规则：

```
pass in from pool/13 to any
```

请注意，必须在装入包含对池的引用的规则文件之前装入池文件。如果不这样做，则池是未定义的，如以下输出所示：

```
# ipfstat -io
empty list for ipfilter(out)
block in from pool/13(!) to any
```

即使稍后添加池，所添加的池也不会更新内核规则集合。另外，还需要重新装入引用池的规则文件。

有关用于编写包过滤规则的完整语法和句法，请参见 ippool(4) 手册页。

包过滤器钩子

从 Solaris 10 7/07 发行版开始，包过滤器钩子替代了 pfil 模块，用于启用 IP 过滤器。在早期的 Solaris 发行版中，要求配置 pfil 模块作为设置 IP 过滤器的附加步骤。这一额外的配置要求增加了出现错误进而可能导致 IP 过滤器无法正常工作的风险。此外，在 IP 和设备驱动程序之间插入 pfil STREAMS 模块会导致性能下降。最后，pfil 模块无法在区域之间执行包拦截。

包过滤器钩子的使用简化了启用 IP 过滤器的过程。通过这些钩子，IP 过滤器使用 pre-routing（输入）和 post-routing（输出）过滤器阀控制进出 Oracle Solaris 系统的包流。

有了包过滤器钩子，就不必使用 pfil 模块。因此，与该模块相关的下列组件也已被删除。

- pfil 驱动程序
- pfil 守护进程
- svc:/network/pfil SMF 服务

有关与启用 IP 过滤器相关的任务，请参见第 26 章，IP 过滤器（任务）。

IP 过滤器和 pfil STREAMS 模块

注 –

pfil 模块只能在以下 Solaris 发行版中与 IP 过滤器一起使用：

- Solaris 10 3/05 发行版
- Solaris 10 1/06 发行版
- Solaris 10 6/06 发行版
- Solaris 10 11/06 发行版

从 Solaris 10 7/07 发行版开始，pfil 模块已被包过滤器钩子替代，不再与 IP 过滤器一起使用。

pfil STREAMS 模块是启用 IP 过滤器所必需的。但是，IP 过滤器不提供将模块推送到每个接口上的自动机制。相反，pfil STREAMS 模块由 SMF 服务 svc:/network/pfil 进行管理。要在网络接口上激活过滤，需要首先配置 pfil.ap 文件。然后激活 svc:/network/pfil 服务，以便为网络接口提供 pfil STREAMS 模块。要使 STREAMS 模块生效，必须重新引导系统，或者必须取消检测要过滤的每个网络接口，然后重新检测。要激活 IPv6 包过滤功能，需要检测接口的 inet6 版本。

如果未找到网络接口的 pfil 模块，请将 SMF 服务置于维护状态。导致此情况的最常见原因是错误编辑了 /etc/ipf/pfil.ap 文件。如果将服务置于维护模式，则在过滤日志文件中会记录事件。

有关与激活 IP 过滤器关联的任务，请参见第 571 页中的 “配置 IP 过滤器”。

用于 IP 过滤器的 IPv6

从 Solaris 6/06 发行版开始，IP 过滤器可以支持 IPv6。IPv6 包过滤可以基于源/目标 IPv6 地址、包含 IPv6 地址的池和 IPv6 扩展头进行过滤。

IPv6 在许多方面都与 IPv4 类似。但是，这两个版本的 IP 的包头和包大小是不同的，这是 IP 过滤器的重要注意事项。称为 *jumbogram* 的 IPv6 包包含长度超过 65,535 字节的数据报。IP 过滤器不支持 IPv6 jumbogram。要了解有关其他 IPv6 功能的更多信息，请参见第 63 页中的 “IPv6 的主要特征”。

注 – 有关 jumbogram 的更多信息，请参阅 Internet 工程任务组 (Internet Engineering Task Force, IETF) 的文档 "IPv6 Jumbograms" (RFC 2675)。[http://www.ietf.org/rfc/rfc2675.txt]

与 IPv6 关联的 IP 过滤器任务和与 IPv4 关联的任务差异不大。最明显的差异是，前者将 -6 选项与某些命令一起使用。ipf 命令和 ipfstat 命令都包括用于 IPv6 包过滤的 -6 选项。使用带有 -6 选项的 ipf 命令可以装入和刷新 IPv6 包过滤规则。要显示 IPv6 统计信息，请使用带有 -6 选项的 ipfstat 命令。尽管没有用于 IPv6 支持的关联选项，ipmon 和 ippool 命令仍支持 IPv6。ipmon 命令已增强为包含 IPv6 包的日志记录。ippool 命令支持具有 IPv6 地址的包。可以创建仅包含 IPv4 地址或 IPv6 地址的池，或创建在同一池内既包含 IPv4 地址又包含 IPv6 地址的池。

可以使用 ipf6.conf 文件创建用于 IPv6 的包过滤规则集合。缺省情况下，ipf6.conf 配置文件包括在 /etc/ipf 目录中。与其他过滤配置文件一样，ipf6.conf 文件存储在 /etc/ipf 目录中时也会在引导过程中自动装入。您也可以在其他位置创建和存储 IPv6 配置文件，然后将其手动装入。

设置用于 IPv6 的包过滤规则后，请通过检测接口的 inet6 版本激活 IPv6 包过滤功能。

有关 IPv6 的更多信息，请参见第 3 章，IPv6 介绍（概述）。有关与 IP 过滤器相关的任务，请参见第 26 章，IP 过滤器（任务）。

IP 过滤器手册页

下表包括与 IP 过滤器相关的手册页文档。

手册页	说明
ipf(1M)	使用 ipf 命令可以完成以下任务： ■ 处理包过滤规则集合。 ■ 禁用和启用过滤。 ■ 重置统计信息，并使内核中的接口列表与当前接口状态列表重新同步。
ipf(4)	包含用于创建 IP 过滤器包过滤规则的语法和句法。
ipfilter(5)	提供开放源代码 IP 过滤器许可信息。
ipfs(1M)	使用 ipfs 命令保存 NAT 信息和状态表信息并在重新引导后将其恢复。
ipfstat(1M)	使用 ipfstat 命令检索和显示有关包处理的统计信息。
ipmon(1M)	使用 ipmon 命令打开日志设备并查看包过滤和 NAT 的记录包。
ipnat(1M)	使用 ipnat 命令可以完成以下任务： ■ 处理 NAT 规则。 ■ 检索和显示 NAT 统计信息。

手册页	说明
ipnat(4)	包含用于创建 NAT 规则的语法和句法。
ippool(1M)	使用 ippool 命令创建和管理地址池。
ippool(4)	包含用于创建 IP 过滤器地址池的语法和句法。
ndd(1M)	显示 pfil STREAMS 模块的当前过滤参数和可调参数的当前值。

◆ ◆ ◆ 第 26 章

IP 过滤器（任务）

本章提供有关任务的逐步说明。有关 IP 过滤器的概述信息，请参见第 25 章，Oracle Solaris 中的 IP 过滤器（概述）。

本章包含以下信息：

配置 IP 过滤器

以下任务列表提供了与配置 IP 过滤器相关的过程。

表 26–1 配置 IP 过滤器（任务列表）

任务	说明	参考
最初启用 IP 过滤器。	缺省情况下不启用 IP 过滤器。必须手动启用它，或者使用 /etc/ipf/ 目录中的配置文件并重新引导系统。从 Solaris 10 7/07 发行版开始，包过滤器钩子替代了 pfil 模块，用于启用 IP 过滤器。	第 572 页中的 “如何启用 IP 过滤器”
重新启用 IP 过滤器。	如果 IP 过滤器被取消激活或禁用，则可以通过重新引导系统或通过使用 ipf 命令重新启用 IP 过滤器。	第 573 页中的 “如何重新启用 IP 过滤器”

表 26–1 配置 IP 过滤器（任务列表） （续）

任务	说明	参考
启用回送过滤	作为一个选项，您可以启用回送过滤，例如，过滤区域之间的流量。	第 574 页中的 “如何启用回送过滤”

▼ 如何启用 IP 过滤器

使用此过程可在运行 Solaris 10 7/07 OS 或更高版本的系统上启用 IP 过滤器。如果系统运行的 Solaris 10 发行版早于 Solaris 10 7/07 OS 并且要启用 IP 过滤器，请参见第 576 页中的 “使用 `pfil` 模块”。

1 承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 创建包过滤规则集。

包过滤规则集合包含由 IP 过滤器使用的包过滤规则。如果希望在引导时装入包过滤规则，请编辑 `/etc/ipf/ipf.conf` 文件以实现 IPv4 包过滤。对于 IPv6 包过滤规则，请使用 `/etc/ipf/ipf6.conf` 文件。如果不希望在引导时装入包过滤规则，请将这些规则放置在所选的文件中，然后手动激活包过滤。有关包过滤的信息，请参见第 563 页中的 “使用 IP 过滤器的包过滤功能”。有关使用配置文件的信息，请参见第 600 页中的 “创建和编辑 IP 过滤器配置文件”。

3 （可选）创建网络地址转换 (Network Address Translation, NAT) 配置文件。

注 – 网络地址转换 (Network Address Translation, NAT) 不支持 IPv6。

如果要使用网络地址转换，请创建 `ipnat.conf` 文件。如果希望在引导时装入 NAT 规则，请创建一个名为 `/etc/ipf/ipnat.conf` 的文件，在其中放置 NAT 规则。如果不希望在引导时装入 NAT 规则，请将 `ipnat.conf` 文件放置在所选的位置中，然后手动激活 NAT 规则。

有关 NAT 的更多信息，请参见第 565 页中的 “使用 IP 过滤器的 NAT 功能”。

4 （可选）创建地址池配置文件。

如果要将一组地址作为单个地址池引用，请创建 `ipool.conf` 文件。如果希望在引导时装入地址池配置文件，请创建一个名为 `/etc/ipf/ippool.conf` 的文件，在其中放置地址池。如果不希望在引导时装入地址池配置文件，请将 `ippool.conf` 文件放置在所选的位置中，然后手动激活这些规则。

一个地址池可以只包含 IPv4 地址和 IPv6 地址中的一种，也可以同时包含这两种地址。

有关地址池的更多信息，请参见第 566 页中的 “使用 IP 过滤器的地址池功能”。

5 （可选）启用回送流量的过滤。

如果打算过滤系统中配置的区域之间的流量，则必须启用回送过滤。请参见第 574 页中的 “如何启用回送过滤”。还要确保定义了应用于这些区域的相应规则集合。

6 激活 IP 过滤器。

```
# svcadm enable network/ipfilter
```

▼ 如何重新启用 IP 过滤器

在暂时禁用包过滤后，可以重新启用它。

1 承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 使用以下方法之一启用 IP 过滤器并激活过滤：

- 重新引导计算机。

  ```
  # reboot
  ```

  注 – 若启用了 IP 过滤器，则重新引导后会装入以下文件（如果它们存在）：`/etc/ipf/ipf.conf` 文件、`/etc/ipf/ipf6.conf` 文件（使用 IPv6 时）或 `/etc/ipf/ipnat.conf`。

- 执行以下系列命令以启用 IP 过滤器并激活过滤：

  a. 启用 IP 过滤器。

  ```
  # ipf -E
  ```

  b. 激活包过滤。

  ```
  # ipf -f filename
  ```

  c. （可选）激活 NAT。

  ```
  # ipnat -f filename
  ```

  注 – 网络地址转换 (Network Address Translation, NAT) 不支持 IPv6。

▼ 如何启用回送过滤

注 – 仅当系统运行 Solaris 10 7/07 发行版或更高版本时，才能过滤回送流量。在以前的 Oracle Solaris 10 发行版中，不支持回送过滤。

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

2. **如果 IP 过滤器正在运行，则将其停止。**

   ```
   # svcadm disable network/ipfilter
   ```

3. **通过在文件开头添加下面的行来编辑 /etc/ipf.conf 或 /etc/ipf6.conf 文件：**

   ```
   set intercept_loopback true;
   ```

   此行必须位于文件中定义的所有 IP 过滤器规则之前。不过，可以在此行之前插入注释，与以下示例类似：

   ```
   #
   # Enable loopback filtering to filter between zones
   #
   set intercept_loopback true;
   #
   # Define policy
   #
   block in all
   block out all
   <other rules>
   ...
   ```

4. **启动 IP 过滤器。**

   ```
   # svcadm enable network/ipfilter
   ```

5. **要验证回送过滤的状态，请使用以下命令：**

   ```
   # ipf -T ipf_loopback
   ipf_loopback    min 0   max 0x1 current 1
   #
   ```

   如果已禁用回送过滤，该命令将生成以下输出：

   ```
   ipf_loopback    min 0   max 0x1 current 0
   ```

取消激活和禁用 IP 过滤器

在以下情况下，可能希望取消激活或禁用包过滤和 NAT：

- 要进行测试
- 在认为系统问题是由 IP 过滤器所导致时，对这些问题进行故障排除

以下任务列表提供了与取消激活或禁用 IP 过滤器功能相关的过程。

表 26–2 取消激活和禁用 IP 过滤器（任务列表）

任务	说明	参考
取消激活包过滤。	使用 `ipf` 命令取消激活包过滤。	第 575 页中的 “如何取消激活包过滤”
取消激活 NAT。	使用 `ipnat` 命令取消激活 NAT。	第 576 页中的 “如何取消激活 NAT”
禁用包过滤和 NAT。	使用 `ipf` 命令禁用包过滤和 NAT。	第 576 页中的 “如何禁用包过滤”

▼ 如何取消激活包过滤

以下过程通过从活动的过滤规则集合中清除包过滤规则，取消激活 IP 过滤器包过滤。该过程不禁用 IP 过滤器。通过将规则添加到规则集，可以重新激活 IP 过滤器。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **使用以下方法之一取消激活 IP 过滤器规则：**

- 从内核中删除活动规则集。

  ```
  # ipf -Fa
  ```

  此命令取消激活所有的包过滤规则。
- 删除传入包的过滤规则。

  ```
  # ipf -Fi
  ```

  此命令取消激活传入包的包过滤规则。
- 删除传出包的过滤规则。

  ```
  # ipf -Fo
  ```

  此命令取消激活传出包的包过滤规则。

▼ 如何取消激活 NAT

以下过程通过从活动的 NAT 规则集合中清除 NAT 规则，取消激活 IP 过滤器 NAT 规则。该过程不禁用 IP 过滤器。通过将规则添加到规则集，可以重新激活 IP 过滤器。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

2 **从内核中删除 NAT。**

```
# ipnat -FC
```

`-C` 选项删除当前 NAT 规则列表中的所有项。`-F` 选项删除当前 NAT 转换表（它显示当前活动的 NAT 映射）中的所有活动项。

▼ 如何禁用包过滤

在执行此过程时，包过滤和 NAT 都会从内核中删除。如果使用此过程，则必须重新启用 IP 过滤器以便重新激活包过滤和 NAT。有关更多信息，请参见第 573 页中的 "如何重新启用 IP 过滤器"。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

2 **禁用包过滤，并允许所有包传入网络。**

```
# ipf –D
```

注 – `ipf -D` 命令从规则集合中清除规则。重新启用过滤时，必须将规则添加到规则集合。

使用 pfil 模块

本节介绍如何使用 `pfil` STREAMS 模块激活或取消激活 IP 过滤器以及如何查看 `pfil` 统计信息。该过程只适用于运行下列 Solaris 发行版之一的系统：

- Solaris 10 3/05 发行版
- Solaris 10 1/06 发行版
- Solaris 10 6/06 发行版
- Solaris 10 11/06 发行版

以下任务列表提供了与配置 pfil 模块关联的过程。

表 26–3 使用 pfil 模块（任务列表）

任务	说明	参考
启用 IP 过滤器	缺省情况下不启用 IP 过滤器。必须手动启用它，或者使用 /etc/ipf/ 目录中的配置文件并重新引导系统。	第 577 页中的 “如何在以前的 Solaris 发行版中启用 IP 过滤器”
为包过滤激活 NIC	配置 pfil 模块，以在 NIC 上激活包过滤	第 579 页中的 “如何为包过滤激活 NIC”
在 NIC 上取消激活 IP 过滤器	删除 NIC 并允许所有包通过 NIC。	第 580 页中的 “如何在 NIC 上取消激活 IP 过滤器”
查看 pfil 统计信息。	使用 ndd 命令查看 pfil 模块的统计信息，以帮助对 IP 过滤器进行故障排除。	第 582 页中的 “如何查看 IP 过滤器的 pfil 统计信息”

▼ 如何在以前的 Solaris 发行版中启用 IP 过滤器

IP 过滤器与 Oracle Solaris 一同安装。但是，缺省情况下不启用包过滤。使用以下过程可以激活 IP 过滤器。

注 – 如果您的系统运行的是 Solaris 10 7/07 发行版或更高版本，请执行第 572 页中的 “如何启用 IP 过滤器”过程（使用包过滤器钩子）。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **启动所选的文件编辑器，然后编辑 /etc/ipf/pfil.ap 文件。**

此文件包含主机上网络接口卡 (Network Interface Card, NIC) 的名称。缺省情况下，这些名称已被注释掉。对传输要过滤的网络通信流量的设备名称取消注释。如果未列出您系统的 NIC 名称，则添加一行以指定 NIC。

```
# vi /etc/ipf/pfil.ap
# IP Filter pfil autopush setup
#
# See autopush(1M) manpage for more information.
#
# Format of the entries in this file is:
#
#major  minor lastminor modules
```

```
#le     -1      0       pfil
#qe     -1      0       pfil
hme     -1      0       pfil (Device has been uncommented for filtering)
#qfe    -1      0       pfil
#eri    -1      0       pfil
#ce     -1      0       pfil
#bge    -1      0       pfil
#be     -1      0       pfil
#vge    -1      0       pfil
#ge     -1      0       pfil
#nf     -1      0       pfil
#fa     -1      0       pfil
#ci     -1      0       pfil
#el     -1      0       pfil
#ipdptp -1      0       pfil
#lane   -1      0       pfil
#dmfe   -1      0       pfil
```

3 **通过重新启动 network/pfil 服务实例，激活对 /etc/ipf/pfil.ap 文件所做的更改。**

```
# svcadm restart network/pfil
```

4 **创建包过滤规则集。**

包过滤规则集合包含由 IP 过滤器使用的包过滤规则。如果希望在引导时装入包过滤规则，请编辑 /etc/ipf/ipf.conf 文件以实现 IPv4 包过滤。对于 IPv6 包过滤规则，请使用 /etc/ipf/ipf6.conf 文件。如果不希望在引导时装入包过滤规则，请将这些规则放置在所选的文件中，然后手动激活包过滤。有关包过滤的信息，请参见第 563 页中的 “使用 IP 过滤器的包过滤功能”。有关使用配置文件的信息，请参见第 600 页中的 “创建和编辑 IP 过滤器配置文件”。

5 **（可选）创建网络地址转换 (Network Address Translation, NAT) 配置文件。**

注 – 网络地址转换 (Network Address Translation, NAT) 不支持 IPv6。

如果要使用网络地址转换，请创建 ipnat.conf 文件。如果希望在引导时装入 NAT 规则，请创建一个名为 /etc/ipf/ipnat.conf 的文件，在其中放置 NAT 规则。如果不希望在引导时装入 NAT 规则，请将 ipnat.conf 文件放置在所选的位置中，然后手动激活 NAT 规则。

有关 NAT 的更多信息，请参见第 565 页中的 “使用 IP 过滤器的 NAT 功能”。

6 **（可选）创建地址池配置文件。**

如果要将一组地址作为单个地址池引用，请创建 ipool.conf 文件。如果希望在引导时装入地址池配置文件，请创建一个名为 /etc/ipf/ippool.conf 的文件，在其中放置地址池。如果不希望在引导时装入地址池配置文件，请将 ippool.conf 文件放置在所选的位置中，然后手动激活这些规则。

一个地址池可以只包含 IPv4 地址和 IPv6 地址中的一种，也可以同时包含这两种地址。

有关地址池的更多信息，请参见第 566 页中的 “使用 IP 过滤器的地址池功能”。

7 **使用以下方法之一激活 IP 过滤器：**

- 启用 IP 过滤器并重新引导计算机。

```
# svcadm enable network/ipfilter
# reboot
```

注 – 如果无法在 NIC 上安全地使用 `ifconfig unplumb` 和 `ifconfig plumb` 命令，则需要重新引导。

- 通过使用 `ifconfig unplumb` 和 `ifconfig plumb` 命令启用 NIC。然后启用 IP 过滤器。必须检测接口的 `inet6` 版本，以便实现 IPv6 包过滤。

```
# ifconfig hme0 unplumb
# ifconfig hme0 plumb 192.168.1.20 netmask 255.255.255.0 up
# ifconfig hme0 inte6 unplumb
# ifconfig hme0 inet6 plumb fec3:f849::1/96 up
# svcadm enable network/ipfilter
```

有关 `ifconfig` 命令的更多信息，请参见 `ifconfig(1M)` 手册页。

▼ 如何为包过滤激活 NIC

如果 `/etc/ipf/ipf.conf` 文件（或使用 IPv6 时的 `/etc/ipf/ipf6.conf` 文件）存在，则在引导时启用 IP 过滤器。如果需要在启用 IP 过滤器后在 NIC 上启用过滤，请使用以下过程。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **启动所选的文件编辑器，然后编辑 `/etc/ipf/pfil.ap` 文件。**

此文件包含主机上 NIC 的名称。缺省情况下，这些名称已被注释掉。对传输要过滤的网络通信流量的设备名称取消注释。如果未列出您系统的 NIC 名称，则添加一行以指定 NIC。

```
# vi /etc/ipf/pfil.ap
# IP Filter pfil autopush setup
#
# See autopush(1M) manpage for more information.
#
# Format of the entries in this file is:
#
#major  minor lastminor modules

#le     -1      0       pfil
#qe     -1      0       pfil
```

```
hme     -1      0       pfil (Device has been uncommented for filtering)
#qfe    -1      0       pfil
#eri    -1      0       pfil
#ce     -1      0       pfil
#bge    -1      0       pfil
#be     -1      0       pfil
#vge    -1      0       pfil
#ge     -1      0       pfil
#nf     -1      0       pfil
#fa     -1      0       pfil
#ci     -1      0       pfil
#el     -1      0       pfil
#ipdptp -1      0       pfil
#lane   -1      0       pfil
#dmfe   -1      0       pfil
```

3 **通过重新启动 network/pfil 服务实例，激活对 /etc/ipf/pfil.ap 文件所做的更改。**

```
# svcadm restart network/pfil
```

4 **使用以下方法之一启用 NIC：**

- 重新引导计算机。

  ```
  # reboot
  ```

  注 – 如果无法在 NIC 上安全地使用 `ifconfig unplumb` 和 `ifconfig plumb` 命令，则需要重新引导。

- 将 `ifconfig` 命令与 `unplumb` 和 `plumb` 选项一起使用，以启用要过滤的 NIC。必须检测每个接口的 `inet6` 版本，以便实现 IPv6 包过滤。

  ```
  # ifconfig hme0 unplumb
  # ifconfig hme0 plumb 192.168.1.20  netmask 255.255.255.0  up
  # ifconfig hme0 inet6 unplumb
  # ifconfig hme0 inet6 plumb fec3:f840::1/96 up
  ```

