配置和管理 Oracle® Solaris 11.1 网络

文件号码 E36458–02
2012 年 10 月

ORACLE

121210@25097

目录

前言

欢迎阅读《配置和管理 Oracle Solaris 11.1 网络》。本书是“建立 Oracle Solaris 11.1 网络”系列的一部分，该系列介绍了配置 Oracle Solaris 网络的基本主题和过程。本书假定您已经安装 Oracle Solaris。您应该已经可以配置网络，或者已经可以配置网络上所需的任何网络软件。

目标读者

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

获取 Oracle 支持

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

印刷约定

下表介绍了本书中的印刷约定。

表 P–1 印刷约定

字体或符号	含义	示例
`AaBbCc123`	命令、文件和目录的名称；计算机屏幕输出	编辑 `.login` 文件。 使用 `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:`

表 P–1 印刷约定 （续）

字体或符号	含义	示例
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 章

规划网络部署

本章简要介绍了规划网络设置时的各种注意事项。这些问题将有助于您以有组织的、具有成本效益的方式部署网络。请注意，本书并不详细讨论如何规划网络，只提供一般说明。

本书假定您熟悉基本网络概念和术语。有关 Oracle Solaris 11 中如何实现 TCP/IP 协议套件的说明，请参见《Oracle Solaris 11 联网介绍》中的“Oracle Solaris 中的网络栈”。

网络规划（任务列表）

下表列出了用于规划网络配置的各个任务。

任务	说明	参考
了解所规划的网络拓扑的硬件要求。	确定网络站点所需的设备类型。	第 12 页中的 “确定网络硬件” 有关特定设备类型的信息，请参阅设备制造商提供的文档。
确定要使用的 IP 地址类型并获取已注册的 IP 地址。	选择是部署纯 IPv4 网络、IPv6 网络，还是部署使用这两种类型的 IP 地址的网络。获取用于与 Internet 中的公共网络通信的唯一 IP 地址。	第 12 页中的 “确定网络的 IP 地址寻址格式” 第 14 页中的 “获取网络的 IP 号”。
确定命名方案以标识网络中的主机以及要使用的名称服务。	创建要指定给网络中的系统的名称列表，并决定是使用 NIS、LDAP、DNS，还是使用本地 /etc 目录中的网络数据库。	第 15 页中的 “管理主机名” 第 15 页中的 “选择名称服务和目录服务”
如有必要，建立管理细分并为子网设计策略。	决定站点是否需要将网络划分为子网以便为管理细分提供服务	第 16 页中的 “使用子网”

任务	说明	参考
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。	第 16 页中的 “为网络规划路由器”
决定是否要在整体网络配置方案中创建虚拟网络。	您可能需要在系统中创建虚拟网络，以减少网络的硬件使用量。	《在 Oracle Solaris 11.1 中使用虚拟网络》

确定网络硬件

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。

您必须做出的有关硬件的某些规划决策遵循：

- 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的类型和数量，包括可能需要的服务器
- 要安装在这些系统中的网络设备
- 要使用的网络介质类型，如以太网等
- 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

注 – 有关路由器如何运作的说明，请参见第 16 页中的 “为网络规划路由器”。有关网桥的概述，请参见《管理 Oracle Solaris 11.1 网络性能》中的“桥接概述”。

确定网络的 IP 地址寻址格式

当您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

简而言之，IP 地址类型包括以下几种：

IPv4 地址

这些 32 位地址是 TCP/IP 的原始 IP 寻址格式。随后，IETF 开发了**无类域间路由** *(Classless Inter-Domain Routing, CIDR)* 地址，作为解决 IPv4 地址缺乏和全局 Internet 路由表容量受限的中短期补救方法。

有关更多信息，请参阅以下资源：

- Internet Protocol DARPA Internet Program Protocol Specification (`http://tools.ietf.org/html/rfc791`)（Internet 协议 DARPA Internet 程序协议规范）
- Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan (`http://tools.ietf.org/html/rfc4632`)（无类域间路由 (Classless Inter-domain Routing, CIDR)：Internet 址分配和聚合计划）