  有关 `ifconfig` 命令的更多信息，请参见 `ifconfig(1M)` 手册页。

▼ 如何在 NIC 上取消激活 IP 过滤器

如果需要在 NIC 上停止过滤包，请使用以下过程。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 启动所选的文件编辑器，然后编辑 /etc/ipf/pfil.ap 文件。

此文件包含主机上 NIC 的名称。已经用于过滤网络通信流量的 NIC 被取消注释。注释掉不再希望用来过滤网络通信流量的设备名称。

```
# vi /etc/ipf/pfil.ap
# IP Filter pfil autopush setup
#
# See autopush(1M) manpage for more information.
#
# Format of the entries in this file is:
#
#major  minor lastminor modules

#le     -1      0       pfil
#qe     -1      0       pfil
#hme    -1      0       pfil (Commented-out device no longer filters network traffic)
#qfe    -1      0       pfil
#eri    -1      0       pfil
#ce     -1      0       pfil
#bge    -1      0       pfil
#be     -1      0       pfil
#vge    -1      0       pfil
#ge     -1      0       pfil
#nf     -1      0       pfil
#fa     -1      0       pfil
#ci     -1      0       pfil
#el     -1      0       pfil
#ipdptp -1      0       pfil
#lane   -1      0       pfil
#dmfe   -1      0       pfil
```

3 使用以下方法之一取消激活 NIC：

- 重新引导计算机。

  ```
  # reboot
  ```

  注 – 如果无法在 NIC 上安全地使用 `ifconfig unplumb` 和 `ifconfig plumb` 命令，则需要重新引导。

- 将 `ifconfig` 命令与 `unplumb` 和 `plumb` 选项一起使用，以取消激活 NIC。必须取消检测每个接口的 `inet6` 版本，以便取消激活 IPv6 包过滤。请执行以下步骤。系统中的样例设备为 `hme`：

  a. 标识取消激活的设备的主要编号。

  ```
  # grep hme /etc/name_to_major
  hme 7
  ```

  b. 显示 `hme0` 的当前 `autopush` 配置。

  ```
  # autopush -g -M 7 -m 0
     Major     Minor     Lastminor       Modules
         7       ALL           -           pfil
  ```

  c. 删除 `autopush` 配置。

```
# autopush -r -M 7 -m 0
```

d. 打开设备并为设备指定 IP 地址。

```
# ifconfig hme0 unplumb
# ifconfig hme0 plumb 192.168.1.20  netmask 255.255.255.0  up
# ifconfig hme0 inet6 unplumb
# ifconfig hme0 inet6 plumb fec3:f840::1/96 up
```

有关 `ifconfig` 命令的更多信息，请参见 `ifconfig(1M)` 手册页。

▼ 如何查看 IP 过滤器的 pfil 统计信息

在对 IP 过滤器进行故障排除时，可以查看 `pfil` 统计信息。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看 `pfil` 统计信息。**

```
# ndd -get /dev/pfil qif_status
```

示例 26–1 查看 IP 过滤器的 pfil 统计信息

以下示例显示如何查看 **`pfil`** 统计信息。

```
# ndd -get /dev/pfil qif_status
ifname ill q OTHERQ num sap hl nr nw bad copy copyfail drop notip nodata
   notdata
QIF6 0 300011247b8 300011248b0 6 806 0 4 9 0 0 0 0 0 0 0 0
dmfe1 3000200a018 30002162a50 30002162b48 5 800 14 171 13681 0 0 0 0 0 0 0
```

使用 IP 过滤器规则集合

以下任务列表提供了与 IP 过滤器规则集合相关的过程。

表 26–4 使用 IP 过滤器规则集合（任务列表）

任务	说明	参考
管理、查看和修改 IP 过滤器包过滤规则集合。		第 584 页中的“管理 IP 过滤器的包过滤规则集合”
	查看活动的包过滤规则集。	第 584 页中的“如何查看活动的包过滤规则集合”

表 26–4 使用 IP 过滤器规则集合（任务列表） （续）

任务	说明	参考
	查看非活动的包过滤规则集合。	第 584 页中的 “如何查看非活动的包过滤规则集合”
	激活不同的活动规则集合。	第 585 页中的 “如何激活不同的或更新的包过滤规则集”
	删除规则集合。	第 586 页中的 “如何删除包过滤规则集合”
	将规则添加到规则集合。	第 586 页中的 “如何将规则附加到活动的包过滤规则集合”
		第 587 页中的 “如何将规则附加到非活动的包过滤规则集合”
	在活动和非活动的规则集合之间切换。	第 588 页中的 “如何在活动和非活动的包过滤规则集合之间切换”
	从内核中删除非活动规则集合。	第 589 页中的 “如何从内核中删除非活动的包过滤规则集合”
管理、查看和修改 IP 过滤器 NAT 规则。		第 589 页中的 “管理 IP 过滤器的 NAT 规则”
	查看活动的 NAT 规则。	第 589 页中的 “如何查看活动的 NAT 规则”
	删除 NAT 规则。	第 590 页中的 “如何删除 NAT 规则”
	将其他规则添加到 NAT 规则。	第 590 页中的 “如何将规则附加到 NAT 规则”
管理、查看和修改 IP 过滤器地址池。		第 591 页中的 “管理 IP 过滤器的地址池”
	查看活动的地址池。	第 591 页中的 “如何查看活动地址池”
	删除地址池。	第 592 页中的 “如何删除地址池”
	将其他规则添加到地址池。	第 592 页中的 “如何将规则附加到地址池”

管理 IP 过滤器的包过滤规则集合

启用后，活动和非活动的包过滤规则集合都可以驻留在内核中。活动规则集合确定正在对传入包和传出包执行的过滤。非活动规则集合也存储规则，但不会使用这些规则，除非使非活动规则集合成为活动规则集合。可以管理、查看和修改活动和非活动的包过滤规则集合。

▼ 如何查看活动的包过滤规则集合

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看装入到内核中的活动包过滤规则集合。**

```
# ipfstat -io
```

示例 26–2 查看活动的包过滤规则集合

以下示例显示装入到内核中的活动包过滤规则集合的输出。

```
# ipfstat -io
empty list for ipfilter(out)
pass in quick on dmfe1 from 192.168.1.0/24 to any
pass in all
block in on dmfe1 from 192.168.1.10/32 to any
```

▼ 如何查看非活动的包过滤规则集合

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看非活动的包过滤规则集合。**

```
# ipfstat -I -io
```

示例 26–3 查看非活动的包过滤规则集合

以下示例显示非活动的包过滤规则集合的输出。

```
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
```

▼ 如何激活不同的或更新的包过滤规则集

如果要执行以下任一任务，请使用以下过程：

- 激活当前 IP 过滤器正在使用的包过滤规则集合之外的另一个包过滤规则集合。
- 重新装入最近已更新的同一过滤规则集合。

1 承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

2 选择以下步骤之一：

- 如果要激活完全不同的规则集合，请在您选择的单独文件中创建一个新规则集合。
- 通过编辑包含该规则集合的配置文件来更新当前规则集合。

3 删除当前的规则集合，并装入新规则集合。

```
# ipf -Fa -f filename
```

filename 可以是包含新规则集合的新文件，也可以是包含活动规则集合的更新文件。

活动规则集合将从内核中删除。*filename* 文件中的规则将成为活动规则集合。

注 – 即使是要重新装入当前配置文件，也仍需发出该命令。否则，将继续使用旧规则集合，而不会应用更新的配置文件中的已修改规则集合。

请勿使用 `ipf -D` 或 `svcadm restart` 之类的命令来装入更新的规则集合。此类命令会在装入新规则集合之前禁用防火墙，从而会公开您的网络。

示例 26–4 激活不同的包过滤规则集合

以下示例说明如何在单独的配置文件 `/etc/ipf/ipf.conf` 中将一个包过滤规则集合替换为另一个包过滤规则集合。

```
# ipfstat -io
empty list for ipfilter(out)
pass in quick on dmfe all
# ipf -Fa -f /etc/ipf/ipf.conf
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
```

示例 26–5 重新装入更新的包过滤规则集合

以下示例说明如何重新装入当前处于活动状态且已更新的包过滤规则集合。在此示例中，使用的文件是 `/etc/ipf/ipf.conf`。

```
# ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any

(Edit the /etc/ipf/ipf.conf configuration file.)

# ipf -Fa -f /etc/ipf/ipf.conf
# ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any
block in quick on elx10 from 192.168.0.0/12 to any
```

▼ 如何删除包过滤规则集合

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **删除规则集合。**

   ```
   # ipf -F [a|i|o]
   ```

   `-a` 从规则集合中删除所有过滤规则。

   `-i` 删除传入包的过滤规则。

   `-o` 删除传出包的过滤规则。

示例 26–6 删除包过滤规则集合

以下示例显示如何从活动的过滤规则集合中删除所有过滤规则。

```
# ipfstat -io
block out log on dmf0 all
block in log quick from 10.0.0.0/8 to any
# ipf -Fa
# ipfstat -io
empty list for ipfilter(out)
empty list for ipfilter(in)
```

▼ 如何将规则附加到活动的包过滤规则集合

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **使用以下方法之一将规则附加到活动规则集合：**

- 在命令行上使用 `ipf -f -` 命令，将规则附加到规则集合。

  ```
  # echo "block in on dmfe1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
  ```

- 执行以下命令：

  a. 在所选的文件中创建规则集合。

  b. 将已创建的规则添加到活动规则集合。

  ```
  # ipf -f filename
  ```

  filename 中的规则将添加到活动规则集合的结尾。由于 IP 过滤器使用“最后一个匹配规则”算法，因此，除非使用 `quick` 关键字，否则所添加的规则将确定过滤优先级。如果包与包含 `quick` 关键字的规则匹配，则执行该规则的操作，且不检查后续规则。

示例 26–7 将规则附加到活动的包过滤规则集合

以下示例显示如何从命令行将规则添加到活动的包过滤规则集合。

```
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
# echo "block in on dmfe1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
```

▼ 如何将规则附加到非活动的包过滤规则集合

1 承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 在所选的文件中创建规则集合。

3 将已创建的规则添加到非活动规则集合。

```
# ipf -I -f filename
```

filename 中的规则将添加到非活动规则集合的结尾。由于 IP 过滤器使用“最后一个匹配规则”算法，因此，除非使用 `quick` 关键字，否则所添加的规则将确定过滤优先级。如果包与包含 `quick` 关键字的规则匹配，则执行该规则的操作，且不检查后续规则。

示例 26–8 将规则附加到非活动规则集合

以下示例显示如何将规则从文件添加到非活动规则集合。

```
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
# ipf -I -f /etc/ipf/ipf.conf
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
```

▼ 如何在活动和非活动的包过滤规则集合之间切换

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **在活动和非活动的规则集合之间切换。**

   ```
   # ipf -s
   ```

   使用此命令，可以在内核中活动和非活动的规则集合之间切换。请注意，如果非活动规则集合为空，则没有包过滤。

示例 26–9 在活动和非活动的包过滤规则集合之间切换

以下示例显示使用 `ipf -s` 命令如何导致非活动规则集合成为活动规则集合，并导致活动规则集合成为非活动规则集合。

- 运行 `ipf -s` 命令之前，`ipfstat -I -io` 命令的输出显示非活动规则集合中的规则。`ipfstat -io` 命令的输出显示活动规则集合中的规则。

  ```
  # ipfstat -io
  empty list for ipfilter(out)
  block in log quick from 10.0.0.0/8 to any
  block in on dmfe1 proto tcp from 10.1.1.1/32 to any
  # ipfstat -I -io
  pass out quick on dmfe1 all
  pass in quick on dmfe1 all
  block in log quick from 10.0.0.0/8 to any
  ```

- 运行 `ipf -s` 命令后，`ipfstat -I -io` 和 `ipfstat -io` 命令的输出显示两个规则集合的内容已设置。

  ```
  # ipf -s
  Set 1 now inactive
  # ipfstat -io
  pass out quick on dmfe1 all
  pass in quick on dmfe1 all
  block in log quick from 10.0.0.0/8 to any
  # ipfstat -I -io
  empty list for inactive ipfilter(out)
  block in log quick from 10.0.0.0/8 to any
  ```

```
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
```

▼ 如何从内核中删除非活动的包过滤规则集合

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **在“全部刷新”命令中指定非活动规则集合。**

```
# ipf -I -Fa
```

此命令从内核中清除非活动规则集合。

注 – 如果随后运行 `ipf -s`，则空的非活动规则集合将成为活动规则集合。空的活动规则集合意味着不会执行过滤。

示例 26–10 从内核中删除非活动的包过滤规则集合

以下示例显示如何清除非活动的包过滤规则集以便删除所有规则。

```
# ipfstat -I -io
empty list for inactive ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
# ipf -I -Fa
# ipfstat -I -io
empty list for inactive ipfilter(out)
empty list for inactive ipfilter(in)
```

管理 IP 过滤器的 NAT 规则

使用以下过程可以管理、查看和修改 NAT 规则。

▼ 如何查看活动的 NAT 规则

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看活动的 NAT 规则。**

```
# ipnat -l
```

示例 26–11 查看活动的 NAT 规则

以下示例显示活动 NAT 规则集合的输出。

```
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
```

▼ 如何删除 NAT 规则

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **删除当前的 NAT 规则。**

```
# ipnat -C
```

示例 26–12 删除 NAT 规则

以下示例显示如何删除当前 NAT 规则中的项。

```
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
# ipnat -C
1 entries flushed from NAT list
# ipnat -l
List of active MAP/Redirect filters:

List of active sessions:
```

▼ 如何将规则附加到 NAT 规则

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **使用以下方法之一将规则附加到活动规则集合：**

- 在命令行上使用 `ipnat -f -` 命令，将规则附加到 NAT 规则集合。

  ```
  # echo "map dmfe0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
  ```

- 执行以下命令：

  a. 在所选的文件中创建其他 NAT 规则。

  b. 将已创建的规则添加到活动的 NAT 规则。

  ```
  # ipnat -f filename
  ```

  filename 中的规则将添加到 NAT 规则的结尾。

示例 26–13 将规则附加到 NAT 规则集合

以下示例显示如何从命令行将规则添加到 NAT 规则集。

```
# ipnat -l
List of active MAP/Redirect filters:

List of active sessions:
# echo "map dmfe0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
```

管理 IP 过滤器的地址池

使用以下过程可以管理、查看和修改地址池。

▼ 如何查看活动地址池

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看活动地址池。**

```
# ippool -l
```

示例 26–14 查看活动地址池

以下示例显示如何查看活动地址池的内容。

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

▼ 如何删除地址池

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **删除当前地址池中的项。**

   ```
   # ippool -F
   ```

示例 26–15 删除地址池

以下示例显示如何删除地址池。

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
# ippool -F
1 object flushed
# ippool -l
```

▼ 如何将规则附加到地址池

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **使用以下方法之一将规则附加到活动规则集合：**

   - 在命令行上使用 `ippool -f -` 命令，将规则附加到规则集合。

     ```
     # echo "table role = ipf type = tree number = 13
     {10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24};" | ippool -f -
     ```

   - 执行以下命令：

     a. 在所选的文件中创建其他地址池。

     b. 将已创建的规则添加到活动地址池。

        ```
        # ippool -f filename
        ```

        filename 中的规则将添加到活动地址池的结尾。

示例 26–16 将规则附加到地址池

以下示例显示如何从命令行将地址池添加到地址池规则集。

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
# echo "table role = ipf type = tree number = 100
 {10.0.0.0/32, 172.16.1.2/32, 192.168.1.0/24};" | ippool -f -
# ippool -l
table role = ipf type = tree number = 100
        { 10.0.0.0/32, 172.16.1.2/32, 192.168.1.0/24; };
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

显示 IP 过滤器的统计信息

表 26–5 显示 IP 过滤器的统计信息（任务列表）

任务	说明	参考
查看状态表。	使用 `ipfstat` 命令查看状态表，以获取有关包过滤的信息。	第 593 页中的 “如何查看 IP 过滤器的状态表”
查看状态统计信息。	使用 `ipfstat -s` 命令查看有关包状态信息的统计信息。	第 594 页中的 “如何查看 IP 过滤器的状态统计信息”
查看 NAT 统计信息。	使用 `ipnat -s` 命令查看 NAT 统计信息。	第 595 页中的 “如何查看 IP 过滤器的 NAT 统计信息”
查看地址池统计信息。	使用 `ippool -s` 命令查看地址池统计信息。	第 595 页中的 “如何查看 IP 过滤器的地址池统计信息”

▼ 如何查看 IP 过滤器的状态表

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看状态表。**

```
# ipfstat
```

注 – 可以使用 `-t` 选项以 top 实用程序格式查看状态表。

示例 26–17 查看 IP 过滤器的状态表

以下示例显示如何查看状态表。

```
# ipfstat
bad packets:            in 0    out 0
 input packets:         blocked 160 passed 11 nomatch 1 counted 0 short 0
output packets:         blocked 0 passed 13681 nomatch 6844 counted 0 short 0
 input packets logged:  blocked 0 passed 0
output packets logged:  blocked 0 passed 0
 packets logged:        input 0 output 0
 log failures:          input 0 output 0
fragment state(in):     kept 0  lost 0
fragment state(out):    kept 0  lost 0
packet state(in):       kept 0  lost 0
packet state(out):      kept 0  lost 0
ICMP replies:   0       TCP RSTs sent:  0
Invalid source(in):     0
Result cache hits(in):  152     (out):  6837
IN Pullups succeeded:   0       failed: 0
OUT Pullups succeeded:  0       failed: 0
Fastroute successes:    0       failures:       0
TCP cksum fails(in):    0       (out):  0
IPF Ticks:      14341469
Packet log flags set: (0)
        none
```

▼ 如何查看 IP 过滤器的状态统计信息

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **查看状态统计信息。**

   ```
   # ipfstat -s
   ```

示例 26–18 查看 IP 过滤器的状态统计信息

以下示例显示如何查看状态统计信息。

```
# ipfstat -s
IP states added:
        0 TCP
        0 UDP
        0 ICMP
        0 hits
        0 misses
        0 maximum
        0 no memory
```

```
        0 max bucket
        0 active
        0 expired
        0 closed
State logging enabled

State table bucket statistics:
        0 in use
        0.00% bucket usage
        0 minimal length
        0 maximal length
        0.000 average length
```

▼ 如何查看 IP 过滤器的 NAT 统计信息

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看 NAT 统计信息。**

```
# ipnat -s
```

示例 26–19 查看 IP 过滤器的 NAT 统计信息

以下示例显示如何查看 NAT 统计信息。

```
# ipnat -s
mapped  in      0       out     0
added   0       expired 0
no memory       0       bad nat 0
inuse   0
rules   1
wilds   0
```

▼ 如何查看 IP 过滤器的地址池统计信息

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看地址池统计信息。**

```
# ippool -s
```

示例 26–20 查看 IP 过滤器的地址池统计信息

以下示例说明如何查看地址池统计信息。

```
# ippool -s
Pools:  3
Hash Tables:    0
Nodes:  0
```

处理 IP 过滤器的日志文件

表 26–6 使用 IP 过滤器的日志文件（任务列表）

任务	说明	参考
创建日志文件。	创建单独的 IP 过滤器日志文件。	第 596 页中的 “如何为 IP 过滤器设置日志文件”
查看日志文件。	使用 `ipmon` 命令查看状态日志文件、NAT 日志文件和常规日志文件。	第 597 页中的 “如何查看 IP 过滤器的日志文件”
清除包日志缓冲区。	使用 `ipmon -F` 命令删除包日志缓冲区的内容。	第 598 页中的 “如何清除包日志文件”
将记录的包保存到文件中。	将记录的包保存到文件中，以供日后参考。	第 599 页中的 “如何将记录的包保存到文件中”

▼ 如何为 IP 过滤器设置日志文件

缺省情况下，IP 过滤器的所有日志信息都记录在 `syslogd` 文件中。应设置一个日志文件来单独记录 IP 过滤器流量信息，以将其与可能记录在缺省日志文件中的其他数据相区分。请执行以下步骤。

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **通过添加以下两行来编辑 `/etc/syslog.conf` 文件：**

   ```
   # Save IP Filter log output to its own file
   local0.debug             /var/log/log-name
   ```

注 – 在第二行中，确保使用 Tab 键而不是 Spacebar 来分隔 `local0.debug` 与 `/var/log/`*log-name*。

3 **创建新日志文件。**

```
# touch /var/log/log-name
```

4 **重新启动系统日志服务。**

```
# svcadm restart system-log
```

示例 26–21 **创建 IP 过滤器日志**

以下示例说明如何创建 `ipmon.log` 以归档 IP 过滤器信息。

在 `/etc/syslog.conf` 中：

```
# Save IP Filter log output to its own file
local0.debug             /var/log/ipmon.log
```

在命令行中：

```
# touch /var/log/ipmon.log
# svcadm restart system-log
```

▼ 如何查看 IP 过滤器的日志文件

开始之前 应创建一个单独的日志文件来记录 IP 过滤器数据。请参阅第 596 页中的 “如何为 IP 过滤器设置日志文件”。

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **查看状态日志文件、NAT 日志文件或常规日志文件。要查看日志文件，请键入以下命令，并使用适当的选项：**

```
# ipmon -o [S|N|I] filename
```

`S` 显示状态日志文件。

`N` 显示 NAT 日志文件。

`I` 显示常规 IP 日志文件。

要查看所有状态日志文件、NAT 日志文件和常规日志文件，请使用所有选项：

```
# ipmon -o SNI filename
```

- **如果您已事先手动停止了 ipmon 守护进程，则还可以使用以下命令来显示状态日志文件、NAT 日志文件和 IP 过滤器日志文件：**

  ```
  # ipmon -a filename
  ```

  注 – 如果 ipmon 守护进程仍在运行，请勿使用 ipmon -a 语法。通常，该守护进程会在系统引导期间自动启动。发出 ipmon -a 命令还会打开 ipmon 的另一个副本。在此情况下，两个副本将读取相同的日志信息，只有一个副本会获得特定日志消息。

有关查看日志文件的更多信息，请参见 ipmon(1M) 手册页。

示例 26–22 查看 IP 过滤器的日志文件

以下示例显示了来自 /var/ipmon.log 的输出。

```
# ipmon -o SNI /var/ipmon.log
02/09/2004 15:27:20.606626 hme0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

或

```
# pkill ipmon
# ipmon -aD /var/ipmon.log
02/09/2004 15:27:20.606626 hme0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

▼ 如何清除包日志文件

1. **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

   可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2. **刷新包日志缓冲区。**

   ```
   # ipmon -F
   ```

示例 26–23 清除包日志文件

以下示例显示删除日志文件时的输出。即使未在日志文件中存储任何内容，系统也将提供一个报告，如此示例所示。