下表列出了 CIDR 表示法和点分十进制格式的子网。

表 1–1 CIDR 前缀及其等效的十进制值

CIDR 网络前缀	等效的点分十进制表示的子网	可用 IP 地址
/19	255.255.224.0	8,192
/20	255.255.240.0	4,096
/21	255.255.248.0	2,048
/22	255.255.252.0	1,024
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32

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《在 Oracle Solaris 11.1 中使用 DHCP》中的第 1 章 “关于 DHCP（概述）”。

IPv6 地址

128 位 IPv6 地址比 IPv4 提供的地址空间大。与 CIDR 格式的 IPv4 地址一样，IPv6 地址也是无类的，使用前缀指定用于定义站点网络的地址部分。有关 IPv6 寻址的详细信息，请参见 Internet Protocol, Version 6 (IPv6) Specification (`http://tools.ietf.org/html/rfc2460`)（Internet 协议版本 6 (IPv6) 规范）。

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。这些专用地址用于专用网络中的网络通信流量。此外，这些地址还用于文档。

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

对于 IPv6 地址，前缀 `2001:db8::/32` 是专用于文档示例的特殊 IPv6 前缀。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。

要使专用网络能够与 Internet 中的外部网络通信，必须从相应的组织为网络获取一个已注册的 IP 号。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`)。

命名网络中的实体

TCP/IP 协议使用系统的 IP 地址在网络中查找系统。但是，与 IP 地址相比，主机名使您可以更容易标识系统。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

为将构成网络的系统规划命名方案。对于充当服务器并具有多个 NIC 的系统，必须提供至少一个与其主网络接口的 IP 地址关联的主机名。

网络中不能有两台具有相同主机名的计算机。因此，每个主机名必须对每个系统唯一。但是，指定了唯一名称的主机或系统可以具有多个 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

在 Oracle Solaris 中，您可以从三种类型的名称服务中选择：本地文件、NIS 和 DNS。名称服务维护有关网络中的计算机的重要信息，例如主机名、IP 地址、以太网地址等。除名称服务以外，您也可以使用 LDAP 目录服务来代替名称服务。有关 Oracle Solaris 中名称服务的介绍，请参阅《在 Oracle Solaris 11.1 中使用命名和目录服务》中的第 I 部分, “关于命名和目录服务”。

在 OS 安装过程中，应提供服务器、客户机或独立系统的主机名和 IP 地址。安装程序会将此信息添加到要供网络服务在为网络提供服务时使用的 `hosts` 数据库中。

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。

对于名称服务，您可以选择以下类型之一：

- NIS 或 DNS－NIS 和 DNS 名称服务维持网络中多个服务器上的网络数据库。《在 Oracle Solaris 11.1 中使用命名和目录服务》介绍了这些名称服务并说明了如何配置数据库。此外，该指南还详细说明了“名称空间”和“管理域”的概念。
- 本地文件－如果您未实现 NIS、LDAP 或 DNS，网络将使用**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

许多网络将其主机和路由器组织到管理域分层结构中。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，例如 `.com`、`.org`、`.edu`、`.gov` 等。

使用子网

使用子网与需要进行管理细分以解决大小和控制问题有关。网络中包含的主机和服务器越多，管理任务就会越复杂。创建管理细分并使用子网后，管理复杂网络将变得更容易。有关为网络设置管理细分的决定由以下因素确定：

- **网络大小**

  即使是在规模相对较小的网络（其细分位于广泛的地理区域）中，子网也是很有用的。

- **用户组共有的公共需求**

  例如，您的网络可能局限在一栋建筑物内，支持相对较少的计算机。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 31 页中的 “在网络上配置组件系统”。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 58 页中的 “配置 IPv6 路由器”。

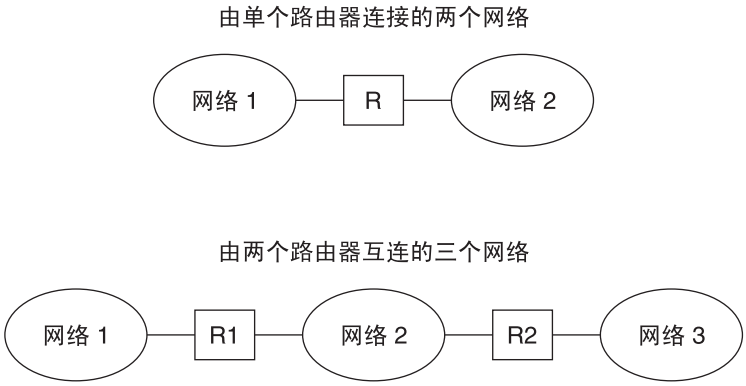
网络拓扑概述

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 37 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。

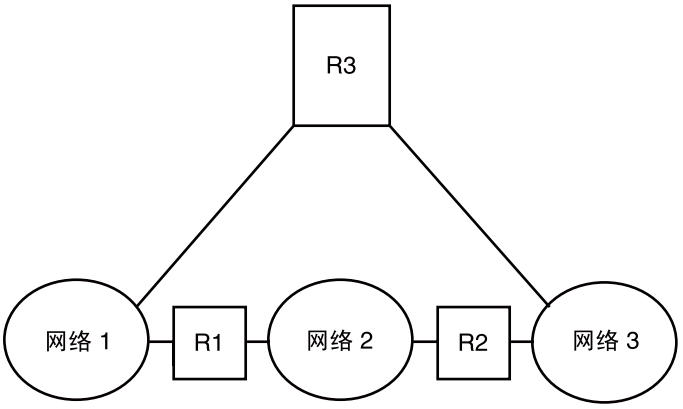
图 1–1 基本网络拓扑



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。

下图显示了一种更为复杂的情况。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 1–2 在网络间提供其他路径的网络拓扑



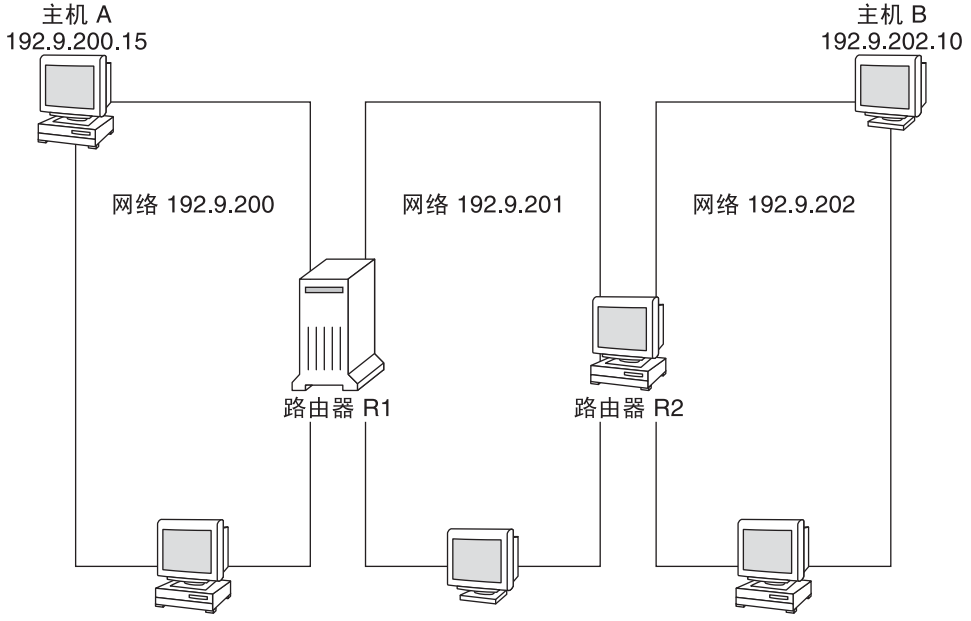
路由器如何传送包

作为包头的一部分的接受者 IP 地址可确定包的路由方式。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 37 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 1–3 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。