```
# ipmon -F
0 bytes flushed from log buffer
0 bytes flushed from log buffer
0 bytes flushed from log buffer
```

▼ 如何将记录的包保存到文件中

1 **承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。**

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的"Configuring RBAC (Task Map)"。

2 **将记录的包保存到文件中。**

```
# cat /dev/ipl > filename
```

继续将包记录到 *filename* 文件中，直到您通过键入 `Ctrl-C` 组合键使命令行提示符重新出现来中断该过程。

示例 26–24 将记录的包保存到文件中

以下示例显示将记录的包保存到文件中时所出现的结果。

```
# cat /dev/ipl > /tmp/logfile
^C#

# ipmon -f /tmp/logfile
02/09/2004 15:30:28.708294 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 52 -S IN
02/09/2004 15:30:28.708708 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.792611 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 70 -AP IN
02/09/2004 15:30:28.872000 hme0 @0:1 p 129.146.157.149,33923 ->
 129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.872142 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 43 -AP IN
02/09/2004 15:30:28.872808 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.872951 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 47 -AP IN
02/09/2004 15:30:28.926792 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
.
.
(output truncated)
```

创建和编辑 IP 过滤器配置文件

必须直接编辑配置文件以创建和修改规则集合及地址池。配置文件遵循标准的 UNIX 语法规则：

- 井号 (#) 指示包含注释的行。
- 规则和注释可以共存于同一行上。
- 允许使用额外的空格来增强规则的可读性。
- 规则可以延续多行。在行尾使用反斜杠 (\) 以指示规则在下一行上继续。

▼ 如何为 IP 过滤器创建配置文件

以下过程介绍如何设置以下文件：

- 包过滤配置文件
- NAT 规则配置文件
- 地址池配置文件

1 承担拥有 IP 过滤器管理权限配置文件的角色，或者成为超级用户。

可以将 IP Filter Management（IP 过滤器管理）权限配置文件指定给您创建的角色。要创建该角色并将其指定给用户，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 启动所选的文件编辑器。为要配置的功能创建或编辑配置文件。

- 要为包过滤规则创建配置文件，请编辑 `ipf.conf` 文件。

  IP 过滤器使用放置在 `ipf.conf` 文件中的包过滤规则。如果在 `/etc/ipf/ipf.conf` 文件中放置包过滤的规则文件，则在引导系统时会装入此文件。如果不希望在引导时装入过滤规则，请将其放置在所选的文件中。然后可以使用 `ipf` 命令激活规则，如第 585 页中的 “如何激活不同的或更新的包过滤规则集”中所述。

  有关创建包过滤规则的信息，请参见第 563 页中的 “使用 IP 过滤器的包过滤功能”。

  注 – 如果 `ipf.conf` 文件为空，则没有过滤。空的 `ipf.conf` 文件相当于具有以下规则集合：

  ```
  pass in all
  pass out all
  ```

- 要为 NAT 规则创建配置文件，请编辑 `ipnat.conf` 文件。

IP 过滤器使用放置在 ipnat.conf 文件中的 NAT 规则。如果在 /etc/ipf/ipnat.conf 文件中放置 NAT 的规则文件，则在引导系统时会装入此文件。如果不希望在引导时装入 NAT 规则，请将 ipnat.conf 文件放置在所选的位置中。然后可以使用 ipnat 命令激活 NAT 规则。

有关为 NAT 创建规则的信息，请参见第 565 页中的 “使用 IP 过滤器的 NAT 功能”。

- 要为地址池创建配置文件，请编辑 ippool.conf 文件。

  IP 过滤器使用放置在 ippool.conf 文件中的地址池。如果在 /etc/ipf/ippool.conf 文件中放置地址池的规则文件，则在引导系统时会装入此文件。如果不希望在引导时装入地址池，请将 ippool.conf 文件放置在所选的位置中。然后可以使用 ippool 命令激活地址池。

  有关创建地址池的信息，请参见第 566 页中的 “使用 IP 过滤器的地址池功能”。

IP 过滤器配置文件示例

以下示例说明了在过滤配置中使用的包过滤规则。

示例 26–25 IP 过滤器主机配置

此示例显示了具有 elxl 网络接口的主机上的配置。

```
# pass and log everything by default
pass in log on bge0 all
pass out log on bge0 all

# block, but don't log, incoming packets from other reserved addresses
block in quick on bge0 from 10.0.0.0/8 to any
block in quick on bge0 from 172.16.0.0/12 to any

# block and log untrusted internal IPs. 0/32 is notation that replaces
# address of the machine running Solaris IP Filter.
block in log quick from 192.168.1.15 to <thishost>
block in log quick from 192.168.1.43 to <thishost>

# block and log X11 (port 6000) and remote procedure call
# and portmapper (port 111) attempts
block in log quick on bge0 proto tcp from any to bge0/32 port = 6000 keep state
block in log quick on bge0 proto tcp/udp from any to bge0/32 port = 111 keep state
```

此规则集合以两个无限制规则开始，分别允许将任何内容传入和传出 elxl 接口。第二个规则集合阻止从专用地址空间 10.0.0.0 和 172.16.0.0 传入的任何包进入防火墙。下一个规则集合阻止来自主机的特定内部地址。最后一个规则集合阻止从端口 6000 和端口 111 上传入的包。

示例 26–26 IP 过滤器服务器配置

此示例显示用作 Web 服务器的主机的配置。此计算机具有 eri 网络接口。

示例 26–26 IP 过滤器服务器配置 （续）

```
# web server with an eri interface
# block and log everything by default; then allow specific services
# group 100 - inbound rules
# group 200 - outbound rules
# (0/32) resolves to our IP address)
*** FTP proxy ***


# block short packets which are packets fragmented too short to be real.
block in log quick all with short


# block and log inbound and outbound by default, group by destination
block in log on eri0 from any to any head 100
block out log on eri0 from any to any head 200


# web rules that get hit most often
pass in quick on eri0 proto tcp from any \
to eri0/32 port = http flags S keep state group 100
pass in quick on eri0 proto tcp from any \
to eri0/32 port = https flags S keep state group 100


# inbound traffic - ssh, auth
pass in quick on eri0 proto tcp from any \
to eri0/32 port = 22 flags S keep state group 100
pass in log quick on eri0 proto tcp from any \
to eri0/32 port = 113 flags S keep state group 100
pass in log quick on eri0 proto tcp from any port = 113 \
to eri0/32 flags S keep state group 100


# outbound traffic - DNS, auth, NTP, ssh, WWW, smtp
pass out quick on eri0 proto tcp/udp from eri0/32 \
to any port = domain flags S keep state group 200
pass in quick on eri0 proto udp from any port = domain to eri0/32 group 100

pass out quick on eri0 proto tcp from eri0/32 \
to any port = 113 flags S keep state group 200
pass out quick on eri0 proto tcp from eri0/32 port = 113 \
to any flags S keep state group 200

pass out quick on eri0 proto udp from eri0/32 to any port = ntp group 200
pass in quick on eri0 proto udp from any port = ntp to eri0/32 port = ntp group 100

pass out quick on eri0 proto tcp from eri0/32 \
to any port = ssh flags S keep state group 200

pass out quick on eri0 proto tcp from eri0/32 \
to any port = http flags S keep state group 200
pass out quick on eri0 proto tcp from eri0/32 \
to any port = https flags S keep state group 200

pass out quick on eri0 proto tcp from eri0/32 \
```

示例 26–26 IP 过滤器服务器配置 （续）

```
to any port = smtp flags S keep state group 200


# pass icmp packets in and out
pass in quick on eri0 proto icmp from any to eri0/32  keep state group 100
pass out quick on eri0 proto icmp from eri0/32 to any keep state group 200


# block and ignore NETBIOS packets
block in quick on eri0 proto tcp from any \
to any port = 135 flags S keep state group 100

block in quick on eri0 proto tcp from any port = 137 \
to any flags S keep state group 100
block in quick on eri0 proto udp from any to any port = 137 group 100
block in quick on eri0 proto udp from any port = 137 to any group 100

block in quick on eri0 proto tcp from any port = 138 \
to any flags S keep state group 100
block in quick on eri0 proto udp from any port = 138 to any group 100

block in quick on eri0 proto tcp from any port = 139 to any flags S keep state
group 100
block in quick on eri0 proto udp from any port = 139 to any group 100
```

示例 26–27 IP 过滤器路由器配置

此示例显示具有内部接口 `ce0` 和外部接口 `ce1` 的路由器的配置。

```
# internal interface is ce0 at 192.168.1.1
# external interface is ce1 IP obtained via DHCP
# block all packets and allow specific services
*** NAT ***
*** POOLS ***


# Short packets which are fragmented too short to be real.
block in log quick all with short


# By default, block and log everything.
block in log on ce0 all
block in log on ce1 all
block out log on ce0 all
block out log on ce1 all


# Packets going in/out of network interfaces that aren't on the loopback
# interface should not exist.
block in log quick on ce0 from 127.0.0.0/8 to any
block in log quick on ce0 from any to 127.0.0.0/8
block in log quick on ce1 from 127.0.0.0/8 to any
block in log quick on ce1 from any to 127.0.0.0/8
```

示例 26–27 IP 过滤器路由器配置 （续）

```
# Deny reserved addresses.
block in quick on ce1 from 10.0.0.0/8 to any
block in quick on ce1 from 172.16.0.0/12 to any
block in log quick on ce1 from 192.168.1.0/24 to any
block in quick on ce1 from 192.168.0.0/16 to any


# Allow internal traffic
pass in quick on ce0 from 192.168.1.0/24 to 192.168.1.0/24
pass out quick on ce0 from 192.168.1.0/24 to 192.168.1.0/24


# Allow outgoing DNS requests from our servers on .1, .2, and .3
pass out quick on ce1 proto tcp/udp from ce1/32 to any port = domain keep state
pass in quick on ce0 proto tcp/udp from 192.168.1.2 to any port = domain keep state
pass in quick on ce0 proto tcp/udp from 192.168.1.3 to any port = domain keep state


# Allow NTP from any internal hosts to any external NTP server.
pass in quick on ce0 proto udp from 192.168.1.0/24 to any port = 123 keep state
pass out quick on ce1 proto udp from any to any port = 123 keep state


# Allow incoming mail
pass in quick on ce1 proto tcp from any to ce1/32 port = smtp keep state
pass in quick on ce1 proto tcp from any to ce1/32 port = smtp keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = smtp keep state


# Allow outgoing connections: SSH, WWW, NNTP, mail, whois
pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = 22 keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = 22 keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = 443 keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = 443 keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = nntp keep state
block in quick on ce1 proto tcp from any to any port = nntp keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = nntp keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = smtp keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = whois keep state
pass out quick on ce1 proto tcp from any to any port = whois keep state


# Allow ssh from offsite
pass in quick on ce1 proto tcp from any to ce1/32 port = 22 keep state


# Allow ping out
pass in quick on ce0 proto icmp all keep state
```

示例 26–27 IP 过滤器路由器配置 （续）

```
pass out quick on ce1 proto icmp all keep state


# allow auth out
pass out quick on ce1 proto tcp from ce1/32 to any port = 113 keep state
pass out quick on ce1 proto tcp from ce1/32 port = 113 to any keep state


# return rst for incoming auth
block return-rst in quick on ce1 proto tcp from any to any port = 113 flags S/SA


# log and return reset for any TCP packets with S/SA
block return-rst in log on ce1 proto tcp from any to any flags S/SA


# return ICMP error packets for invalid UDP packets
block return-icmp(net-unr) in proto udp all
```

第 5 部 分

IPMP

本部分介绍 IP 网络多路径 (IP Network Multipathing, IPMP) 以及有关管理 IPMP 的任务。IPMP 为系统上连接到同一链路的接口提供故障检测和故障转移。

◆ ◆ ◆ 第 27 章

27 IPMP 介绍（概述）

IP 网络多路径 (IP Network Multipathing, IPMP) 为在同一 IP 链路上具有多个接口的系统提供物理接口故障检测和透明网络访问故障转移功能。IPMP 还为具有多个接口的系统提供了包负荷分配。

本章包含以下信息：

有关 IPMP 配置任务，请参阅第 28 章，管理 IPMP（任务）。

为什么应该使用 IPMP

IPMP 为具有多个物理接口的系统提供增强的可靠性、可用性和网络性能。有时，连接到该接口的物理接口或网络硬件可能会出现故障或者需要维护。过去，在这种情况下，系统便无法再通过与故障接口关联的任何 IP 地址进行联系。并且，使用这些 IP 地址与系统相连的任何现有连接都将被断开。

通过 IPMP，可以将一个或多个物理接口配置到 IP 多路径组（*IPMP* **组**）中。配置 IPMP 后，系统将自动监视 IPMP 组中的接口是否出现故障。如果组中的接口出现故障或者被移除以进行维护，则 IPMP 自动迁移或**故障转移**故障接口的 IP 地址。这些地址的接收者是故障接口的 IPMP 组中的工作接口。IPMP 的故障转移功能可以保持连接并防止断开任何现有连接。此外，通过自动在 IPMP 组中的一组接口中分配网络通信流量，IPMP 提高了总体网络性能。此过程称为**负荷分配**。

Oracle Solaris IPMP 组件

Oracle Solaris IPMP 涉及以下软件：

- `in.mpathd` 守护进程（在 `in.mpathd(1M)` 手册页中进行了完整说明）。
- `/etc/default/mpathd` 配置文件（也在 `in.mpathd(1M)` 手册页中进行了说明）。
- `ifconfig` 用于 IPMP 配置的选项，如 `ifconfig(1M)` 手册页中所述。

多路径守护进程 `in.mpathd`

`in.mpathd` 守护进程检测接口故障，然后实现故障转移和故障恢复的各种过程。在 `in.mpathd` 检测到故障或修复后，守护进程将发送 `ioctl` 以执行故障转移或故障恢复。`ip` 内核模块（用于实现 `ioctl`）将自动且透明地进行网络访问故障转移。

注 – 在同一组网络接口卡上使用 IPMP 时，不要使用 Alternate Pathing。同样，在使用 Alternate Pathing 时，不应使用 IPMP。可以在不同的接口组上同时使用 Alternate Pathing 和 IPMP。有关 Alternate Pathing 的更多信息，请参阅《Sun Enterprise Server Alternate Pathing 2.3.1 User Guide》。

`in.mpathd` 守护进程通过在属于 IPMP 组的所有接口上发出探测器来检测故障和修复。`in.mpathd` 守护进程还通过监视组中每个接口上的 `RUNNING` 标志来检测故障和修复。有关更多信息，请参阅 `in.mpathd(1M)` 手册页。

注 – 不支持 DHCP 管理 IPMP 数据地址。如果尝试对这些地址使用 DHCP，DHCP 最终会放弃对这些地址的控制。请勿对数据地址使用 DHCP。

IPMP 术语和概念

本小节介绍本书 IPMP 章节中涉及的术语和概念。

IP 链路

在 IPMP 术语中，*IP* **链路**是一种通信工具或介质，节点可以通过它在 Internet 协议套件的数据链路层上进行通信。IP 链路的类型可能包括简单以太网、桥接以太网、集线器或异步传输模式 (Asynchronous Transfer Mode, ATM) 网络。IP 链路可以具有一个或多个 IPv4 子网号和（如果适用）一个或多个 IPv6 子网前缀。不能将一个子网号或前缀指定给多个 IP 链路。在 ATM LANE 中，一个 IP 链路便是一个仿真局域网 (Local Area Network, LAN)。对于地址解析协议 (Address Resolution Protocol, ARP)，其作用范围是单个 IP 链路。

注 – 其他与 IP 相关的文档（如 RFC 2460，Internet Protocol, Version 6 (IPv6) Specification，使用术语**链路**而非 *IP* **链路**。第 VI 部分使用术语 *IP* **链路**以避免与 IEEE 802 相混淆。在 IEEE 802 中，**链路**是指从以太网网络接口卡 (Network Interface Card, NIC) 到以太网交换机的一根线。

物理接口

物理接口提供系统与 IP 链路的连接。此连接通常实现为设备驱动程序和 NIC。如果系统具有连接到同一链路的多个接口，则可以将 IPMP 配置为在其中某个接口出现故障时执行故障转移。有关物理接口的更多信息，请参阅第 615 页中的 “IPMP 接口配置”。

网络接口卡

网络接口卡是一个可以内置到系统中的网络适配器。NIC 也可以是一个单独的卡，以用作从系统到 IP 链路的接口。一些 NIC 可以具有多个物理接口。例如，`qfe` NIC 可以具有四个接口：`qfe0` 到 `qfe3`。

IPMP 组

IP 多路径组（*IPMP* 组）由同一系统中使用同一 IPMP 组名称配置的一个或多个物理接口组成。IPMP 组中的所有接口都必须连接到同一 IP 链路。同一（非空）字符串 IPMP 组名称用于标识组中的所有接口。只要 NIC 属于同一类型，就可以将不同速度 NIC 中的接口放置在同一 IPMP 组中。例如，可以在同一组中配置 100 MB 以太网 NIC 的接口和 1 GB 以太网 NIC 的接口。再假定您有两个 100 MB 的以太网 NIC。可以将其中一个接口配置为 10 MB，并且仍将这两个接口放置在同一 IPMP 组中。

不能将介质类型不同的两个接口放置到一个 IPMP 组中。例如，不能将 ATM 接口与以太网接口放置在同一组中。

故障检测和故障转移

故障检测是检测一个接口或从接口到 Internet 层设备的路径何时不再工作的过程。IPMP 为系统提供检测接口何时出现故障的功能。

IPMP 检测以下类型的通信故障：

- 接口的传输或接收路径出现故障。
- 接口到 IP 链路的连接已关闭。
- 交换机上的端口不传输或接收包。
- IPMP 组中的物理接口在系统引导时不存在。

检测到故障后，IPMP 开始进行故障转移。**故障转移**是将网络访问从出现故障的接口切换到同一组中正常工作的物理接口的自动过程。网络访问除包括 IPv6 单播和多播通信外，还包括 IPv4 单播、多播和广播通信。仅当在 IPMP 组中配置了多个接口时，才可以发生故障转移。故障转移过程可确保对网络的不间断访问。

修复检测和故障恢复

修复检测是检测 NIC 或从 NIC 到某个 Internet 层设备的路径在出现故障后何时开始正常工作的过程。在检测到已修复的 NIC 后，IPMP 执行**故障恢复**（将网络访问切换回已修复接口的过程）。修复检测假定已启用故障恢复。有关更多信息，请参见第 618 页中的 “检测物理接口修复”。

目标系统

基于探测器的故障检测使用**目标系统**确定接口的状态。每个目标系统都必须连接到与 IPMP 组成员相同的 IP 链路。本地系统上的 `in.mpathd` 守护进程将 ICMP 探测器消息发送到每个目标系统。探测器消息有助于确定 IPMP 组中每个接口的运行状况。

有关在基于探测器的故障检测中使用目标系统的更多信息，请参阅第 617 页中的 “基于探测器的故障检测”。

外发负荷分配

在配置 IPMP 后，可以在多个 NIC 中分配外发网络包，而不会影响包的排序。此过程称为**负荷分配**。通过负荷分配可以达到较高的吞吐量。仅当网络通信流向使用多个连接的多个目标时，才会发生负荷分配。

动态重新配置

动态重新配置 (Dynamic Reconfiguration, DR) 是指在系统运行时重新配置系统而对现有操作影响很小或者没有影响的能力。并非所有 Sun 平台都支持 DR。有些 Sun 平台可能仅支持某些类型硬件的 DR。在支持 NIC 的 DR 的平台上，可以使用 IPMP 透明地故障转移网络访问，从而为系统提供不间断的网络访问。

有关 IPMP 如何支持 DR 的更多信息，请参阅第 620 页中的 “IPMP 和动态重新配置”。

IPMP 的基本要求

IPMP 内置于 Oracle Solaris 中，无需任何特殊硬件。Oracle Solaris 支持的任何接口都可以与 IPMP 一起使用。但是，IPMP 对网络配置和拓扑有以下要求：

- IPMP 组中的所有接口都必须具有唯一的 MAC 地址。

  请注意，在缺省情况下，基于 SPARC 的系统上的网络接口都共享单个 MAC 地址。因此，必须显式更改缺省值，才能在基于 SPARC 的系统上使用 IPMP。有关更多信息，请参阅第 625 页中的 “如何规划 IPMP 组”。
- IPMP 组中的所有接口都必须具有相同的介质类型。有关更多信息，请参阅第 611 页中的 “IPMP 组”。
- IPMP 组中的所有接口都必须位于同一 IP 链路上。有关更多信息，请参阅第 611 页中的 “IPMP 组”。

注 – 不支持同一个链路层（L2 或第二层）广播域上存在多个 IPMP 组。L2 广播域通常对应于特定子网。因此对于每个子网，只能配置一个 IPMP 组。

- 根据故障检测要求，可能需要使用特定类型的网络接口，或者在每个网络接口上配置其他 IP 地址。请参阅第 617 页中的 “基于链路的故障检测”和第 617 页中的 “基于探测器的故障检测”。

IPMP 寻址

可以在 IPv4 网络以及双栈、IPv4 和 IPv6 网络中配置 IPMP 故障检测。使用 IPMP 配置的接口支持以下两种类型的地址：数据地址和测试地址。

数据地址

数据地址是在引导时指定给或通过 `ifconfig` 命令手动指定给 NIC 的接口的常规 IPv4 和 IPv6 地址。通过接口的标准 IPv4 和（如果适用）IPv6 包流量被视为**数据通信**。

测试地址

测试地址是由 `in.mpathd` 守护进程使用的特定于 IPMP 的地址。对于要使用基于探测器的故障和修复检测的接口，至少必须为其配置一个测试地址。

注 – 仅当希望使用基于探测器的故障检测时，才需要配置测试地址。

`in.mpathd` 守护进程使用测试地址与 IP 链路上的其他目标交换 ICMP 探测器（也称为**探测器通信**）。探测器通信有助于确定接口及其 NIC 的状态，其中包括接口是否已出现故障。探测器检验接口的发送和接收路径是否正常工作。

可以使用 IP 测试地址配置每个接口。对于双栈网络中的接口，可以配置 IPv4 测试地址或/和IPv6 测试地址。

在接口出现故障后，测试地址将一直保留在故障接口上，以便 `in.mpathd` 可以继续发送探测器以检查后续修复。必须专门配置测试地址，以便应用程序不会意外地使用它们。有关更多信息，请参阅第 614 页中的 “防止应用程序使用测试地址”。

有关基于探测器的故障检测的更多信息，请参阅第 617 页中的 “基于探测器的故障检测”。

IPv4 测试地址

通常，可以将子网中的任何 IPv4 地址用作测试地址。IPv4 测试地址无需是可路由的。由于 IPv4 地址是许多站点的有限资源，因此您可能希望将不可路由的 RFC 1918 专用地址用作测试地址。请注意，`in.mpathd` 守护进程与测试地址在同一子网中的其他主机仅交换 ICMP 探测器。如果使用 RFC 1918 样式的测试地址，请确保使用适当的 RFC 1918 子网中的地址配置 IP 链路上的其他系统（首选路由器）。然后 `in.mpathd` 守护进程便可以成功地与目标系统交换探测器。

IPMP 示例将来自 `192.168.0/24` 网络的 RFC 1918 地址用作 IPv4 测试地址。有关 RFC 1918 专用地址的更多信息，请参阅 RFC 1918, Address Allocation for Private Internets (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`)。

有关如何配置 IPv4 测试地址，请参阅第 626 页中的 “如何配置具有多个接口的 IPMP 组”。

IPv6 测试地址

唯一的有效 IPv6 测试地址是物理接口的链路本地地址。无需将单独的 IPv6 地址用作 IPMP 测试地址。IPv6 链路本地地址基于接口的介质访问控制 (Media Access Control, MAC) 地址。当引导时接口变成启用了 IPv6 的接口或通过 `ifconfig` 手动配置接口时，将自动配置链路本地地址。

要确定接口的链路本地地址，请在启用了 IPv6 的节点上运行 `ifconfig` *interface* 命令。检查输出中是否包含以前缀 `fe80`（链路本地前缀）开头的地址。以下 `ifconfig` 输出中的 `NOFAILOVER` 标志指示，`hme0` 接口的链路本地地址 `fe80::a00:20ff:feb9:17fa/10` 被用作测试地址。

```
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:17fa/10
```

有关链路本地地址的更多信息，请参阅第 71 页中的 “链路本地单播地址”。

如果在 IPMP 组的所有接口上同时激活了 IPv4 和 IPv6，则无需配置单独的 IPv4 测试地址。`in.mpathd` 守护进程可以将 IPv6 链路本地地址用作测试地址。

要创建 IPv6 测试地址，请参阅第 626 页中的 “如何配置具有多个接口的 IPMP 组”任务。

防止应用程序使用测试地址

配置测试地址后，需要确保应用程序未使用此地址。否则，如果接口出现故障，则无法再访问应用程序，因为在故障转移操作期间测试地址并不故障转移。要确保 IP 不选择将测试地址用于常规应用程序，请将测试地址标记为 `deprecated`。

除非应用程序显式绑定到过时的地址，否则 IPv4 不会将该地址用作源地址进行任何通信。`in.mpathd` 守护进程显式绑定到此类地址，以便发送和接收探测器通信。但是，如果应用程序未显式绑定到某个地址，且接口上唯一标记为 `UP` 的地址同时也标记为过时，则最后的做法是使用该地址作为源地址。

注 – 发生故障转移和故障恢复时，检测重复地址 (Duplicate Address Detection, DAD) 仍在运行期间，应用程序可能会接收将过时地址用作源地址的数据包。此行为是可以预料到的。通常，DAD 完成后，应用程序不再处理过时的地址。但是，对于 TCP 包，可能出现一种罕见的异常。TCP 连接选择了某个特定源地址后，在该连接用于传输期间无法改用该地址。连接用于传输的时间可能持续很长。在这种极端情况下，就有可能出现 DAD 完成后，应用程序仍继续使用过时地址的现象。

因为 IPv6 链路本地地址通常不存在于名称服务中，所以 DNS 和 NIS 应用程序不使用链路本地地址进行通信。因此，不能将 IPv6 链路本地地址标记为 `deprecated`。

不应将 IPv4 测试地址放置在 DNS 和 NIS 名称服务表中。在 IPv6 中，通常不将链路本地地址放置在名称服务表中。

IPMP 接口配置

IPMP 配置通常由同一系统上连接到同一 IP 链路的两个或更多物理接口组成。这些物理接口可能在同一 NIC 上，也可能不在同一 NIC 上。这些接口被配置为同一 IPMP 组的成员。如果系统在第二个 IP 链路上具有其他接口，则必须将这些接口配置为另一个 IPMP 组。

单个接口可以在自己的 IPMP 组中进行配置。单接口 IPMP 组的行为与具有多个接口的 IPMP 组相同。但是，对于只有一个接口的 IPMP 组，不能进行故障转移和故障恢复。

也可通过与配置 IP 接口之外的组相同的步骤在 IPMP 组内配置 VLAN。有关过程，请参见第 626 页中的 “配置 IPMP 组”。在第 612 页中的 “IPMP 的基本要求”中列出的同一要求适用于将 VLAN 配置到 IPMP 组。

注意 – 当将 VLAN 配置为 IPMP 组时，用于命名 VLAN 的约定可能导致错误。有关 VLAN 名称的更多详细信息，请参见《系统管理指南：IP 服务》中第 136 页中的 “VLAN 标记和物理连接点”。考虑四种 VLAN 示例，`bge1000`、`bge1001`、`bge2000` 和 `bge2001`。IPMP 实现需要这些 VLAN 按如下方式进行分组：`bge1000` 和 `bge1001` 属于同一 VLAN 1 中的一个组，而 `bge2000` 和 `bge2001` 属于同一 VLAN 2 中的另一个组。由于 VLAN 的名称，很容易发生诸如将属于不同链路的 VLAN 混入一个 IPMP 组的错误，例如，`bge1000` 和 `bge2000`。

IPMP 组中的待机接口

除非 IPMP 组中的某个其他接口出现故障，否则不会使用该组中的**待机接口**进行数据通信。在出现故障时，故障接口上的数据地址将迁移到待机接口。然后，会像对待其他活动接口一样对待待机接口，直到修复故障接口为止。一些故障转移可能不选择待机接口。相反，这些故障转移可能选择比待机接口具有更少配置为 UP 的数据地址的活动接口。

在待机接口上应仅配置测试地址。IPMP 不允许将数据地址添加到通过 ifconfig 命令配置为 standby 的接口。创建此类型配置的任何尝试都将失败。同样，如果将已具有数据地址的接口配置为 standby，则这些地址将自动地故障转移到 IPMP 组中的其他接口。由于存在这些限制，因此在将接口设置为 standby 之前，必须使用 ifconfig 命令将所有测试地址标记为 -deprecated 和 failover。有关如何配置待机接口，请参阅第 632 页中的 “如何为 IPMP 组配置待机接口”。

常见的 IPMP 接口配置

如第 613 页中的 “IPMP 寻址”所述，IPMP 组中的接口根据接口配置处理有规律的数据通信和探测器通信。使用 ifconfig 命令的 IPMP 选项可以创建配置。

活动接口是既传输数据通信又传输探测器通信的物理接口。通过执行第 626 页中的 “如何配置具有多个接口的 IPMP 组”或第 634 页中的 “如何配置单接口 IPMP 组”中所述的任务，可以将接口配置为“活动”。

以下是两种常见的 IPMP 配置类型：

活动-活动配置 一个双接口 IPMP 组，其中的两个接口都为“活动”，即它们始终可能既传输探测器通信又传输数据通信。

活动-待机配置 一个双接口 IPMP 组，其中一个接口被配置为 "standby"（待机）。

检查接口的状态

通过发出 ifconfig *interface* 命令，可以检查接口的状态。有关 ifconfig 状态报告的一般信息，请参阅第 178 页中的 “如何获取有关特定接口的信息”。

例如，可以使用 ifconfig 命令获取待机接口的状态。当待机接口未承载任何数据地址时，该接口具有指示其状态的 INACTIVE 标志。可以在 ifconfig 输出中该接口的状态行中看到此标志。

IPMP 故障检测和恢复功能

in.mpathd守护进程处理以下类型的故障检测：

- 基于链路的故障检测（如果 NIC 驱动程序支持该故障检测）
- 基于探测器的故障检测（配置测试地址时）
- 检测引导时缺少的接口

in.mpathd(1M) 手册页对 in.mpathd 守护进程如何处理接口故障的检测进行了完整说明。

基于链路的故障检测

只要接口支持基于链路的故障检测，会始终启用该类故障检测。Oracle Solaris 当前发行版中支持以下 Sun 网络驱动程序：

- hme
- eri
- ce
- ge
- bge
- qfe
- dmfe
- e1000g
- igb
- ixgb
- nge
- nxge
- rge
- xge

要确定第三方接口是否支持基于链路的故障检测，请参阅制造商文档。

这些网络接口驱动程序会监视接口的链路状态，并在链路状态更改时通知联网子系统。收到更改通知后，联网子系统会根据需要设置或清除该接口的 RUNNING 标志。守护进程在检测到接口的 RUNNING 标志已被清除时，会立即使该接口失效。

基于探测器的故障检测

in.mpathd 守护进程会对 IPMP 组中具有测试地址的每个接口执行基于探测器的故障检测。基于探测器的故障检测涉及使用测试地址发送和接收 ICMP 探测器消息。这些消息通过接口发送到同一 IP 链路上的一个或多个目标系统。有关测试地址的介绍，请参阅第 613 页中的 “测试地址”。有关配置测试地址的信息，请参阅第 626 页中的 “如何配置具有多个接口的 IPMP 组”。

in.mpathd 守护进程确定要动态探测的目标系统。会自动将连接到 IP 链路的路由器选为探测目标。如果在链路上不存在路由器，则 in.mpathd 会将探测器发送到链路上的相邻主机。发送到所有主机多播地址（在 IPv4 中为 224.0.0.1，在 IPv6 中为 ff02::1）的多播包可确定要用作目标系统的主机。对回显包作出响应的前几个主机将被选作探测目标。如果 in.mpathd 找不到响应 ICMP 回显包的路由器或主机，则 in.mpathd 无法检测基于探测器的故障。

可以使用主机路由显式配置要由 in.mpathd 使用的目标系统的列表。有关说明，请参阅第 630 页中的 “配置目标系统”。

为确保 IPMP 组中的每个接口都正常工作，in.mpathd 将通过 IPMP 组中的所有接口分别探测所有目标。如果对五个连续的探测器未做出任何响应，则 in.mpathd 认为接口已出现故障。探测速率取决于**故障检测时间** (Failure Detection Time, FDT)。故障检测时间的缺省值是 10 秒。不过，可以在 /etc/default/mpathd 文件中调整故障检测时间。有关说明，请转至第 642 页中的 “如何配置 /etc/default/mpathd 文件”。

对于 10 秒的修复检测时间，探测速率约为每两秒发送一个探测器。最短的修复检测时间是故障检测时间的两倍，缺省情况下为 20 秒，因为必须收到对 10 个连续探测器的回复。故障检测时间和修复检测时间仅适用于基于探测器的故障检测。

注 – 由 VLAN 组成的 IPMP 组，基于链路的故障检测通过物理链路实现，因此影响该链路上的所有 VLAN。基于探测器的故障检测通过 VLAN 链路执行。例如，同时在一个组内配置bge0/bge1 和 bge1000/bge1001。如果已拔出 bge0 的电缆，那么基于链路的故障检测将立即报告 bge0 和 bge1000 都好像已发生故障。但是，如果 bge0 上的所有探测器目标变得不可访问，将仅报告 bge0 发生故障，因为 bge1000 在其自己的 VLAN 上有其自己的探测器目标。

组故障

当 IPMP 组中的所有接口看起来同时出现故障时，就会出现**组故障**。对于组故障，in.mpathd 守护进程不执行故障转移。此外，当所有目标系统同时出现故障时，也不会执行故障转移。在这种情况下，in.mpathd 会清除其当前所有的目标系统并搜索新的目标系统。

检测物理接口修复

要使 in.mpathd 守护进程将某个接口视为要进行修复，必须为该接口设置 RUNNING 标志。如果使用基于探测器的故障检测，in.mpathd 守护进程必须先从接口收到对 10 个连续探测器包的响应，才会将该接口视为已修复。接口被视为已修复时，故障转移到其他接口的任何地址都将故障恢复到已修复的接口。如果接口在出现故障之前被配置为“活动”，则在修复之后该接口可以恢复发送和接收通信。

接口故障转移期间发生的情况

以下两个示例说明了一种典型配置，以及该配置在接口出现故障时如何自动更改。当 hme0 接口出现故障时，请注意所有数据地址都将从 hme0 移动到 hme1。

示例 27–1 接口出现故障之前的接口配置

```
hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
      mtu 1500 index 2
```

示例 27–1 接口出现故障之前的接口配置 （续）

```
     inet 192.168.85.19 netmask ffffff00 broadcast 192.168.85.255
     groupname test
hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
     mtu 1500
     index 2 inet 192.168.85.21 netmask ffffff00 broadcast 192.168.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
8     inet 192.168.85.20 netmask ffffff00 broadcast 192.168.85.255
     groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
     mtu 1500
     index 2 inet 192.168.85.22 netmask ffffff00 broadcast 192.168.85.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
     inet6 fe80::a00:20ff:feb9:19fa/10
     groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
     inet6 fe80::a00:20ff:feb9:1bfc/10
     groupname test
```

示例 27–2 接口出现故障之后的接口配置

```
hme0: flags=19000842<BROADCAST,RUNNING,MULTICAST,IPv4,
      NOFAILOVER,FAILED> mtu 0 index 2
      inet 0.0.0.0 netmask 0
      groupname test
hme0:1: flags=19040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
      NOFAILOVER,FAILED> mtu 1500 index 2
      inet 192.168.85.21 netmask ffffff00 broadcast 10.0.0.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 192.168.85.20 netmask ffffff00 broadcast 192.168.85.255
      groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
      NOFAILOVER> mtu 1500
      index 2 inet 192.168.85.22 netmask ffffff00 broadcast 10.0.0.255
hme1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 6
      inet 192.168.85.19 netmask ffffff00 broadcast 192.168.18.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER,FAILED> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:19fa/10
      groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:1bfc/10
      groupname test
```

可以看到，在 hme0 上设置了 FAILED 标志，以指示此接口已出现故障。还可以看到，创建了 hme1:2，而 hme1:2 最初是 hme0。然后，地址 192.168.85.19 变成可通过 hme1 进行访问。

与 192.168.85.19 关联的多播成员仍可以接收包，但是它们现在通过 hme1 接收包。当进行地址 192.168.85.19 从 hme0 到 hme1 的故障转移时，在 hme0 上创建了伪地址 0.0.0.0。由于创建了伪地址，因此仍可以访问 hme0。如果没有 hme0，hme0:1 便无法存在。在执行后续的故障恢复时，将删除伪地址。

同样，进行了 IPv6 地址从 hme0 到 hme1 的故障转移。在 IPv6 中，多播成员与接口索引关联。多播成员也从 hme0 故障转移到 hme1。而且还移动了 in.ndpd 配置的所有地址。示例中未说明此操作。

in.mpathd 守护进程继续通过故障接口 hme0 进行探测。守护进程在 20 秒的缺省修复检测时间内收到 10 个连续回复后，确定该接口已修复。由于在 hme0 上也设置了 RUNNING 标志，因此守护进程调用故障恢复。在故障恢复后，将恢复原始配置。

有关故障和修复期间在控制台上记录的所有错误消息的说明，请参见 in.mpathd(1M) 手册页。

IPMP 和动态重新配置

使用动态重新配置 (Dynamic Reconfiguration, DR) 功能，可以在系统运行的同时重新配置系统硬件（如接口）。本节介绍 DR 如何与 IPMP 交互操作。

在支持 NIC 的 DR 的系统上，可以使用 IPMP 保持连通性和防止断开现有的连接。在支持 DR 并使用 IPMP 的系统上，可以安全地连接、拆离或重新连接 NIC。这是因为 IPMP 已集成到重新配置调整管理器 (Reconfiguration Coordination Manager, RCM) 框架中。*RCM* 用于管理系统组件的动态重新配置。

通常使用 cfgadm 命令执行 DR 操作。但是，一些平台可提供其他方法。有关详细信息，请参阅平台文档。可以从以下资源中找到有关 DR 的特定文档。

表 27–1 动态重新配置的文档资源

说明	参考
有关 cfgadm 命令的详细信息	cfgadm(1M) 手册页
有关 Sun Cluster 环境中 DR 的特定信息	《Sun Cluster 3.1 System Administration Guide》
有关 Sun Fire 环境中 DR 的特定信息	《Sun Fire 880 Dynamic Reconfiguration Guide》
有关 DR 和 cfgadm 命令的介绍性信息	《System Administration Guide: Devices and File Systems》中的第 4 章 “Dynamically Configuring Devices (Tasks)”
在支持 DR 的系统上管理 IPMP 组的任务	第 638 页中的 “在支持动态重新配置的系统上替换出现故障的物理接口”

连接 NIC

可以随时使用 `ifconfig` 命令将接口添加到 IPMP 组，如第 626 页中的 “如何配置具有多个接口的 IPMP 组”所述。因此，可以检测在系统引导后连接的系统组件上的任何接口并将其添加到现有的 IPMP 组。或者，如果适当，可以将新添加的接口配置到其 IPMP 组中。

这些接口和其上配置的数据地址可供 IPMP 组立即使用。但是，为了使系统在重新引导后自动配置和使用这些接口，必须为每个新接口创建 `/etc/hostname.`*interface* 文件。有关说明，请参阅第 129 页中的 “如何在安装系统后配置物理接口”。

如果在连接接口时 `/etc/hostname.`*interface* 文件已存在，则 RCM 将根据此文件的内容自动配置接口。这样，该接口接收的配置与系统引导后其将接收的配置相同。

拆离 NIC

首先检查拆离包含 NIC 的系统组件的所有请求以确保可以保持连通性。例如，缺省情况下，无法拆离不在 IPMP 组中的 NIC。也无法拆离包含 IPMP 组中仅有的工作接口的 NIC。但是，如果必须移除系统组件，则可以使用 `cfgadm` 的 `-f` 选项覆盖此行为，如 `cfgadm(1M)` 手册页中所述。

如果检查成功，则与已拆离的 NIC 关联的数据地址将故障转移到同一组中的工作 NIC，如同被拆离的 NIC 出现了故障。拆离 NIC 时，会取消 NIC 接口上配置的所有测试地址。然后，从系统中取消检测 NIC。如果其中的任一步骤失败，或者同一系统组件上其他硬件的 DR 失败，则会将先前的配置恢复到其原始状态。您应接收到有关此事件的状态消息。否则，拆离请求成功完成。可以从系统中移除组件。未断开任何现有连接。

重新连接 NIC

RCM 记录与从正在运行的系统拆离的任何 NIC 关联的配置信息。因此，RCM 会像连接新的 NIC 一样，重新连接以前拆离的 NIC。即，RCM 仅执行检测。

但是，重新连接的 NIC 通常具有现有的 `/etc/hostname.`*interface* 文件中指定的配置。在这种情况下，RCM 会根据现有 `/etc/hostname.`*interface* 文件中指定的配置。此外，RCM 会通知 `in.mpathd` 守护进程最初在重新连接的接口上承载的每个数据地址。因此，在重新连接的接口正常工作后，其所有数据地址都将故障恢复到重新连接的接口，如同已修复该接口。

如果重新连接的 NIC 没有 `/etc/hostname.`*interface* 文件，则没有可用的配置信息。RCM 没有有关如何配置接口的信息。此情况的一个结果是，不会故障恢复以前故障转移到其他接口的地址。

系统引导时缺少的 NIC

系统引导时不存在的 NIC 是特殊的故障检测实例。在引导时，启动脚本会跟踪所有具有无法检测的 /etc/hostname.*interface* 文件的接口。在这类接口的 /etc/hostname.*interface* 文件中的所有数据地址将自动驻留在 IPMP 组中的替换接口上。

在此类事件中，会收到与以下内容类似的错误消息：

```
moving addresses from failed IPv4 interfaces: hme0 (moved to hme1)
moving addresses from failed IPv6 interfaces: hme0 (moved to hme1)
```

如果不存在备用接口，则收到与以下内容类似的错误消息：

```
moving addresses from failed IPv4 interfaces: hme0 (couldn't move;
   no alternative interface)
 moving addresses from failed IPv6 interfaces: hme0 (couldn't move;
   no alternative interface)
```

注 – 在此故障检测实例中，只有在缺少的接口的 /etc/hostname.*interface* 文件中显式指定的数据地址才会移动到备用接口。不会获取或移动通常通过其他方法（如通过 RARP 或 DHCP）获取的任何地址。

如果使用 DR 重新连接与在系统引导时缺少的接口同名的另一接口，则 RCM 将自动检测该接口。然后，RCM 根据接口的 /etc/hostname.*interface* 文件内容配置接口。最后，RCM 故障恢复所有数据地址，如同已修复接口。因此，最终的网络配置与在该接口存在的情况下引导系统时进行的配置完全相同。

◆ ◆ ◆ 第 28 章

管理 IPMP（任务）

本章介绍使用 IP 网络多路径 (IP Network Multipathing, IPMP) 管理接口组的任务。本章主要讨论以下主题：

- 第 623 页中的 “配置 IPMP（任务列表）”
- 第 624 页中的 “使用 IPMP 组获得高可用性”
- 第 635 页中的 “维护 IPMP 组”
- 第 638 页中的 “在支持动态重新配置的系统上替换出现故障的物理接口”
- 第 639 页中的 “恢复系统引导时不存在的物理接口”
- 第 641 页中的 “修改 IPMP 配置”

有关 IPMP 概念的概述，请参阅第 27 章，IPMP 介绍（概述）。

配置 IPMP（任务列表）

本节包含指向本章所介绍任务的链接。

配置和管理 IPMP 组（任务列表）

任务	说明	参考
规划 IPMP 组。	列出所有辅助信息和必需任务，然后才能配置 IPMP 组。	第 625 页中的 “如何规划 IPMP 组”
使用多个接口配置 IPMP 接口组。	将多个接口配置为 IPMP 组的成员。	第 626 页中的 “如何配置具有多个接口的 IPMP 组”
配置 IPMP 组，该组的一个接口是待机接口。	将多接口 IPMP 组的其中一个接口配置为待机接口。	第 632 页中的 “如何为 IPMP 组配置待机接口”

任务	说明	参考
配置由单个接口组成的 IPMP 组。	创建单接口 IPMP 组。	第 634 页中的 “如何配置单接口 IPMP 组”
显示物理接口所属的 IPMP 组。	说明如何从 `ifconfig` 命令的输出中获取接口的 IPMP 组的名称。	第 635 页中的 “如何显示接口的 IPMP 组成员关系”
将接口添加到 IPMP 组。	将新接口配置为现有 IPMP 组的成员。	第 636 页中的 “如何将接口添加到 IPMP 组”
从 IPMP 组中删除接口。	说明如何从 IPMP 组中删除接口。	第 636 页中的 “如何从 IPMP 组中删除接口”
将接口从现有 IPMP 组移动到其他组。	在 IPMP 组之间移动接口。	第 637 页中的 “如何将接口从一个 IPMP 组移动到另一个组”
更改 `in.mpathd` 守护进程的三个缺省设置。	定制 `in.mpathd` 守护进程的故障检测时间和其他参数。	第 642 页中的 “如何配置 `/etc/default/mpathd` 文件”

在支持动态重新配置的接口上管理 IPMP（任务列表）

任务	说明	参考
删除出现故障的接口。	删除系统上出现故障的接口。	第 638 页中的 “如何删除出现故障的物理接口（DR 分离）”
替换出现故障的接口。	替换出现故障的接口。	第 639 页中的 “如何替换出现故障的物理接口（DR 连接）”
恢复在引导时未配置的接口。	恢复出现故障的接口。	第 640 页中的 “如何恢复系统引导时不存在的物理接口”

使用 IPMP 组获得高可用性

本节提供了配置 IPMP 组的过程，并介绍如何将一个接口配置为待机接口。

规划 IPMP 组

将系统上的接口配置为 IPMP 组的一部分之前，需要进行一些配置前规划。

▼ 如何规划 IPMP 组

以下过程包括配置 IPMP 组之前要收集的规划任务和信息。不必顺序执行这些任务。

1 确定系统上的哪些接口将是 IPMP 组的一部分。

IPMP 组通常由至少两个连接到同一 IP 链路的物理接口组成。但是，如果需要，可以配置单接口 IPMP 组。有关 IPMP 组的简介，请参阅第 615 页中的 “IPMP 接口配置”。例如，可以在同一 IPMP 组下配置同一台以太网交换机或同一个 IP 子网。可以在同一 IPMP 组中配置任意数量的接口。

不能将 `ifconfig` 命令的 `group` 参数用于逻辑接口。例如，可以将 `group` 参数用于 `hme0`，但是不能用于 `hme0:1`。

2 验证组中的每个接口是否具有唯一的 MAC 地址。

有关说明，请参阅第 133 页中的 “SPARC: 如何确保接口的 MAC 地址是唯一的”。

3 为 IPMP 组选择一个名称。

任何非空的名称都适合用作组的名称。您可能希望使用一个可以标识接口所连接到的 IP 链路的名称。

4 确保在 IPMP 组中的所有接口上推送并配置了同一组 STREAMS 模块。

同一组中的所有接口必须按相同顺序配置相同的 STREAMS 模块。

a. 检查即将包含在 IPMP 组中的所有接口上 STREAMS 模块的顺序。

通过使用 `ifconfig` *interface* `modlist` 命令，可以输出 STREAMS 模块的列表。例如，以下是 `hme0` 接口的 `ifconfig` 输出：