部署虚拟网络

此 Oracle Solaris 发行版支持在单个网络中通过配置区域以及虚拟网卡 (VNIC) 来创建虚拟网络。VNIC 是在物理 NIC 上创建的网络接口。组合区域和 VNIC 是将包含大量物理系统的巨型数据中心整合为少量系统的有效方式。有关虚拟联网的更多信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

◆ ◆ ◆ 第 2 章

使用 IPv6 地址的注意事项

如果决定在网络上使用 IPv6 地址，本章通过介绍其他注意事项来对第 1 章，规划网络部署进行了补充说明。

如果除了使用 IPv4 地址外，还计划使用 IPv6 地址，请确保当前的 ISP 支持这两种地址类型。否则，需要找到单独的 ISP 以支持 IPv6 地址。

有关 IPv6 概念的介绍，请参阅以下资源：Internet Protocol, Version 6 (IPv6) Specification (`http://www.ietf.org/rfc/rfc2460.txt`)（Internet 协议版本 6 (IPv6) 规范）。

IPv6 规划（任务列表）

下表列出了计划在网络上实施 IPv6 时的不同注意事项。

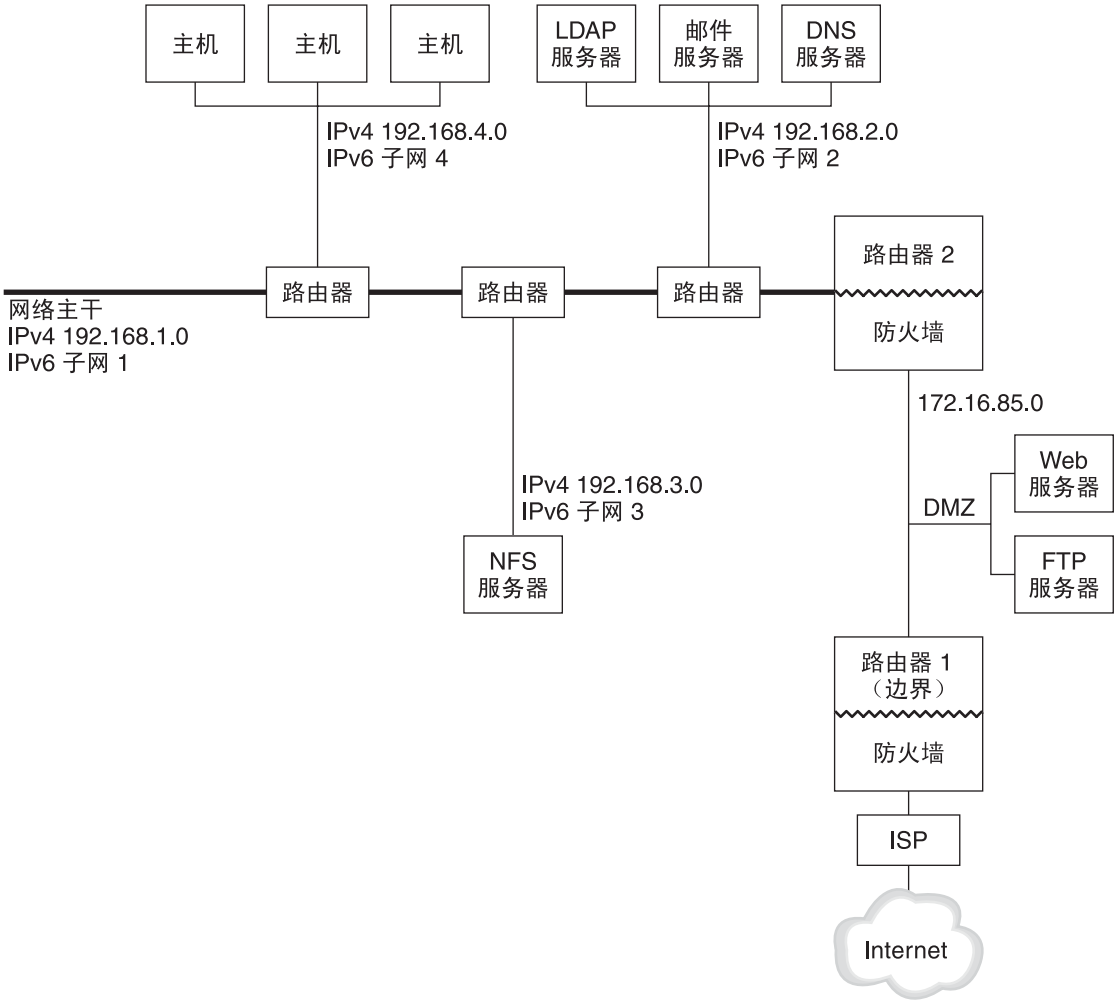
任务	说明	参考
准备硬件以支持 IPv6。	确保硬件可以升级到 IPv6。	第 24 页中的 “确保硬件支持 IPv6”
确保应用程序能够支持 IPv6。	验证应用程序是否可以在 IPv6 环境中运行。	第 26 页中的 “配置网络服务以支持 IPv6”
制定隧道使用计划。	确定应当使用哪些路由器来建立与其他子网或外部网络连接的隧道。	第 28 页中的 “网络中使用隧道的规划”

任务	说明	参考
计划如何保证网络安全并制定 IPv6 安全策略。	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。 决定实现安全的方式，例如使用此发行版的 IP 过滤器、IP 安全体系结构 (IPsec)、Internet 密钥交换 (Internet Key Exchange, IKE) 和其他安全功能。	第 28 页中的 “IPv6 实现的安全注意事项” 《在 Oracle Solaris 11.1 中保护网络安全》
为网络上的系统制定寻址计划。	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。此步骤包括获取网络的站点前缀以及规划 IPv6 子网（如果需要）。	第 25 页中的 “为节点制定 IPv6 寻址计划”

IPv6 网络拓扑方案

通常，IPv6 用于也使用 IPv4 的混合网络拓扑中，如下图所示。此图用作后续小节中说明 IPv6 配置任务的参考信息。

图 2–1 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 2–1 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。

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 期间和之后，可以成功运行 IPv4 相关的操作。在已使用 IPv4 的操作网络上部署 IPv6 时，请确保不要中断正在进行的操作。

以下小节说明了在准备实现 IPv6 时需要考虑的事项。

确保硬件支持 IPv6

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 – 本书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。有关详细信息和解决办法，请参见《Troubleshooting Network Issues》中的“IPv4 Router Cannot Be Upgraded to IPv6”。

对于 IPv6 服务器的每个 NIC，手动配置 IPv6 地址的接口 ID 部分，而不是使用相邻节点搜索协议自动获取该 ID。通过此方式，如果替换了 NIC，则可以将相同的接口 ID 应用于替换的 NIC。由相邻节点搜索协议自动生成的不同 ID 可能会导致服务器行为异常。

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 25 页中的 “获取站点前缀”
- 第 25 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。

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。有关详细信息，请参见《Troubleshooting Network Issues》中的"Current ISP Does Not Support 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 2–1 中 LDAP 服务器的内部接口可能会变为 `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 2–1 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）。

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 2–1 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`

配置网络服务以支持 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 发行版或 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《Troubleshooting Network Issues》中的“Problems After Upgrading Services to IPv6”。

▼ 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 28 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 27 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《在 Oracle Solaris 11.1 中管理 IP 服务质量》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 IPv6 的 DNS 支持相关的更多信息，请参阅《在 Oracle Solaris 11.1 中使用命名和目录服务》。

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。

网络中使用隧道的规划

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 2–1 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 99 页中的 “配置隧道（任务列表）”。有关隧道的相关概念的信息，请参阅第 91 页中的 “IP 隧道概述”。

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《在 Oracle Solaris 11.1 中保护网络安全》中的第 6 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《在 Oracle Solaris 11.1 中保护网络安全》中的第 9 章 “Internet 密钥交换（概述）”。