```
# ifconfig hme0 modlist
    0 arp
    1 ip
    2 hme
```

接口通常作为网络驱动程序直接位于 IP 模块之下，如 `ifconfig hme0 modlist` 的输出中所示。它们应该不需要其他配置。

但是，某些技术（如 NCA 或 IP 过滤器）会将自身作为 STREAMS 模块插入到 IP 模块和网络驱动程序之间。同一 IPMP 组的接口的行为方式可能会出现问题。

如果 STREAMS 模块是有状态的，即使将相同模块推送到组中的所有接口上，在进行故障转移时仍可能会出现意外行为。但是，假定按相同顺序将 STREAMS 模块推送到 IPMP 组中的所有接口上，则可以使用无状态 STREAMS 模块。

b. 按 IPMP 组的标准顺序推送接口的模块。

`ifconfig` *interface* **`modinsert`** *module-name*

```
ifconfig hme0 modinsert ip
```

5 **在 IPMP 组的所有接口上使用相同的 IP 寻址格式。**

如果为 IPv4 配置了一个接口，则必须为 IPv4 配置组的所有接口。假定有一个由多个 NIC 的接口组成的 IPMP 组。如果在一个 NIC 的接口上进行 IPv6 寻址，则必须将 IPMP 组中的所有接口配置为支持 IPv6。

6 **检查 IPMP 组中的所有接口是否已连接到同一 IP 链路。**

7 **检验 IPMP 组是否包含具有不同网络介质类型的接口。**

组合在一起的接口应属于相同的接口类型，如 `/usr/include/net/if_types.h` 中定义的那样。例如，不能将以太网接口和令牌环接口组合在一个 IPMP 组中。此外，不能将令牌总线接口与异步传输模式 (Asynchronous Transfer Mode, ATM) 接口组合在同一 IPMP 组中。

8 **对于具有 ATM 接口的 IPMP，请在 LAN 仿真模式下配置 ATM 接口。**

使用经典的 IP over ATM 的接口不支持 IPMP。

配置 IPMP 组

本节包含有关具有至少两个物理接口的典型 IPMP 组的配置任务。

- 有关多接口 IPMP 组的简介，请参阅第 611 页中的 “IPMP 组”。
- 有关规划任务，请参阅第 624 页中的 “规划 IPMP 组”。
- 有关如何配置仅有一个物理接口的 IPMP 组，请参阅第 633 页中的 “配置具有单个物理接口的 IPMP 组”。

▼ 如何配置具有多个接口的 IPMP 组

将 VLAN 配置到 IPMP 组时，以下用于配置 IPMP 组的步骤也适用。

开始之前 需要已经配置将来的 IPMP 组中所有接口的 IPv4 地址，如果适用，还需配置 IPv6 地址。

注意 – 必须为每个子网或 L2 广播域只配置一个 IPMP 组。有关详细信息，请参见第 612 页中的 “IPMP 的基本要求”

1 **在要配置接口的系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **将每个物理接口放入 IPMP 组。**

```
# ifconfig interface group group-name
```

例如，要将 hme0 和 hme1 放入组 testgroup1 中，应键入以下命令：

```
# ifconfig hme0 group testgroup1
# ifconfig hme1 group testgroup1
```

请避免在组名中使用空格。ifconfig 状态显示不会显示空格。因此，请勿创建两个类似的组名，其中唯一的区别是一个名称还包含空格。如果其中一个组名包含空格，则这些组名在状态显示中看起来是相同的。

在双栈环境中，如果将某个接口的 IPv4 实例放入特定组中，则 IPv6 实例会自动放入同一组中。

3 （可选）在一个或多个物理接口上配置 IPv4 测试地址。

仅当要在特定接口上使用基于探测器的故障检测时，才需要配置测试地址。测试地址将配置为在 ifconfig 命令中指定的物理接口的逻辑接口。

如果组中的一个接口将成为待机接口，请勿在此时配置该接口的测试地址。作为第 632 页中的 “如何为 IPMP 组配置待机接口”的一部分，可以为待机接口配置一个测试地址。

请使用 ifconfig 命令的以下语法配置测试地址：

```
# ifconfig interface addif ip-address parameters -failover deprecated up
```

例如，可以为主网络接口 hme0 创建以下测试地址：

```
# ifconfig hme0 addif 192.168.85.21 netmask + broadcast + -failover deprecated up
```

此命令可为主网络接口 hme0 设置以下参数：

- 设置为 192.168.85.21 的地址
- 设置为缺省值的网络掩码和广播地址
- 设置的 -failover 和 deprecated 选项

注 – 必须将 IPv4 测试地址标记为 deprecated，才能防止应用程序使用该测试地址。

---

4 检查特定接口的 IPv4 配置。

通过键入 ifconfig *interface*，始终可以查看接口的当前状态。有关查看接口状态的更多信息，请参阅第 178 页中的 “如何获取有关特定接口的信息”。

通过指定分配给测试地址的逻辑接口，可以获取有关物理接口的测试地址配置的信息。

```
# ifconfig hme0:1
    hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
    mtu 1500 index 2
    inet 192.168.85.21 netmask ffffff00 broadcast 192.168.85.255
```

5 （可选）如果适用，请配置 IPv6 测试地址。

```
# ifconfig interface inet6 -failover
```

具有 IPv6 地址的物理接口会放入与接口的 IPv4 地址相同的 IPMP 组中。如果将具有 IPv4 地址的物理接口配置到 IPMP 组中，则会出现此情况。如果首先将具有 IPv6 地址的物理接口放入 IPMP 组中，则具有 IPv4 地址的物理接口也会隐式放入同一 IPMP 组中。

例如，要使用 IPv6 测试地址配置 `hme0`，可键入以下内容：

```
# ifconfig hme0 inet6 -failover
```

无需将 IPv6 测试地址标记为 deprecated 即可防止应用程序使用该测试地址。

6 检查 IPv6 配置。

```
# ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:17fa/10
            groupname test
```

IPv6 测试地址是指接口的链路本地地址。

7 （可选）重新引导后保留 IPMP 组配置。

- 对于 IPv4，请将以下行添加到 `/etc/hostname.`*interface* 文件：

  interface-address `<parameters> group` *group-name* `up \`
  `addif` *logical-interface* `-failover deprecated <parameters> up`

  在此情况下，测试 IPv4 地址仅在下次重新引导时配置。如果要在当前会话中调用配置，请执行步骤 1、2 和 3（可选）。

- 对于 IPv6，请将以下行添加到 `/etc/hostname6.`*interface* 文件：

  `-failover group` *group-name* `up`

  此测试 IPv6 地址仅在下次重新引导时配置。如果要在当前会话中调用配置，请执行步骤 1、2 和 5（可选）。

8 （可选）通过重复步骤 1 至 6，将更多接口添加到 IPMP 组中。

可以将新接口添加到实时系统上的现有组中。但是，重新引导后更改会丢失。

示例 28–1 配置具有两个接口的 IPMP 组

假定您要执行以下操作：

- 将网络掩码和广播地址设置为缺省值。
- 使用测试地址 `192.168.85.21` 配置接口。

可键入以下命令：

```
# ifconfig hme0 addif 192.168.85.21 netmask + broadcast + -failover deprecated up
```

必须将 IPv4 测试地址标记为 `deprecated`，才能防止应用程序使用该测试地址。请参见第 626 页中的 “如何配置具有多个接口的 IPMP 组”。

要启用地址的故障转移属性，可使用不带连字符的 failover 选项。

IPMP 组中的所有测试 IP 地址都必须使用相同的网络前缀。测试 IP 地址必须属于单个 IP 子网。

示例 28–2 重新引导后保留 IPv4 IPMP 组配置

假定您要创建一个名为 testgroup1 的具有以下配置的 IPMP 组：

- 数据地址为 192.168.85.19 的物理接口 hme0
- 测试地址为 192.168.85.21 的逻辑接口

注 – 在本示例中，将物理接口和数据地址联系在了一起。逻辑接口和测试地址也是如此。但是，接口“类型”和地址类型之间并没有内在的关系。

- 设置的 deprecated 和 -failover 选项
- 设置为缺省值的网络掩码和广播地址

可将以下行添加到 /etc/hostname.hme0 文件：

```
192.168.85.19 netmask + broadcast + group testgroup1 up \
    addif 192.168.85.21 deprecated -failover netmask + broadcast + up
```

同样，要将第二个接口 hme1 放入同一组 testgroup1 中并且配置测试地址，可添加以下行：

```
192.168.85.20 netmask + broadcast + group testgroup1 up \
    addif 192.168.85.22 deprecated -failover netmask + broadcast + up
```

示例 28–3 重新引导后保留 IPv6 IPMP 组配置

要为具有 IPv6 地址的接口 hme0 创建测试组，可将以下行添加到 /etc/hostname6.hme0 文件：

```
-failover group testgroup1 up
```

同样，要将第二个接口 hme1 放入组 testgroup1 中并且配置测试地址，可将以下行添加到 /etc/hostname6.hme1 文件：

```
-failover group testgroup1 up
```

故障排除 在 IPMP 组的配置过程中，in.mpathd 向系统控制台或 syslog 文件输出大量消息。这些消息实质是提示性消息，表示 IPMP 配置工作正常。

- 此消息表示已将接口 hme0 添加到 IPMP 组 testgroup1。但是，hme0 未配置测试地址。要启用基于探测器的故障检测，需要为接口指定测试地址。

```
May 24 14:09:57 host1 in.mpathd[101180]:
No test address configured on interface hme0;
disabling probe-based failure detection on it.
testgroup1
```

- 对于已添加到 IPMP 组中的仅具有 IPv4 地址的所有接口，都会显示此消息。

```
May 24 14:10:42 host4 in.mpathd[101180]:
NIC qfe0 of group testgroup1 is not
plumbed for IPv6 and may affect failover capability
```

- 为接口配置测试地址后，应该显示此消息。

```
Created new logical interface hme0:1
May 24 14:16:53 host1 in.mpathd[101180]:
Test address now configured on interface hme0;
 enabling probe-based failure detection on it
```

另请参见 如果希望 IPMP 组具有活动-待机配置，请转到第 632 页中的 “如何为 IPMP 组配置待机接口”。

配置目标系统

基于探测器的故障检测涉及目标系统的使用，如第 617 页中的 “基于探测器的故障检测”中所述。对于某些 IPMP 组，`in.mpathd` 会使用大量的缺省目标。但是，对于另一些 IPMP 组，则可能需要为基于探测器的故障检测配置特定目标。通过将路由表中的主机路由设置为探测目标，可以完成基于探测器的故障检测。缺省路由器的前面将列出在路由表中配置的任何主机路由。因此，IPMP 会使用明确定义的主机路由来选择目标。可以使用以下两种方法中的任一种直接指定目标：手动设置主机路由或创建可以成为启动脚本的 shell 脚本。

在评定网络中的哪些主机可能成为合适的目标时，请考虑以下标准。

- 确保将来的目标可用并且正在运行。建立其 IP 地址的列表。
- 确保目标接口与要配置的 IPMP 组位于同一网络中。
- 目标系统的网络掩码和广播地址必须与 IPMP 组中的地址相同。
- 目标主机必须能够应答使用基于探测器的故障检测的接口发出的 ICMP 请求。

▼ 如何为基于探测器的故障检测手动指定目标系统

1 使用您的用户帐户登录到要在其中配置基于探测器的故障检测的系统。

2 将路由添加到要用作基于探测器的故障检测中的目标的特定主机。

```
$ route add -host destination-IP gateway-IP -static
```

将 *destination-IP* 和 *gateway-IP* 的值替换为要用作目标的主机的 IPv4 地址。例如，可以键入以下内容以指定目标系统 `192.168.85.137`，该目标系统与 IPMP 组 `testgroup1` 中的接口位于同一子网中。

```
$ route add -host 192.168.85.137 192.168.85.137 -static
```

3 将路由添加到网络中要用作目标系统的其他主机。

▼ 如何在 shell 脚本中指定目标系统

1 在已配置 IPMP 组的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 创建一个将静态路由设置为建议目标的 shell 脚本。

例如，可以创建一个名为 ipmp.targets 的包含以下内容的 shell 脚本：

```
TARGETS="192.168.85.117 192.168.85.127 192.168.85.137"

case "$1" in
        'start')
            /usr/bin/echo "Adding static routes for use as IPMP targets"
          for target in $TARGETS; do
      /usr/sbin/route add -host $target $target
          done
                  ;;
        'stop')
              /usr/bin/echo "Removing static routes for use as IPMP targets"
         for target in $TARGETS; do
        /usr/sbin/route delete -host $target $target
         done
                  ;;
  esac
```

3 将 shell 脚本复制到启动脚本目录中。

```
# cp ipmp.targets /etc/init.d
```

4 更改新启动脚本的权限。

```
# chmod 744 /etc/init.d/ipmp.targets
```

5 更改新启动脚本的所有权。

```
# chown root:sys /etc/init.d/ipmp.targets
```

6 在 /etc/init.d 目录中为启动脚本创建链接。

```
# ln /etc/init.d/ipmp.targets /etc/rc2.d/S70ipmp.targets
```

文件名 S70ipmp.targets 中的 S70 前缀会将新脚本相对其他启动脚本正确进行排序。

配置待机接口

如果希望 IPMP 组具有活动-待机配置，请使用此过程。有关此类型配置的更多信息，请参阅第 615 页中的 “IPMP 接口配置”。

▼ 如何为 IPMP 组配置待机接口

开始之前
- 必须已将所有接口配置为 IPMP 组的成员。
- 在要成为待机接口的接口上，不应配置测试地址。

有关配置 IPMP 组和指定测试地址的信息，请参阅第 626 页中的 “如何配置具有多个接口的 IPMP 组”。

1 在要配置待机接口的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 将一个接口配置为待机接口并指定测试地址。

```
# ifconfig interface plumb \
ip-address other-parameters deprecated -failover standby up
```

待机接口只能具有一个 IP 地址，即测试地址。设置 `standby up` 选项之前，必须先设置 `-failover` 选项。对于 `<other-parameters>`，请使用您的配置所需的参数，如 `ifconfig(1M)` 手册页中所述。

- 例如，要创建 IPv4 测试地址，可键入以下命令：

```
# ifconfig hme1 plumb 192.168.85.22 netmask + broadcast + deprecated -failover standby up
```

`hme1`	将 `hme1` 定义为要配置为待机接口的物理接口。
`192.168.85.22`	将此测试地址指定给待机接口。
`deprecated`	表示测试地址不用于外发包。
`-failover`	表示在接口出现故障时测试地址不进行故障转移。
`standby`	将接口标记为待机接口。

- 例如，要创建 IPv6 测试地址，可键入以下命令：

```
# ifconfig hme1 plumb -failover standby up
```

3 检查待机接口配置的结果。

```
# ifconfig hme1
hme1: flags=69040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,
      STANDBY,INACTIVE mtu 1500
         index 4 inet 192.168.85.22 netmask ffffff00 broadcast 19.16.85.255
         groupname test
```

`INACTIVE` 标志表示此接口不用于任何外发包。此待机接口上发生故障转移时，会清除 `INACTIVE` 标志。

注 – 通过键入 `ifconfig` *interface* 命令，始终可以查看接口的当前状态。有关查看接口状态的更多信息，请参阅第 178 页中的 “如何获取有关特定接口的信息”。

4 **（可选）重新引导后保留 IPv4 待机接口。**

将待机接口指定给同一 IPMP 组，并为该待机接口配置测试地址。

例如，要将 `hme1` 配置为待机接口，可将以下行添加到 `/etc/hostname.hme1` 文件：

```
192.168.85.22 netmask + broadcast + deprecated group test -failover standby up
```

5 **（可选）重新引导后保留 IPv6 待机接口。**

将待机接口指定给同一 IPMP 组，并为该待机接口配置测试地址。

例如，要将 `hme1` 配置为待机接口，可将以下行添加到 `/etc/hostname6.hme1` 文件：

```
-failover group test standby up
```

示例 28–4 为 IPMP 组配置待机接口

假定您要创建具有以下配置的测试地址：

- 作为待机接口的物理接口 `hme2`
- 测试地址为 `192.168.85.22`
- 设置的 `deprecated` 和 `-failover` 选项
- 设置为缺省值的网络掩码和广播地址

应键入以下内容：

```
# ifconfig hme2 plumb 192.168.85.22 netmask + broadcast + \
deprecated -failover standby up
```

仅当地址标记为 `NOFAILOVER` 地址时，才会将接口标记为待机接口。

通过键入以下内容，可以删除接口的待机状态：

```
# ifconfig interface -standby
```

配置具有单个物理接口的 IPMP 组

如果 IPMP 组中仅有一个接口，则无法进行故障转移。但是，可以通过将接口指定给 IPMP 组对该接口启用故障检测。不必配置专用的测试 IP 地址，即可为单接口 IPMP 组建立故障检测。可以使用单个 IP 地址发送数据和检测故障。

▼ 如何配置单接口 IPMP 组

1 在将来具有单接口 IPMP 组的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 对于 IPv4，创建单接口 IPMP 组。

使用以下语法将单个接口指定给 IPMP 组。

 # ifconfig *interface* **group** *group-name*

以下示例会将接口 `hme0` 指定给 IPMP 组 `v4test`：

```
# ifconfig hme0 group v4test
```

执行此步骤之后，IPMP 会对接口启用基于链路的故障检测。

此外，还可以使用 `ifconfig` 命令的 `-failover` 子命令启用基于探测器的故障检测。以下示例通过使用当前分配给 `hme0` 的 IP 地址，在 `hme0` 上启用基于探测器的故障检测。

```
# ifconfig hme0 -failover
```

请注意，与多接口组不同，同一 IP 地址既可充当数据地址，又可充当测试地址。要让应用程序将测试地址用作数据地址，不能针对单接口 IPMP 组将测试地址标记为 `deprecated`。

3 对于 IPv6，创建单接口 IPMP 组。

使用以下语法将单个接口指定给 IPMP 组：

 # ifconfig *interface* **inet6 group** *group-name*

例如，要将单接口 `hme0` 添加到 IPMP 组 `v6test` 中，请键入以下内容：

```
# ifconfig hme0 inet6 group v6test
```

执行此步骤之后，IPMP 会对接口启用基于链路的故障检测。

此外，还可以使用 `ifconfig` 命令的 `-failover` 子命令启用基于探测器的故障检测。以下示例通过使用当前分配给 `hme0` 的 IP 地址，在 `hme0` 上启用基于探测器的故障检测。

```
# ifconfig hme0 inet6 -failover
```

请注意，与多接口组不同，同一 IP 地址既可充当数据地址，又可充当测试地址。要让应用程序将测试地址用作数据地址，不能针对单接口 IPMP 组将测试地址标记为 `deprecated`。

在单物理接口配置中，无法检验所探测的目标系统是否已出现故障或者接口是否已出现故障。仅可以通过一个物理接口来探测目标系统。如果子网中仅有一个缺省路由

器，则当组中仅有一个物理接口时，关闭 IPMP。如果存在单独的 IPv4 和 IPv6 缺省路由器，或者存在多个缺省路由器，则需要探测多个目标系统。因此，可以安全地打开 IPMP。

维护 IPMP 组

本节包含维护现有 IPMP 组以及组成这些组的接口的任务。这些任务假定已按第 624 页中的 “使用 IPMP 组获得高可用性”中的说明配置了 IPMP 组。

▼ 如何显示接口的 IPMP 组成员关系

1 **在具有 IPMP 组配置的系统上，成为超级用户或承担等效角色。**

角色包含授权和具有特权的命令。有关角色的更多信息，请参见《System Administration Guide: Security Services》中的“Configuring RBAC (Task Map)”。

2 **显示有关接口的信息，包括接口所属的组。**

```
# ifconfig interface
```

3 **（如果适用）显示接口的 IPv6 信息。**

```
# ifconfig interface inet6
```

示例 28–5 显示物理接口组

要显示 hme0 的组名，可键入以下内容：

```
# ifconfig hme0
    hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
      index 2 inet 192.168.85.19 netmask ffffff00 broadcast 192.168.85.255
      groupname testgroup1
```

要显示仅用于 IPv6 信息的组名，可键入以下内容：

```
# ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:19fa/10
            groupname testgroup1
```

▼ 如何将接口添加到 IPMP 组

1 **在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **将接口添加到 IPMP 组。**

```
# ifconfig interface group group-name
```

在 *interface* 中指定的接口会成为 IPMP 组 *group-name* 的成员。

示例 28–6 将接口添加到 IPMP 组

要将 `hme0` 添加到 IPMP 组 `testgroup2`，可键入以下命令：

```
# ifconfig hme0 group testgroup2
  hme0: flags=9000843<UP ,BROADCAST,RUNNING,MULTICAST,IPv4,NOFAILOVER> mtu 1500 index 2
  inet 192.168.85.19 netmask ff000000 broadcast 10.255.255.255
  groupname testgroup2
  ether 8:0:20:c1:8b:c3
```

▼ 如何从 IPMP 组中删除接口

执行 `ifconfig` 命令的包含空字符串的 `group` 参数时，将从接口的当前 IPMP 组中删除该接口。从组中删除接口时请务必谨慎。如果 IPMP 组中的其他某个接口出现故障，则故障转移可能会提早发生。例如，如果 `hme0` 以前出现过故障，则当 `hme1` 位于同一组中时，所有地址都将故障转移到 `hme1`。从组中删除 `hme1` 会导致 `in.mpathd` 守护进程将所有故障转移地址返回到组中的其他某个接口。如果组中的其他接口都未运行，则故障转移可能无法恢复所有网络访问。

同样，需要取消检测组中的某个接口时，应首先从组中删除该接口。然后，确保该接口已配置了所有的初始 IP 地址。`in.mpathd` 守护进程会尝试恢复从组中删除的接口的初始配置。您需要确保在取消检测接口之前恢复配置。请参阅第 618 页中的 “接口故障转移期间发生的情况”以了解接口在故障转移前后的外观。

1 **在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **从 IPMP 组中删除接口。**

```
# ifconfig interface group ""
```

引号表示空字符串。

示例 28–7 从组中删除接口

要从 IPMP 组 `test` 中删除 `hme0`，可键入以下命令：

```
# ifconfig hme0 group ""
    # ifconfig hme0
    hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
    index 2 inet 192.168.85.19 netmask ffffff00 broadcast 192.168.85.255
    # ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
    inet6 fe80::a00:20ff:feb9:19fa/10
```

▼ 如何将接口从一个 IPMP 组移动到另一个组

如果某个接口属于现有的 IPMP 组，则可以将该接口放入新的 IPMP 组中。无需从当前 IPMP 组中删除该接口。接口放入新组中后，该接口将自动从任何现有的 IPMP 组中删除。

1. **在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **将接口移动到新的 IPMP 组。**

   `# ifconfig` *interface* `group` *group-name*

   如果将接口放入新组中，则会自动从任何现有组中删除该接口。

示例 28–8 将接口移动到其他 IPMP 组

要更改接口 `hme0` 的 IPMP 组，应键入以下内容：

```
# ifconfig hme0 group cs-link
```

此命令会从 IPMP 组 `test` 中删除 `hme0` 接口，然后将其放入组 `cs-link` 中。

在支持动态重新配置的系统上替换出现故障的物理接口

本节包含与管理支持动态重新配置 (Dynamic Reconfiguration, DR) 的系统有关的过程。

注 – 这些任务仅与使用 `ifconfig` 命令配置的 IP 层有关。如果 IP 层前后的层（如 ATM 或其他服务）未自动化，则它们需要使用特定的手动步骤。以下过程中的步骤用于在预分离过程中取消配置接口以及在连接之后配置接口。

▼ 如何删除出现故障的物理接口（DR 分离）

此过程说明如何在支持 DR 的系统上删除物理接口。该过程假设已具备以下条件：

- 物理接口 `hme0` 和 `hme1` 是示例接口。
- 这两个接口都位于同一 IPMP 组中。
- `hme0` 已出现故障。
- 逻辑接口 `hme0:1` 具有测试地址。
- 出现故障的接口将替换为相同的物理接口名称，例如，`hme0` 替换为 `hme0`。

注 – 如果测试地址是使用 `/etc/hostname.hme0` 文件检测的，则可以跳过步骤 2。

1 在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 显示测试地址配置。

```
# ifconfig hme0:1

hme0:1:
flags=9040842<BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
inet 192.168.233.250 netmask ffffff00 broadcast 192.168.233.255
```

替换物理接口时，需要使用此信息重新检测测试地址。

3 删除物理接口。

有关如何删除物理接口的完整说明，请参阅以下资料：

- `cfgadm(1M)` 手册页
- 《Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide》
- 《Sun Enterprise 10000 DR 配置指南》

▼ 如何替换出现故障的物理接口（DR 连接）

此过程说明如何在支持 DR 的系统上替换物理接口。

1 **在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **替换物理接口。**

请参阅以下资料中的说明：

- `cfgadm(1M)` 手册页
- 《Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide》
- 《Sun Enterprise 10000 DR 配置指南》或《Sun Fire 880 Dynamic Reconfiguration User's Guide》

恢复系统引导时不存在的物理接口

注 – 以下过程仅与使用 `ifconfig` 命令配置的 IP 层有关。如果 IP 层前后的层（如 ATM 或其他服务）未自动化，则它们需要使用特定的手动步骤。以下过程中的特定步骤用于在预分离过程中取消配置接口以及在连接之后配置接口。

对于位于 Sun Fire™ 平台的 I/O 板上的接口，动态重新配置后的恢复是自动进行的。如果 NIC 是 Sun Crypto 加速器 I - cPCI 板，则恢复也是自动进行的。因此，对于在 DR 操作过程中恢复的接口，不需要执行以下步骤。有关 Sun Fire x800 和 Sun Fire 15000 系统的更多信息，请参见 `cfgadm_sbd(1M)` 手册页。物理接口将故障恢复到在 `/etc/hostname.`*interface* 文件中指定的配置。有关如何将接口配置为在重新引导后保留配置的详细信息，请参见第 624 页中的 “使用 IPMP 组获得高可用性”。

注 – 在 Sun Fire 传统 (Exx00) 系统上，DR 分离仍须手动进行。但是，DR 连接是自动进行的。

▼ 如何恢复系统引导时不存在的物理接口

恢复系统引导时不存在的物理接口之前，必须完成以下过程。此过程中的示例具有以下配置：

- 物理接口 `hme0` 和 `hme1` 是示例所用的接口。
- 这两个接口都位于同一 IPMP 组中。
- 系统引导时未安装 `hme0`。

注 – 在恢复出现故障的物理接口的过程中，IP 地址的故障恢复需要长达三分钟。此时间可能会随网络通信流量的不同而不同。此时间还取决于通过 `in.mpathd` 守护进程对故障转移的接口进行故障恢复的传入接口的稳定性。

---

1 在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 从控制台日志的故障错误消息中检索故障网络的信息。

请参见 `syslog(3C)` 手册页。错误消息可能与以下内容类似：

```
moving addresses from failed IPv4 interfaces:
hme1 (moved to hme0)
```

此消息表示故障接口 `hme1` 上的 IPv4 地址已故障转移到 `hme0` 接口。

或者，可能收到以下类似消息：

```
moving addresses from failed IPv4 interfaces:
hme1 (couldn't move, no alternative interface)
```

此消息表示在故障接口 `hme1` 所在的组中无法找到活动接口。因此，`hme1` 上的 IPv4 地址无法故障转移。

3 将物理接口连接到系统。

有关如何替换物理接口的说明，请参阅以下内容：

- `cfgadm(1M)` 手册页
- 《Sun Enterprise 10000 DR 配置指南》
- 《Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide》

4 请参阅步骤 2 中的消息内容。如果无法移动地址，请转到步骤 6。如果地址已移动，请继续执行步骤 5。

5 **取消检测在故障转移过程中配置的逻辑接口。**

a. **查看 `/etc/hostname.`*moved-from-interface* 文件的内容，确定哪些逻辑接口已在故障转移过程中配置。**

b. **取消检测每个故障转移 IP 地址。**

```
# ifconfig moved-to-interface removeif moved-ip-address
```

注 – 故障转移地址标记有 `failover` 参数，或者未标记有 `-failover` 参数。无需对标记有 `-failover` 的 IP 地址取消检测。

例如，假定 `/etc/hostname.hme0` 文件的内容包含以下行：

```
inet 10.0.0.4 -failover up group one
addif 10.0.0.5 failover up
addif 10.0.0.6 failover up
```

要取消检测每个故障转移 IP 地址，可键入以下命令：

```
# ifconfig hme0 removeif 10.0.0.5
# ifconfig hme0 removeif 10.0.0.6
```

6 **通过对已删除的每个接口键入以下命令，为已替换的物理接口重新配置 IPv4 信息：**

```
# ifconfig removed-from-NIC <parameters>
```

例如，可键入以下命令：

```
# ifconfig hme1 inet plumb
# ifconfig hme1 inet 10.0.0.4 -failover up group one
# ifconfig hme1 addif 10.0.0.5 failover up
# ifconfig hme1 addif 10.0.0.6 failover up
```

修改 IPMP 配置

使用 IPMP 配置文件 `/etc/default/mpathd` 为 IPMP 组配置以下系统范围的参数。

- `FAILURE_DETECTION_TIME`
- `TRACK_INTERFACES_ONLY_WITH_GROUPS`
- `FAILBACK`

▼ 如何配置 /etc/default/mpathd 文件

1 **在具有 IPMP 组配置的系统上，承担主管理员角色或成为超级用户。**

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2 **编辑 /etc/default/mpathd 文件。**

更改这三个参数中的一个或多个参数的缺省值。

a. **为 FAILURE_DETECTION_TIME 参数键入新值。**

```
FAILURE_DETECTION_TIME=n
```

其中 *n* 是 ICMP 探测器用来检测是否发生接口故障的时间（以秒为单位）。缺省值是 10 秒。

b. **为 FAILBACK 参数键入新值。**

```
FAILBACK=[yes | no]
```

- *yes* – *yes* 值是 IPMP 的缺省故障恢复行为。当检测到故障接口修复时，网络访问故障恢复到已修复的接口，如第 616 页中的 “IPMP 故障检测和恢复功能”中所述。
- *no* – *no* 指示数据通信不会移回已修复的接口。当检测到某个故障接口已修复时，会为此接口设置 *INACTIVE* 标志。此标志表示该接口当前不用于数据通信，但该接口仍可用于探测器通信。

  例如，假设 IPMP 组由两个接口 ce0 和 ce1 组成。然后假定在 */etc/default/mpathd* 中设置了值 `FAILBACK=no`。如果 ce0 出现故障，它的通信将会故障转移至 ce1，这是 IPMP 的预期行为。但是，当 IPMP 检测到 ce0 已修复时，通信不会从 ce1 进行故障恢复，这是因为在 */etc/default/mpathd* 中设置了参数 `FAILBACK=no`。ce0 接口保持 `INACTIVE` 状态并且不用于通信，除非 ce1 接口出现故障。如果 ce1 接口出现故障， ce1 上的地址会迁移回 ce0，其 INACTIVE 标志也随之被清除。发生此迁移的前提条件是 ce0 是该组中唯一的 INACTIVE 接口。如果该组中存在其他的 INACTIVE 接口，这些地址可能会迁移到 ce0 以外的 INACTIVE 接口。

c. **为 TRACK_INTERFACES_ONLY_WITH_GROUPS 参数键入新值。**

```
TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
```

- *yes* – *yes* 值是 IPMP 的缺省行为。此参数使 IPMP 忽略未配置到 IPMP 组中的网络接口。
- *no* – *no* 值为**所有**网络接口设置故障和修复检测，无论它们是否配置到 IPMP 组中。但是，如果在未配置到 IPMP 组中的接口上检测到故障或修复，不会发生故障转移或故障恢复。因此，*no* 值仅用于报告故障，并不能直接提高网络可用性。

3 重新启动 **in.mpathd** 守护进程。

```
# pkill -HUP in.mpathd
```

第 6 部 分

IP 服务质量 (IP Quality of Service, IPQoS)

本部分介绍有关 IP 服务质量 (IP Quality of Service, IPQoS)、Oracle Solaris 操作系统实现区分服务的任务和信息。

◆ ◆ ◆ 第 29 章

29 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：

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 数据包头的能力
- 收集通信流统计信息的流记帐模块
- 通过 UNIX® `kstat` 命令针对通信类进行统计信息收集
- 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》（《IPv4 和 IPv6 包头中区分服务字段（DS 字段）的定义》）(`http://www.ietf.org/rfc/rfc2474.txt?number=2474`)－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》（《保证转发 PHB 组》）(http://www.ietf.org/rfc/rfc2597.txt?number=2597)－说明保证转发 (Assured Forwarding, AF) 单跳行为的工作原理。
- RFC 2598，《An Expedited Forwarding PHB》（《速转发 PHB》）(http://www.ietf.org/rfc/rfc2598.txt?number=2598)－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 665 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，提供商可以根据变化的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。您可以为该类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 715 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` 分类器将通信流整理到基于在 IPQoS 配置文件中配置的特征的各个类中。

有关 ipgpc 的详细信息，请参阅第 715 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 667 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 669 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 670 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* 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 672 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

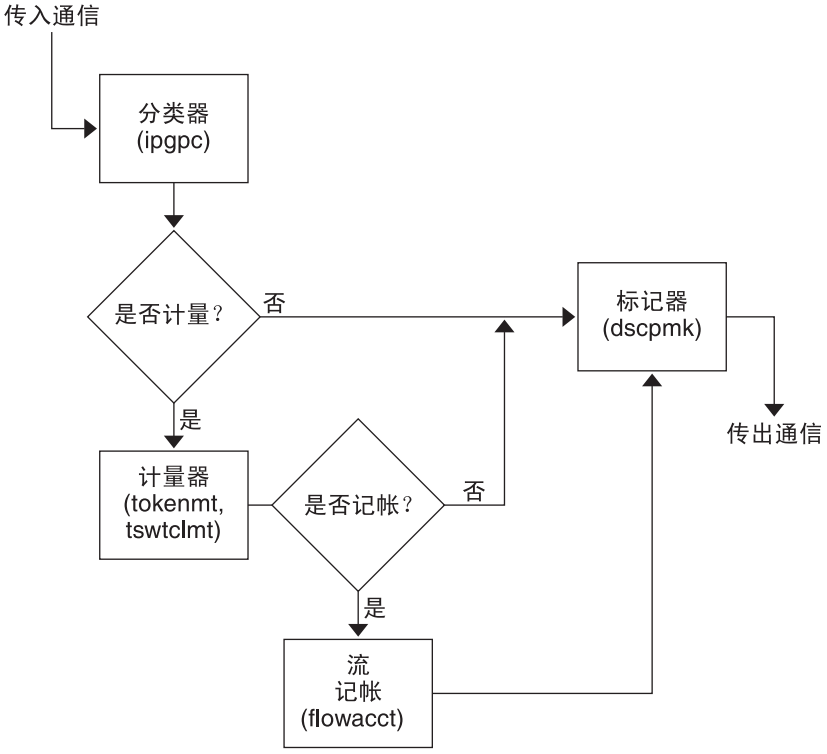
您还可以收集有关其他属性的统计信息，如第 711 页中的 “记录有关通信流量的信息”，以及 `flowacct(7ipp)` 和 `acctadm(1M)` 手册页中所述。

有关规划流记帐策略的信息，请参见第 674 页中的“如何规划流记帐”。

通信如何流过 IPQoS 模块

下图显示传入通信可能用以经过某些 IPQoS 模块的路径。

图 29–1 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元素。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。相反，`dlcosmk` 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 720 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 720 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

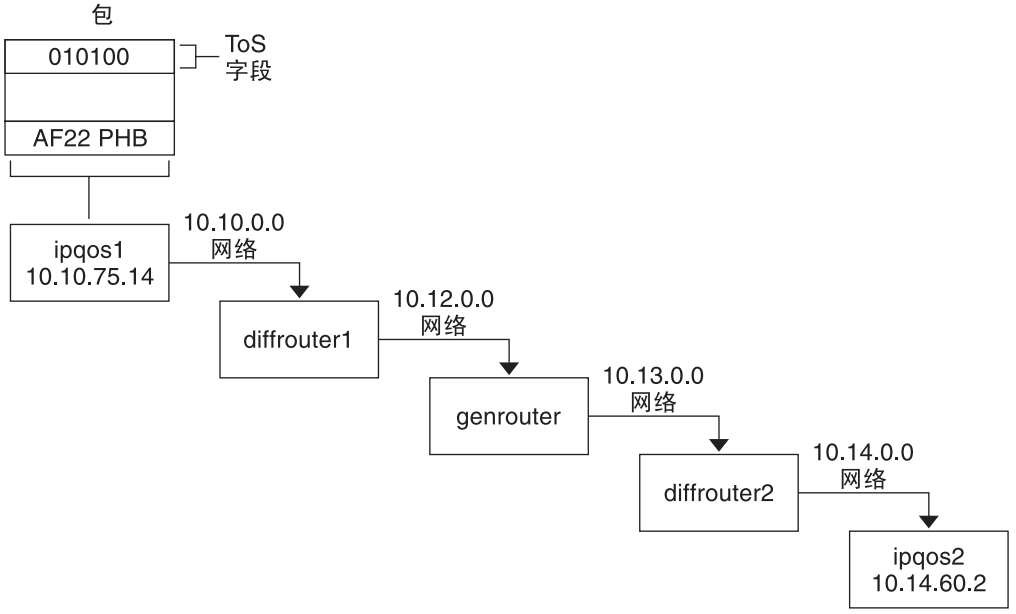
保证转发 *(Assured Forwarding, AF)*单跳行为提供了四种不同的转发类，您可以将这些类指定给包。每种转发类都提供三种丢弃优先级，如表 34–2 中所示。

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 29–2 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。

◆ ◆ ◆ 第 30 章

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：

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 662 页中的 “规划 Diffserv 网络拓扑”。
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 665 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 665 页中的 “规划服务质量策略”。

任务	说明	参考
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，客户会对服务器和服务器整合（例如 Oracle 提供的 Sun Enterprise(TM) 服务器）运行 IPQoS。与此相反，您还可以根据网络需求在桌面系统（例如 UltraSPARC® 系统）上运行 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 的负载平衡器管理的、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。

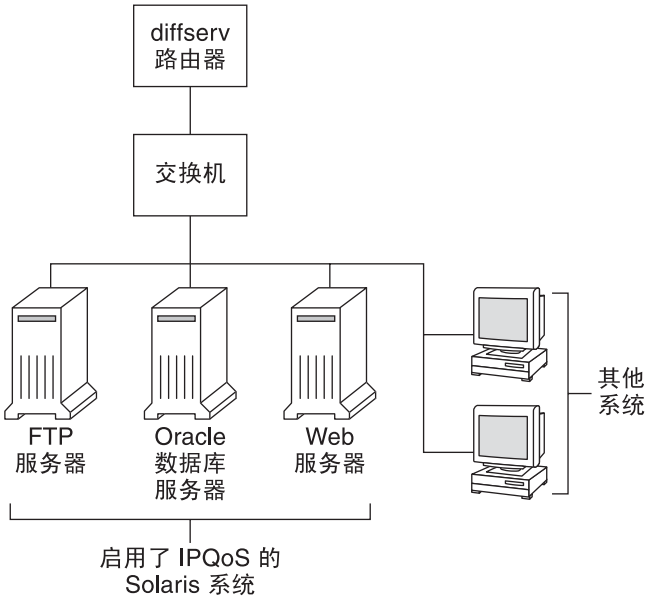
IPQoS 网络拓扑

本节介绍可满足各种网络需求的 IPQoS 策略。

单个主机上的 IPQoS

下图显示启用了 IPQoS 的系统的单个网络。

图 30–1 网络段上的 IPQoS 系统



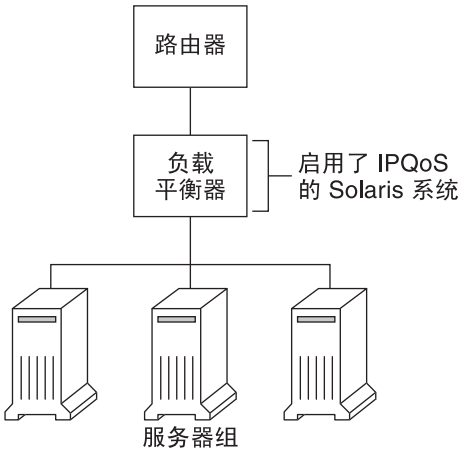
此网络仅为公司内联网的一段。通过在应用服务器和 Web 服务器上启用 IPQoS，您可以控制每个 IPQoS 系统释放传出通信的速率。如果使路由器可识别 Diffserv，则还可以进一步控制传入和传出的通信。

本指南中的示例使用“单个主机上的 IPQoS”方案。有关整个指南中使用的拓扑示例，请参见图 30–4。

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 30–2 启用了 IPQoS 的服务器场的网络



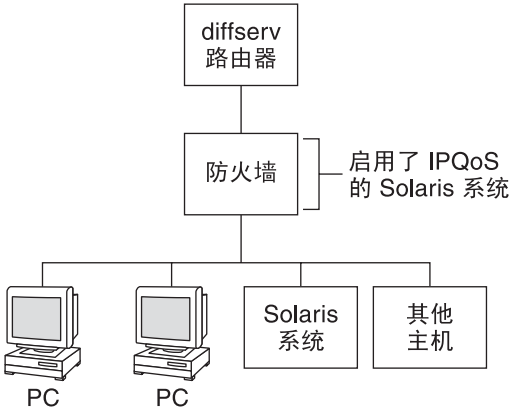
在此类拓扑中，路由器可识别 Diffserv，因此可以对传入和传出的通信进行排队和计速。负载平衡器也可识别 Diffserv，并且服务器场也启用了 IPQoS。负载平衡器可以使用选定器（例如用户 ID 和项目 ID）来提供路由器之外的其他过滤功能。这些选定器包括在应用程序数据中。

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

下图显示了某公司网络的一段，由其他段通过防火墙保障该段的安全。

图 30–3 由启用了 IPQoS 的防火墙保护的网络



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查看、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 30–1 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，以及指向执行各项任务的指南的链接。

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 666 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 667 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 669 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 670 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 672 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 674 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 662 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 653 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 666 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 30–1。

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的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 的系统的问题通信创建类。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别。**

   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 669 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 672 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便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 653 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 34–1。

表 30–2 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 /etc/services 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 /etc/protocols 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	为类指定的优先级别。有关更多信息，请参见第 667 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 667 页中的 “如何定义 QoS 策略类”过程。

1. **至少为您在第 667 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。**

   使用表 30–1 中介绍的 QoS 规划表为您定义的类填充过滤器。

示例 30–1 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 670 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 672 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 674 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 667 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 669 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 653 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 669 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 667 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 718 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 719 页中的 “tswtclmt 计量模块”和 tswtclmt(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 717 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 30–2 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 672 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 674 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 667 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 669 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 670 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 683 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 722 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 656 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 720 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 30–3 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 34–2 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 674 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 667 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 669 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 670 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 672 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 683 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 667 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 669 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 670 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 672 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 674 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 683 页中的 “如何创建 IPQoS 配置文件并定义通信类”。

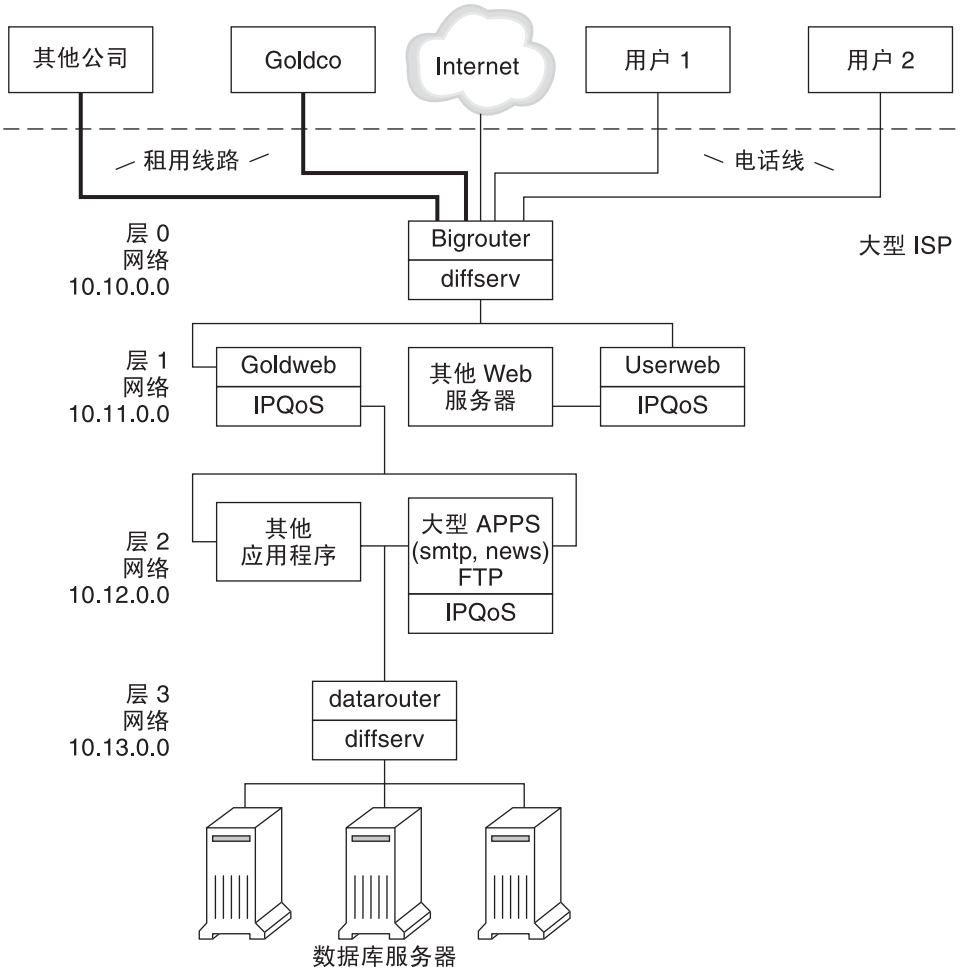
IPQoS 配置示例介绍

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 30–4 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 处购买的作为高级服务一部分的 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。

◆ ◆ ◆ 第 31 章

31 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：

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 665 页中的 “规划服务质量策略”。

在 IPQoS 配置文件中定义 QoS 策略（任务列表）

此任务列表列出用于创建 IPQoS 配置文件的常规任务，以及指向说明执行这些任务的步骤的章节的链接。

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 666 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 665 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 683 页中的 “如何创建 IPQoS 配置文件并定义通信类”
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 685 页中的 “如何在 IPQoS 配置文件中定义过滤器”

任务	说明	参考
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 690 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 698 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 686 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 689 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 704 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 701 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 704 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 665 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 34–3 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 30–4 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 30–4 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 31–1 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
    }
}
action {
    module dscpmk
```

示例 31–1 高级 Web 服务器的 IPQoS 配置文件样例 （续）

```
    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 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 31–2 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
   }
}
action {
```

示例 31–2 尽力服务 Web 服务器的配置样例 （续）

```
    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。本章中的任务使用目录 `/var/ipqos` 作为 IPQoS 配置文件的位置。下面的过程将构建示例 31–1 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 31–1。

---

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 727 页中的 “`action` 语句” 和 `ipqosconf(1M)` 手册页。

3 **添加带有统计信息参数 global_stats 的 params 子句。**

```
params {
        global_stats TRUE
  }
```

在 ipgpc.classify 语句中使用参数 global_stats TRUE 可收集此操作的统计信息。当在类子句定义中指定了 enable_stats TRUE 时，global_stats TRUE 还允许按类收集统计信息。

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 的参数更改为 FALSE 来关闭统计信息收集。

但是，全局统计信息是可以在 params 子句中定义的一种参数类型。有关 params 子句的语法信息和其他详细信息，请参阅第 729 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。

name goldweb	创建类 goldweb 以标识要送达 Goldweb 服务器的通信。
next_action markAF11	指示 ipgpc 模块将 goldweb 类的包传递到 markAF11 操作语句。markAF11 操作语句将调用 dscpmk 标记器。
enable_stats FALSE	用于提取 goldweb 类的统计信息。但是，由于 enable_stats 的值为 FALSE，因此不会打开此类的统计信息。

有关 class 子句语法的详细信息，请参见第 728 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
   }
```

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

收集。

另请参见
- 要为您刚创建的类定义过滤器，请参阅第 685 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 683 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 683 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，必须十分谨慎地使用花括号 ({}) 括住每条 `class` 子句和每条 `filter` 子句。有关花括号的用法示例，请使用示例 31–1。

---

1 打开 IPQoS 配置文件，并定位到已定义的最后一个类的结尾。

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 728 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 686 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 698 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 704 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 685 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 694 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 31–1 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 `dlclosmk` 标记器转发通信的信息，请参阅第 722 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 定义要对通信流采取的标记器操作。

```
    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。有关可能的 AF DSCP 表的信息，请参阅表 34–2。

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。

在 DS 字段中，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 720 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 701 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 689 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 686 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 698 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 704 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 685 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 694 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 683 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 31–1 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 开始可调用流记帐的 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 子句以控制对通信类的记帐。

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

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。
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。

`timeout 10000`	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。
`max_limit 2048`	在流表中针对该操作实例设置活动流记录的最大数目。
`next_action continue`	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流的统计信息，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 701 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 704 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 694 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 31–2 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` 文件必须以局部 `action` 语句开头以调用 `ipgpc` 分类器。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 683 页中的 “如何创建 IPQoS 配置文件并定义通信类”。

3 定义类以标识要送达尽力服务 Web 服务器的通信。

```
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`，因此不会针对此类收集统计信息。

有关此 `class` 子句任务的解释，请参见第 683 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 685 页中的 “如何在 IPQoS 配置文件中定义过滤器”。

5 开始 `action` 语句以调用 `dscpmk` 标记器。

```
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 定义用于处理通信流的标记器参数。

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

`global_stats FALSE`	对 `markAF12` 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{0–63:12}`	将 DSCP 12 指定给通信类 `userweb` 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 `userweb` 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 694 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 701 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 704 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 30–4 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 31–3 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 31–3 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 31–3 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 683 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`	创建名为 `smtp` 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 `smtp` 类启用统计信息收集。但是，由于 `enable_stats` 的值为 `FALSE`，因此不会针对此类收集统计信息。
`next_action markAF13`	指示 `ipgpc` 模块在 `ipgpc` 完成处理之后，将 `smtp` 类的包传递到 `markAF13 action` 语句。
`name news`	创建名为 `news` 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 `ipgpc` 模块在 `ipgpc` 完成处理之后，将 `news` 类的包传递到 `markAF21` 操作语句。
`name ftp`	创建名为 `ftp` 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 `ftp` 类启用统计信息收集。
`next_action meterftp`	指示 `ipgpc` 模块在 `ipgpc` 完成处理之后，将 `ftp` 类的包传递到 `meterftp action` 语句。

有关定义类的更多信息，请参阅第 683 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
    }
}
```

`name smtpout`	为过滤器提供名称 `smtpout`。
`sport smtp`	选择源端口 25 传出的通信，此端口是众所周知的用于 `sendmail` (SMTP) 应用程序的端口。
`class smtp`	标识过滤器所属的类，在此实例中为类 `smtp`。
`name newsout`	为过滤器提供名称 `newsout`。
`sport nntp`	选择名称为 `nntp` 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。
`class news`	标识过滤器所属的类，在此实例中为类 `news`。
`name ftpout`	为过滤器提供名称 `ftpout`。
`sport ftp`	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。
`name ftpdata`	为过滤器提供名称 `ftpdata`。
`sport ftp-data`	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。
`class ftp`	标识 `ftpout` 和 `ftpdata` 过滤器所属的类，在此实例中为 `ftp`。

另请参见

- 要定义过滤器，请参阅第 685 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 696 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 698 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 689 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，您将为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 31–3 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。标记器使用 DS 字段中的 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 34–2。

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。标记器使用 DS 字段中的 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 683 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 698 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 689 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 701 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 704 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 配置文件中使用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 31–3 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 718 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 717 页中的 “计量器模块”。

5 添加标记器 `action` 语句以便为属于 `ftp` 类的非一致的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 标记 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 34–2。

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 通信符合已配置的速率时，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 34–2。

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 704 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 683 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 689 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 701 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 662 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 30–4 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，以及将这些代码点应用到的系统和类。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。

◆ ◆ ◆ 第 32 章

32 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：

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 679 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 ipqosconf 命令来激活 IPQoS 配置文件。	第 704 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 704 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 705 页中的 “如何在引导过程中启用 IPQoS 消息的日志记录”。
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 32–1 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Creating IPQoS Configuration Files for Web Servers中创建的文件 第 681 页中的 “为 Web 服务器创建 IPQoS 配置文件” 为例。有关详细信息，请参阅 ipqosconf(1M)。

1. **在启用 IPQoS 的系统上承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

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 – 当您通过 -a 选项应用 IPQoS 配置文件时，文件中的操作只对当前会话处于活动状态。

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 714 页中的 “收集统计信息”。
- 要记录 ipqosconf 消息，请参阅第 705 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 704 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. **在启用 IPQoS 的系统上承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

2. **测试内核模块中是否存在 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 704 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. **在启用 IPQoS 的系统上承担主管理员角色或成为超级用户。**

   Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章 “使用 Solaris Management Console（任务）”。

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 32–1 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 32–1。

对 IPQoS 错误消息进行故障排除

本节介绍由 IPQoS 生成的错误消息的表以及可能的解决方案。

表 32–1 IPQoS 错误消息

错误消息	说明	解决方案
`Undefined action in parameter` *parameter-name's* `action` *action-name*	在 IPQoS 配置文件中，不存在 *parameter-name* 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` *action-name* `involved in cycle`	在 IPQoS 配置文件中，*action-name* 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` *action-name* `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 fmt_version`	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 683 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 version`	IPQoS 不支持在配置文件中指定的格式版本。	将格式版本改为 `fmt_version 1.0`，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 683 页中的 “如何创建 IPQoS 配置文件并定义通信类”中所述。
`Can't commit a null configuration`	运行 `ipqosconf -c` 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 704 页中的 “如何将新配置应用于 IPQoS 内核模块”。

表 32–1 IPQoS 错误消息 （续）

错误消息	说明	解决方案
`Invalid CIDR mask on line` *line-number*	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` *line-number*	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` *line-number*	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 34–5。
`ipgpc action has incorrect name line` *line-number*	在配置文件中指定给 `ipgpc` 操作的名称不是要求的 `ipgpc.classify`。	将操作重命名为 `ipgpc.classify`。
`Second parameter clause not supported line` *line-number*	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` *action-name*	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` *filter-name* `in action` *action-name*	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` *class-name* `action` *action-name*	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` *action-name*	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 715 页中的 “IPQoS 体系结构和 Diffserv 模型”中的模块项。或者，可以参阅 `ipqosconf(1M)`。
`Mandatory parameter missing for action` *action-name*	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 715 页中的 “IPQoS 体系结构和 Diffserv 模型”中的模块项。或者，可以参阅 `ipqosconf(1M)` 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 `ipgpc` 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，通过将项 `ipgpc_max_classes`*class-number* 添加到 `/etc/system` 文件增加类的最大数目。
`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，通过将项 `ipgpc_max_filters`*filter-number* 添加到 `/etc/system` 文件增加过滤器的最大数目。

表 32–1 IPQoS 错误消息 （续）

错误消息	说明	解决方案
`Invalid/missing parameters for filter` *filter-name* `in action ipgpc`	在配置文件中，过滤器 *filter-name* 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` *line-number*	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` *line-number*	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` *line-number* `is invalid`	在配置文件中，行 *line-number* 上的参数的数组声明无效。	有关包含无效数组的 `action` 语句所调用的数组声明的正确语法，请参阅第 715 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,` *line-number*	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` *line-number*	对于参数而言，不支持在配置文件的 *line-number* 上指定的值。	有关 `action` 语句所调用的模块的可接受值的信息，请参阅第 715 页中的 “IPQoS 体系结构和 Diffserv 模型”中的模块说明。或者，可以参阅 `ipqosconf(1M)` 手册页。
`Unrecognized value, line` *line-number*	对于参数而言，配置文件的 *line-number* 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。有关带有无法识别行号的 `action` 语句所调用模块的说明，请参阅第 715 页中的 “IPQoS 体系结构和 Diffserv 模型”。或者，可以参阅 `ipqosconf(1M)` 手册页。
`Malformed value list line` *line-number*	在配置文件的 *line-number* 上指定的枚举不符合规范语法。	有关带有不规则的值列表的 `action` 语句所调用模块的正确语法，请参阅第 715 页中的 “IPQoS 体系结构和 Diffserv 模型”中关于模块的说明。或者，可以参阅 `ipqosconf(1M)` 手册页。
`Duplicate parameter line` *line-number*	在 *line-number* 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` *line-number*	为配置文件的 *line-number* 上的操作指定的名称使用了预定义名称 "continue" 或 "drop"。	重命名操作以使此操作不使用预先定义的名称。
`Failed to resolve src/dst host name for filter at line` *line-number*`, 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。
`Incompatible address version line` *line-number*	*line-number* 上地址的 IP 版本与先前指定的 IP 地址或 `ip_version` 参数不兼容。	更改两个冲突的项使其兼容。

表 32–1 IPQoS 错误消息 （续）

错误消息	说明	解决方案
`Action at line` *line-number* `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。

◆ ◆ ◆ 第 33 章

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：

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 `acctadm` 命令创建一个文件，用来包含 `flowacct` 处理所得的结果。	第 712 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 `flowacct` 参数。	为 `timer`、`timeout` 和 `max_limit` 参数定义值。	第 689 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 34–4。

在记帐客户机的服务级别协议 (Service-Level Agreement, SLA) 中定义的流记帐对于这些记帐客户机特别有用。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量的数据的任务。

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 698 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 715 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 34–4 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 在启用 IPQoS 的系统上承担主管理员角色或成为超级用户。

Primary Administrator（主管理员）角色拥有 Primary Administrator（主管理员）配置文件。有关如何创建该角色并将其指定给用户，请参见《Oracle Solaris 管理：基本管理》中的第 2 章“使用 Solaris Management Console（任务）”。

2 创建基本流记帐文件。

以下示例说明如何为示例 31–1 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
```

acctadm -e	调用带有 -e 选项的 acctadm。-e 选项会启用后跟的参数。
basic	说明仅会在文件中记录 flowacct 的八个基本属性的数据。
/var/ipqos/goldweb/account.info	指定用于包含 flowacct 所获流记录的文件的全限定路径名。
flow	指示 acctadm 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 `flowacct` 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《System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones》中的“Perl Interface to libexacct”。

另请参见

- 有关扩展记帐功能的详细信息，请参阅《System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones》中的第 4 章 “Extended Accounting (Overview)”。
- 要在 IPQoS 配置文件中定义 `flowacct` 参数，请参阅第 689 页中的 “如何在 IPQoS 配置文件中为类启用记帐”。
- 要打印使用 `acctadm` 创建的文件中的数据，请参阅《System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones》中的“Perl Interface to libexacct”。

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 34–5 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

有关技术信息，请参阅 kstat(1M) 手册页。

示例 33–1 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
```

类：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。如果没有流在表中，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 操作实例发现的包的总数，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。

◆ ◆ ◆ 第 34 章

34 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 715 页中的 “IPQoS 体系结构和 Diffserv 模型”
- 第 726 页中的 “IPQoS 配置文件”
- 第 729 页中的 “ipqosconf 配置实用程序”

有关概述信息，请参阅第 29 章，IPQoS 介绍（概述）。有关规划信息，请参阅第 30 章，规划启用了 IPQoS 的网络（任务）。有关 IPQoS 的配置过程，请参阅第 31 章，创建 IPQoS 配置文件（任务）。

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》（《区分服务的体系结构》）(http://www.ietf.org/rfc/rfc2475.txt?number=2475)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 还包括可与虚拟局域网 (Virtual Local Area Network, VLAN) 设备一起使用的流记帐模块和 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高程度的区分服务。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 会选择所有匹配的包。
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 652 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 726 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 34–1 IPQoS 分类器的过滤选定器

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

表 34–1 IPQoS 分类器的过滤选定器 （续）

选定器	参数	选定的信息
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：	IPQoS 计算机上包流的传输方向。
	LOCAL_IN	将本地通信输入到 IPQoS 系统。
	LOCAL_OUT	将本地通信输出到 IPQoS 系统。
	FWD_IN	输入要转发的通信。
	FWD_OUT	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－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－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 配置为作为单速率或双速率计量器运行。tokenmt 操作实例维护两个可确定通信流是否符合已配置参数的令牌桶。

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：

- 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 31–3。

当 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 34–1 针对 IPQoS 配置文件的可识别颜色 tokenmt 操作

```
action {
    module tokenmt
    name meter1
```

示例 34–1 针对 IPQoS 配置文件的可识别颜色 tokenmt 操作 （续）

```
    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 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》（《区别服务的码率整形器》）(http://www.ietf.org/rfc/rfc2963.txt?number=2963)。

标记器模块

IPQoS 包括两个标记器模块：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 656 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 体系结构定义的两种转发行为是 EF 和 AF，它们使用不同的 DSCP。有关 DSCP 的概述信息，请参阅第 656 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《An Expedited Forwarding PHB》（《加速转发 PHB》）(http://www.ietf.org/rfc/rfc2598.txt)。

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 34–2 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《Assured Forwarding PHB Group》（《保证转发 PHB 组》）(`http://tools.ietf.org/html/rfc2597`)。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 34–2 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 34–3 801.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到 machine2 的通信的简单解决方案。

示例 34–2 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 29–1。有关 `flowacct` 的详细技术信息，请参阅 `flowacct(7ipp)` 手册页。

要启用流记帐，您需要使用 Oracle Solaris `exacct` 记帐功能、`acctadm` 命令以及 `flowacct`。有关设置流记帐的所有步骤的信息，请参阅第 711 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 698 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 系统在由 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 记录包含下表中介绍的属性。

表 34–4 flowacct 记录的属性

属性名称	属性内容	类型
src-addr-*address-type*	始发者的源地址。*address-type* 为 v4（对于 IPv4）或 v6（对于 IPv6），如 IPQoS 配置文件中所指定。	基本
dest-addr-*address-type*	包的目标地址。*address-type* 为 v4（对于 IPv4）或 v6（对于 IPv6），如 IPQoS 配置文件中所指定。	基本
src-port	传出流的源端口。	基本
dest-port	要将此流送达的目标端口号。	基本
protocol	流的协议编号。	基本
total-packets	流中的包数。	基本
total-bytes	流中的字节数。	基本
action-name	记录此流的 flowacct 操作的名称。	基本
creation-time	flowacct 首次发现该流的包的时间。	仅扩展
last-seen	上次发现该流的包的时间。	仅扩展
diffserv-field	此流的传出包头中的 DSCP。	仅扩展
user	从应用程序中获取的 UNIX 用户 ID 或用户名。	仅扩展
projid	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 命令创建用于存储由 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 模块将查看流，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* 必须替换为 basic 或 extended。有关每种文件类型中的属性列表，请参阅表 34–4。

-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 31 章，创建 IPQoS 配置文件（任务）。

示例 34–3 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，您必须始终使用 **`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，您必须始终使用有关配置的信息来替换 *action-name* 或 *module-name*。

示例 34–3 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
     class-stats | ""
                 }
class_name  ::= string
next_action_name  ::= string
class_stats ::= enable_stats boolean
boolean ::= TRUE | FALSE
```

示例 34–3 IPQoS 配置文件的语法 （续）

```
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 715 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
}
```

在分类器 action 语句后面跟有 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 34–5 中的模块之一。
params_clause	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。
cf_clauses	一组零或者更多的 class 子句或 filter 子句。

模块定义

模块定义指示哪个模块要处理 action 语句中的参数。IPQoS 配置文件可以包括以下模块。

表 34–5 IPQoS 模块

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

要针对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信息。有关更多信息，请参阅第 727 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 34–1。

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。按类启用统计信息之前，必须先启用 action 语句统计信息。

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。

词汇表

3DES 请参见 Triple-DES（三重 DES）。

address migration（地址迁移） 是指将地址从一个网络接口移动到另一个网络接口的过程。在因接口出现故障而进行故障转移的过程中，或者在因修复接口而进行故障恢复的过程中，均会发生地址迁移。

address pool（地址池） 在移动 IP 中，由家乡网络管理员指定的一组地址，供需要家乡地址的移动节点使用。

AES 高级加密标准 (Advanced Encryption Standard)。一种对称的 128 位块数据加密技术。美国政府在 2000 年 10 月采用该种算法的 Rijndael 变体作为其加密标准。AES 从而取代了 DES 成为政府的加密标准。

agent advertisement（代理通告） 在移动 IP 中，由家乡代理和外地代理定期发送的消息，以通告它们存在于任一已连接链路上。

agent discovery（代理搜索） 在移动 IP 中，移动节点用来确定它是否移动、其当前位置及其在外地网络中的转交地址的过程。

anycast address（任播地址） 为一组接口（通常属于不同的节点）指定的 IPv6 地址。发送到任播地址的包将被路由到**最近的**具有该地址的接口。包的路由符合路由协议的距离度量原则。

anycast group（任播组） 一组具有相同任播 IPv6 地址的接口。Oracle Solaris 实现的 IPv6 不支持创建任播地址和任播组。不过，Oracle Solaris IPv6 节点可以将通信流量发送到任播组。

asymmetric key cryptography（非对称密钥密码学） 一种加密系统，消息的发送者和接收者使用不同的密钥对消息进行加密和解密。非对称密钥用于为对称密钥加密建立一个安全的隧道。Diffie-Hellman algorithm（Diffie-Hellman 算法）就是一种非对称密钥协议。该加密系统与 symmetric key cryptography（对称密钥密码学）相对。

authentication header（验证头） 为 IP 数据报提供验证和完整性而不提供保密性的扩展头。

autoconfiguration（自动配置） 主机根据站点前缀和本地 MAC 地址自动配置其 IPv6 地址的过程。

bidirectional tunnel（双向隧道） 可以双向传输数据报的隧道。

binding table（绑定表） 在移动 IP 中，将家乡地址与转交地址关联的家乡代理表，其中包括剩余生命周期和有效时间。

Blowfish 一种对称块加密算法，它采用 32 位到 448 位的可变长度密钥。其作者 Bruce Schneier 声称 Blowfish 已针对密钥不经常更改的应用程序进行优化。

broadcast address（广播地址） IPv4 网络地址，其主机部分的所有位全为 0 (10.50.0.0) 或全为 1 (10.50.255.255)。从本地网络上的计算机发送到广播地址的包将被传送到该网络中的所有计算机。

CA 请参见 certificate authority, CA（证书颁发机构）。

care-of address（转交地址） 移动节点的临时地址，它在移动节点连接到外地网络时用作隧道退出点。

certificate authority, CA（证书颁发机构） 可信任的第三方组织或公司，可以颁发用于创建数字签名和公钥/私钥对的数字证书。CA 保证被授予唯一证书的个人的身份。

certificate revocation list, CRL（证书撤销列表） 已由 CA 撤销的公钥证书的列表。CRL 存储在 CRL 数据库中，该数据库通过 IKE 进行维护。

classless inter-domain routing (CIDR) address（无类域间路由地址） 一种不基于网络类（A、B 和 C 类）的 IPv4 地址格式。CIDR 地址的长度为 32 位。它们使用标准的 IPv4 点分十进制表示法格式，并添加网络前缀。此前缀定义网络号和网络掩码。

class（类） 在 IPQoS 中，具有类似特征的一组网络流。可以在 IPQoS 配置文件中定义类。

data address（数据地址） 可以用作数据的源地址或目标地址的 IP 地址。数据地址是 IPMP 组的一部分，可以用来发送和接收组中任何接口上的通信。而且，只要 IPMP 组中有一个接口在工作，就可以连续使用 IPMP 组中的一组数据地址。

datagram（数据报） 请参见 IP datagram（IP 数据报）。

DEPRECATED address（DEPRECATED 地址） 在 IPMP 组中不能用作数据的源地址的 IP 地址。通常，IPMP 测试地址为 DEPRECATED。不过，可以将任何地址标记为 DEPRECATED，以防止将该地址用作源地址。

DES Data Encryption Standard（数据加密标准）。一种对称密钥加密方法，开发于 1975 年，1981 年由 ANSI 标准化为 ANSI X.3.92。DES 使用 56 位密钥。

Diffie-Hellman algorithm（Diffie-Hellman 算法） 也称为公钥密码学。Diffie 和 Hellman 于 1976 年开发的非对称密钥一致性协议。使用该协议，两个用户可以在以前没有任何密钥的情况下通过不安全的介质交换密钥。IKE 协议需要使用 Diffie-Hellman。

diffserv model（diffserv 模型） Internet 工程任务组体系结构标准，用于在 IP 网络上实现区分服务。主要模块是分类器、计量器、标记器、调度程序和丢包器。IPQoS 实现分类器、计量器和标记器模块。diffserv 模型在 RFC 2475《*An Architecture for Differentiated Services*》中进行了介绍。

digital signature（数字签名） 附加到以电子方式传输的消息的数字代码，可唯一地标识发送者。

domain of interpretation, DOI（系统解释域） DOI 定义数据格式、网络通信流量交换类型和安全相关信息的命名约定。安全策略、加密算法和加密模式都属于安全相关信息。

DS codepoint, DSCP（DS 代码点） 一个 6 位值，包含在 IP 数据包头的 DS 字段中时指示必须转发包的方式。

DSA Digital Signature Algorithm（数字签名算法）。一种公钥算法，采用大小可变（512 位到 4096 位）的密钥。美国政府标准 DSS 可达 1024 位。DSA 的输入依赖于 SHA-1。

dual stack（双栈） 一种 TCP/IP 协议栈，IPv4 和 IPv6 均位于网络层，栈的其余部分是完全相同的。如果在安装 Oracle Solaris 的过程中启用 IPv6，则主机将收到 TCP/IP 的双栈版本。

dynamic packet filter（动态包过滤器） 请参见 stateful packet filter（有状态包过滤器）。

dynamic reconfiguration, DR（动态重新配置） 一种功能，允许您在系统运行的同时重新配置系统，而对正在进行的操作影响很小或者没有任何影响。并非所有 Oracle Sun 平台都支持 DR。有些 Oracle Sun 平台可能仅支持某些类型硬件（例如 NIC）的 DR。

encapsulating security payload, ESP（封装安全有效负荷） 为数据报提供完整性和保密性的扩展头。ESP 是 IP 安全体系结构 (IPsec) 的五个组件之一。

encapsulation（封装） 在第一个包中放置头和有效负荷的过程，随后将第一个包放置在第二个包的有效负荷中。

failback（故障恢复） 在检测到已修复接口后恢复对该接口的网络访问的过程。

failover（故障转移） 将网络访问从出现故障的接口切换到正常物理接口的过程。网络访问除包括 IPv6 单播和多播通信外，还包括 IPv4 单播、多播和广播通信。

failure detection（故障检测） 检测一个接口或从接口到 Internet 层设备的路径何时不再工作的过程。IP 网络多路径 (IP Network Multipathing, IPMP) 包括两种类型的故障检测：基于链路的故障检测（缺省）和基于探测的故障检测（可选）。

filter（过滤器） IPQoS 配置文件中定义类特性的规则集合。IPQoS 系统选择符合其 IPQoS 配置文件中过滤器的任何通信流以进行处理。请参见 packet filter（包过滤器）。

firewall（防火墙） 将组织的专用网络或内联网与 Internet 隔离，从而防止它受到外部侵入的任何设备或软件。防火墙可以包括包过滤、代理服务器和 NAT（Network Address Translation，网络地址转换）。

flow accounting（流记帐） 在 IPQoS 中，累积和记录有关通信流的信息的过程。通过在 IPQoS 配置文件中定义 `flowacct` 模块的参数，可以建立流记帐。

foreign agent（外地代理） 移动节点访问的外地网络中的路由器或服务器。

foreign network（外地网络） 除移动节点的家乡网络之外的任何网络。

forward tunnel（正向隧道） 开始于家乡代理并结束于移动节点的转交地址的隧道。

Generic Routing Encapsulation, GRE（通用路由封装） 可选的隧道传送形式，可以由家乡代理、外地代理和移动节点支持。GRE 允许将任何网络层协议的包封装在任何其他（或相同）网络层协议的传送包内。

hash value（散列值） 一个从文本字符串生成的数字。使用散列函数可以确保已传输的消息未被篡改。MD5 和 SHA-1 都属于单向散列函数。

header（头） 请参见 IP header（IP 数据包头）。

HMAC 用于进行消息验证的加密散列方法。HMAC 是密钥验证算法。HMAC 与重复加密散列函数（例如 MD5 或 SHA-1）以及机密共享密钥配合使用。HMAC 的加密能力取决于基础散列函数的特性。

home address（家乡地址） 为移动节点指定的 IP 地址，可延用较长的时间。移动节点连接到 Internet 或组织网络中的任何其他位置时，其家乡地址保持不变。

home agent（家乡代理） 移动节点的家乡网络中的路由器或服务器。

home network（家乡网络） 其网络前缀与移动节点家乡地址的网络前缀匹配的网络。

hop（跃点） 用于标识分隔两个主机的路由器数量的度量。如果源主机和目标主机之间有三个路由器，则这两个主机之间有四个跃点。

host（主机） 不执行包转发的系统。在安装 Oracle Solaris 时，系统在缺省情况下成为主机，即系统无法转发包。一个主机通常具有一个物理接口，尽管它可以具有多个接口。

ICMP Internet Control Message Protocol（Internet 控制消息协议）。用于处理错误和交换控制消息。

ICMP echo request packet（ICMP 回显请求包） 发送到 Internet 上的计算机以要求响应的包。此类包通常称为 "ping" 包。

IKE Internet Key Exchange（Internet 密钥交换）。IKE 用于自动为 IPsec 安全关联 (Security Association, SA) 提供经过验证的加密材料。

Internet Protocol, IP（Internet 协议） 在 Internet 上将数据从一台计算机发送到另一台计算机所用的方法或协议。

IP 请参见 Internet Protocol, IP（Internet 协议）、IPv4 和 IPv6。

IP datagram（IP 数据报） 通过 IP 传输的信息包。IP 数据报包含头和数据。头包括数据报的源地址和目标地址。头中的其他字段有助于标识和重新组合目标中数据报附带的数据。

IP header（IP 数据包头） 唯一标识 Internet 包的二十字节数据。该头包括包的源地址和目标地址。头中存在一个选项，该选项允许添加更多字节。

IP in IP encapsulation（IP-in-IP 封装） 封装在 IP 包中的 IP 包的隧道传送机制。

IP link（IP 链路） 通信工具或介质，节点可以通过它在链路层上进行通信。链路层是紧邻 IPv4/IPv6 层的下一层。例如以太网（简单或桥接）或 ATM 网络。可以将一个或多个 IPv4 子网号或前缀指定给一个 IP 链路。不能将一个子网号或前缀指定给多个 IP 链路。在 ATM LANE 中，一个 IP 链路便是一个仿真 LAN。在使用 ARP 时，ARP 协议的范围是单个 IP 链路。

IP stack（IP 栈） TCP/IP 经常被称为“栈”。这是指数据交换的客户机端和服务器端的所有数据传送时所经过的各层（TCP 层、IP 层，有时还经过其他层）。

IPMP group（IPMP 组） 由具有一组数据地址的一组网络接口组成的 IP 多路径组，系统将这些数据地址视为可互换地址，从而可提高网络可用性和利用率。IPMP 组（包括其所有基础 IP 接口和数据地址）由一个 IPMP 接口表示。

IPQoS 一种软件功能，提供 diffserv model（diffserv 模型）标准的实现以及虚拟 LAN 的流记帐和 802.1 D 标记。使用 IPQoS，可以为用户和应用程序提供不同级别的网络服务（如 IPQoS 配置文件中所定义）。

IPsec IP security（IP 安全性）。为 IP 数据报提供保护的安全体系结构。

IPv4 Internet 协议版本 4IPv4 有时称为 IP。此版本支持 32 位地址空间。

IPv6 Internet 协议版本 6IPv6 支持 128 位地址空间。

key management（密钥管理） 管理安全关联 (Security Association, SA) 的方式。

keystore name（密钥库名称） 管理员为 network interface card, NIC（网络接口卡）上的存储区域（或密钥库）指定的名称。密钥库名称也称为标记或标记 ID。

link layer（链路层） 紧邻 IPv4/IPv6 的下一层。

link-local address（链路本地地址） 在 IPv6 中，用于在单个链路上寻址以实现诸如自动配置地址目的的标识。缺省情况下，链路本地地址是从系统的 MAC 地址创建的。

load spreading（负荷分配） 在一组接口中分配传入或外发通信的过程。通过负荷分配，可以获得较高的吞吐量。仅当网络通信流向使用多个连接的多个目标时，才会发生负荷分配。负荷分配有两种类型：传入负荷分配（对于传入通信）和外发负荷分配（对于外发通信）。

local-use address（本地使用地址） 只能在本地范围内（在子网内或在用户网络内）路由的单播地址。此地址还可以具有本地或全局唯一性范围。

marker（标记器） 1. diffserv 体系结构和 IPQoS 中的一个模块，它使用指示包转发方式的值标记 IP 包的 DS 字段。在 IPQoS 实现中，标记器模块是 `dscpmk`。

2. IPQoS 实现中的一个模块，它使用用户优先级值标记以太网数据报的虚拟 LAN 标记。用户优先级值指示使用 VLAN 设备在网络中转发数据报的方式。此模块称为 `dlcosmk`。

MD5 一种重复加密散列函数，用于进行消息验证（包含数字签名）。该函数于 1991 年由 Rivest 开发。

message authentication code, MAC（消息验证代码） MAC 可确保数据的完整性，并验证数据的来源。MAC 不能防止窃听。

meter（计量器） diffserv 体系结构中的一个模块，用于度量特定类的通信流速率。IPQoS 实现包括以下两个计量器：`tokenmt` 和 `tswtclmt`。

minimal encapsulation（最小封装） 家乡代理、外地代理和移动节点支持的可选 IPv4 嵌套隧道传送形式。最小封装的系统开销比 IP-in-IP 封装少 8 或 12 个字节。

mobile node（移动节点） 可以在使用其 IP 家乡地址保持所有现有通信的同时将其连接点从一个网络切换到另一个网络的主机或路由器。

mobility agent（移动代理） 家乡代理或外地代理。

mobility binding（移动绑定） 家乡地址与转交地址的关联以及该关联的剩余生命周期。

mobility security association（移动安全关联） 一对节点之间的安全措施（如验证算法）的集合，这些安全措施应用于在这两个节点之间交换的移动 IP 协议消息。

MTU Maximum Transmission Unit（最大传输单元）。可以通过链路传输的大小，以八位字节表示。例如，以太网的 MTU 是 1500 个八位字节。

multicast address（多播地址） 以特定方式标识一组接口的 IPv6 地址。发送到多播地址的包将被传送到组中的所有接口。IPv6 多播地址与 IPv4 广播地址具有类似的功能。

multihomed host（多宿主主机） 具有多个物理接口且不执行包转发的系统。多宿主主机可以运行路由协议。

NAT 请参见 network address translation（网络地址转换）。

neighbor advertisement（相邻节点通告） 对相邻节点的请求消息的响应，或一个节点发送未经请求的相邻节点通告以通告链路层地址更改的过程。

neighbor discovery（相邻节点搜索） 一种 IP 机制，使主机可以查找驻留在已连接链路上的其他主机。

neighbor solicitation（相邻节点请求） 由一个节点发送的请求，用于确定相邻节点的链路层地址。相邻节点请求还通过高速缓存的链路层地址验证相邻节点是否仍然可以访问。

Network Access Identifier, NAI（网络访问标识符） 以 user@domain 格式唯一标识移动节点的标识。

network address translation（网络地址转换） NAT。将一个网络中使用的 IP 地址转换为另一个网络中已知的不同 IP 地址的过程。用于限制所需的全局 IP 地址的数目。

network interface card, NIC（网络接口卡） 作为网络接口的网络适配卡。一些 NIC 可以具有多个物理接口，如 `igb` 卡。

node（节点） 在 IPv6 中，启用了 IPv6 的任何系统，而不管是主机还是路由器。

outcome（结果） 作为计量通信流量的结果而执行的操作。IPQoS 计量器具有三种结果：红色、黄色和绿色，如在 IPQoS 配置文件中所定义。

packet filter（包过滤器） 一种防火墙功能，可以配置为允许或禁止指定的包通过防火墙。

packet header（包头） 请参见 IP header（IP 数据包头）。

packet（包） 通过通信线路作为一个单位传输的一组信息。包含 IP header（IP 数据包头）以及 payload（有效负荷）。

payload（有效负荷） 通过包传输的数据。有效负荷不包括将包传输到其目标所需的头信息。

per-hop behavior, PHB（单跳行为） 为通信类指定的优先级。PHB 指示该类的流相对其他通信类的优先顺序。

perfect forward secrecy, PFS（完全正向保密） 在 PFS 中，不能使用保护数据传输的密钥派生其他密钥。此外，也不能使用保护数据传输的密钥的源派生其他密钥。

PFS 仅适用于经过验证的密钥交换。另请参见 Diffie-Hellman algorithm（Diffie-Hellman 算法）。

physical interface（物理接口） 系统与链路的连接。此连接通常作为设备驱动程序以及网络接口卡 (Network Interface Card, NIC) 实现。一些 NIC 可以具有多个连接点，例如 `igb`。

PKI Public Key Infrastructure（公钥基础结构）。由数字证书、证书颁发机构和其他注册机构组成的系统，用于检验和验证 Internet 事务中涉及的各方的有效性。

plumb（检测） 打开与物理接口名称关联的设备的行为。在检测接口时，将设置数据流以便 IP 协议可以使用该设备。在系统的当前会话期间，可以使用 `ifconfig` 命令检测接口。

private address（专用地址） 无法通过 Internet 进行路由的 IP 地址。无需 Internet 连通性的主机上的家乡网络可以使用专用地址。这些地址在《Address Allocation for Private Internets》（《私有网络地址分配》）(`http://www.ietf.org/rfc/rfc1918.txt?number=1918`) 中进行了定义，通常称为 "1918" 地址。

protocol stack（协议栈） 请参见 IP stack（IP 栈）。

proxy server（代理服务器） 位于客户机应用程序（如 Web 浏览器）和另一个服务器之间的服务器。用于过滤请求－例如，阻止对某些 Web 站点的访问。

public key cryptography（公钥密码学） 一种加密系统，它使用两种不同的密钥。公钥对所有用户公开。私钥只对消息接收者公开。IKE 为 IPsec 提供公钥。

redirect（重定向） 在路由器中，通告主机有一个更好的第一跃点节点可以到达特定目标。

registration（注册） 移动节点离开家乡网络时使用其家乡代理和外地代理注册其转交地址的过程。

repair detection（修复检测） 检测 NIC 或从 NIC 到某个第 3 层设备的路径在出现故障后何时开始正常工作的过程。

replay attack（重放攻击） 在 IPsec 中，侵入者捕获了包的攻击。存储的包稍后将替换或重复原先的包。为了避免遭到此类攻击，可以在包中包含一个字段，并使该字段在包的保护密钥的生命周期内递增。

reverse tunnel（反向隧道） 开始于移动节点的转交地址并结束于家乡代理的隧道。

router advertisement（路由器通告） 路由器通告其存在以及各种链路和 Internet 参数的过程，要么是定期进行通告，要么是作为对路由器请求消息的响应进行通告。

router discovery（路由器搜索） 主机查找驻留在已连接链路上的路由器的过程。

router solicitation（路由器请求） 主机请求路由器以立即（而非下一个预定时间）生成路由器通告的过程。

router（路由器） 通常具有多个接口、运行路由协议并转发包的系统。如果只有一个接口的系统是 PPP 链路的端点，则可以将该系统配置为路由器。

RSA 获取数字签名和公钥密码系统的方法。该方法于 1978 年首次由其开发者 Rivest、Shamir 和 Adleman 介绍。

SA 请参见 security association, SA（安全关联）。

SADB Security Associations Database（安全关联数据库）。指定密钥和加密算法的表。在数据的安全传输中会使用这些密钥和算法。

SCTP 请参见 streams control transport protocol（流控制传输协议）。

security association, SA（安全关联） 指定从一个主机到另一个主机的安全属性的关联。

security parameter index, SPI（安全参数索引） 指定安全关联数据库 (Security Associations Database, SADB) 中接收者应该用来对收到的包进行解密的行的一个整数。

security policy database, SPD（安全策略数据库） 指定应用于包的保护级别的数据库。SPD 对 IP 通信流量进行过滤，以确定一个包是应该被废弃、应该以明文方式进行传递还是应该用 IPsec 进行保护。

selector（选定器） 一个元素，专门用于定义应用于特定类的包的条件，以便从网络流中选择该类通信流量。可以在 IPQoS 配置文件的过滤子句中定义选定器。

SHA-1 Secure Hashing Algorithm（安全散列算法）。该算法可以针对长度小于 2^{64} 的任何输入进行运算，以生成消息摘要。SHA-1 算法是 DSA 的输入。

site-local-use address（站点本地使用的地址） 用于在单个站点上寻址的标识。

smurf attack（smurf 攻击） 使用从远程位置定向到一个 IP broadcast address（广播地址）或多个广播地址的 ICMP 回显请求包以造成严重的网络拥塞或故障。

sniff（探查） 在计算机网络中窃听－通常作为自动化程序的一部分，以便从线路中筛选出信息，如明文口令。

SPD 请参见 security policy database, SPD（安全策略数据库）。

SPI 请参见 security parameter index, SPI（安全参数索引）。

spoof（电子欺骗） 使用一个 IP 地址（该地址指示消息来自受信任主机）向计算机发送消息，以获取对该计算机的未经授权的访问。要进行 IP 电子欺骗，黑客必须先使用各种方法查找受信任主机的 IP 地址，然后修改包头以便使这些包看起来像是来自该主机。

stack（栈） 请参见 IP stack（IP 栈）。

standby（待机接口） 不用来传输数据通信流量的物理接口，除非某个其他物理接口出现故障。

stateful packet filter（有状态包过滤器） 可以监视活动连接的状态和使用获取的信息确定允许哪些网络包通过 packet filter（包过滤器）的 firewall（防火墙）。通过跟踪和匹配请求与回复，有状态包过滤器可以筛选出与请求不匹配的回复。

stateless autoconfiguration（无状态自动配置） 主机通过组合其 MAC 地址和 IPv6 前缀（由本地 IPv6 路由器通告）生成自己的 IPv6 地址的过程。

stream control transport protocol（流控制传输协议） 以与 TCP 类似的方式提供面向连接的通信的传输层协议。此外，SCTP 还支持连接多宿主，即连接的端点之一可以具有多个 IP 地址。

symmetric key cryptography（对称密钥密码学） 一种加密系统，其中消息的发送者和接收者共享一个公用密钥。此公用密钥用于对消息进行加密和解密。对称密钥用于对在 IPsec 中大量传输的数据进行加密。DES 就是一个对称密钥系统。

TCP/IP TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/Internet 协议）是 Internet 的基本通信语言或协议。它还可以在专用网络（内联网或外联网）中用作通信协议。

test address（测试地址） IPMP 组中只能用作探测器的源地址或目标地址而不能用作数据通信的源地址或目标地址的 IP 地址

Triple-DES（三重 DES） Triple-Data Encryption Standard（三重数据加密标准）。一种对称密钥加密方法。三重 DES 要求密钥长度为 168 位。三重 DES 也写作 3DES。

tunnel（隧道） datagram（数据报）在被封装时跟踪的路径。请参见 encapsulation（封装）。

unicast address（单播地址） 标识启用了 IPv6 的节点的单个接口的 IPv6 地址。单播地址包括以下几部分：站点前缀、子网 ID 和接口 ID。

user-priority（用户优先级） 一个实现服务类标记的 3 位值，它定义如何在 VLAN 设备网络中转发以太网数据报。

virtual LAN (VLAN) device（虚拟 LAN 设备） 在 IP 协议栈的以太网（数据链路）级别上提供通信流量转发的网络接口。

virtual network interface, VNIC（虚拟网络接口） 提供虚拟网络连通性（不论是否是在物理网络接口上配置的）的伪接口。容器（如专用 IP 区域）在 VNIC 上配置以形成虚拟网络。

virtual network（虚拟网络） 软件和硬件网络资源以及作为单个软件项同时管理的功能组合。**内部**虚拟网络将网络资源整合到单个系统，有时称为“网络集成 (network in a box)”。

virtual private network, VPN（虚拟专用网络） 单个安全逻辑网络，使用跨公共网络（如 Internet）的隧道进行传输。

索引

C

D

E

K

L

M

N

O

S

T

U

V

W

安

封

服

负

更

公

故

管

广

规

过

宏

互

回

名

命

目

匿

配

启

刷

双

私

随

隧

为

文

握

无

物

系

细

下

显

相

消

协

新

星

修

虚

选

循

注

转

状

子

自

组

最