◆ ◆ ◆ 第 3 章

3 配置 IPv4 网络

网络配置涉及两个阶段：组装硬件，然后配置用于实现 TCP/IP 协议的守护进程、文件和服务。

本章介绍如何配置用于实现 IPv4 寻址和服务的网络。

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。特定于 IPv6 网络的任务位于第 4 章，在网络上启用 IPv6。

注 – 在配置 TCP/IP 之前，请查看第 1 章，规划网络部署中列出的各种规划任务。如果您计划使用 IPv6 地址，则另请参阅第 2 章，使用 IPv6 地址的注意事项。

本章包含以下信息：

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
配置系统的 IP 接口。	为系统的 IP 接口指定 IP 地址。	《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”
以本地文件模式配置系统	编辑系统 /etc 目录中的特定配置文件，并配置 nis/domain SMF 服务。	第 34 页中的 “如何以本地文件模式配置系统”
设置网络配置服务器	启用 in.tftp 守护进程，并编辑系统 /etc 目录中的其他配置文件。	第 36 页中的 “如何设置网络配置服务器”
以网络客户机模式配置系统	编辑系统 /etc 目录中的配置文件。	第 36 页中的 “如何以网络客户机模式配置系统”
为网络客户机指定路由策略	将系统配置为使用静态路由或动态路由。	第 45 页中的 “如何在单接口主机上启用静态路由”和第 46 页中的 “如何在单接口系统上启用动态路由”。

开始网络配置之前

在此 Oracle Solaris 发行版中，系统的网络配置由活动的**网络配置文件** *(network configuration profile, NCP)* 管理。如果活动 NCP 为反应性，则系统的网络配置模式为自动，例如，automatic NCP。如果活动 NCP 为 DefaultFixed，则系统的网络配置模式为固定。采用反应性网络配置的系统与采用固定网络配置的系统在行为方式上有所不同。

您所做的任何配置都将应用于活动 NCP。因此，在执行任何配置过程之前，您必须首先知道哪个 NCP 处于活动状态。这样，在您完成配置过程之后，系统的行为方式才会像预想的那样。要确定系统中的活动 NCP，请键入以下命令：

```
# netadm list
TYPE        PROFILE        STATE
ncp         DefaultFixed   online
ncp         Automatic      disabled
loc         Automatic      offline
loc         NoNet          offline
loc         User           offline
loc         DefaultFixed   online
```

状态为 online（联机）的配置文件是系统中的活动 NCP。

要了解有关系统中 NCP 的详细信息，请使用带 -x 选项的 netadm 命令。

```
netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         DefaultFixed   online         active
```

```
ncp         Automatic      disabled       disabled by administrator
loc         Automatic      offline        conditions for activation are unmet
loc         NoNet          offline        conditions for activation are unmet
loc         User           offline        conditions for activation are unmet
loc         DefaultFixed   online         active
```

要切换配置文件类型，如从反应性配置文件切换到固定配置文件，请键入以下命令：

```
# netadm enable -p ncp NCP-name
```

其中 *NCP-name* 是某一类型的 NCP 的名称。

有关配置文件管理的网络配置的介绍，请参见《Oracle Solaris 11 联网介绍》中的“网络配置文件”。有关 NCP 的详细说明，请参阅《在 Oracle Solaris 11.1 中使用反应性网络配置连接系统》。

在网络上配置组件系统

当您配置网络系统时，需要以下配置信息：

- 每个系统的主机名。
- 每个系统的 IP 地址和网络掩码。如果网络被细分为子网，则您必须提供子网号和 IP 地址架构，才能应用于每个子网中的系统，包括其各自的网络掩码。
- 每个系统所属的域名。
- 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。有关路由器以及 Oracle Solaris 支持的路由协议列表的更多信息，请参见第 112 页中的“Oracle Solaris 中的路由协议”。

注 – 您可以在安装 Oracle Solaris 时配置网络。有关说明，请参见《安装 Oracle Solaris 11.1 系统》。

本文档中的各个过程假定您要在安装 OS 后配置网络。

可参考下节中的图 3–1 配置网络组件系统。

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。

图 3–1 具有多个 IPv4 路由器的自治系统

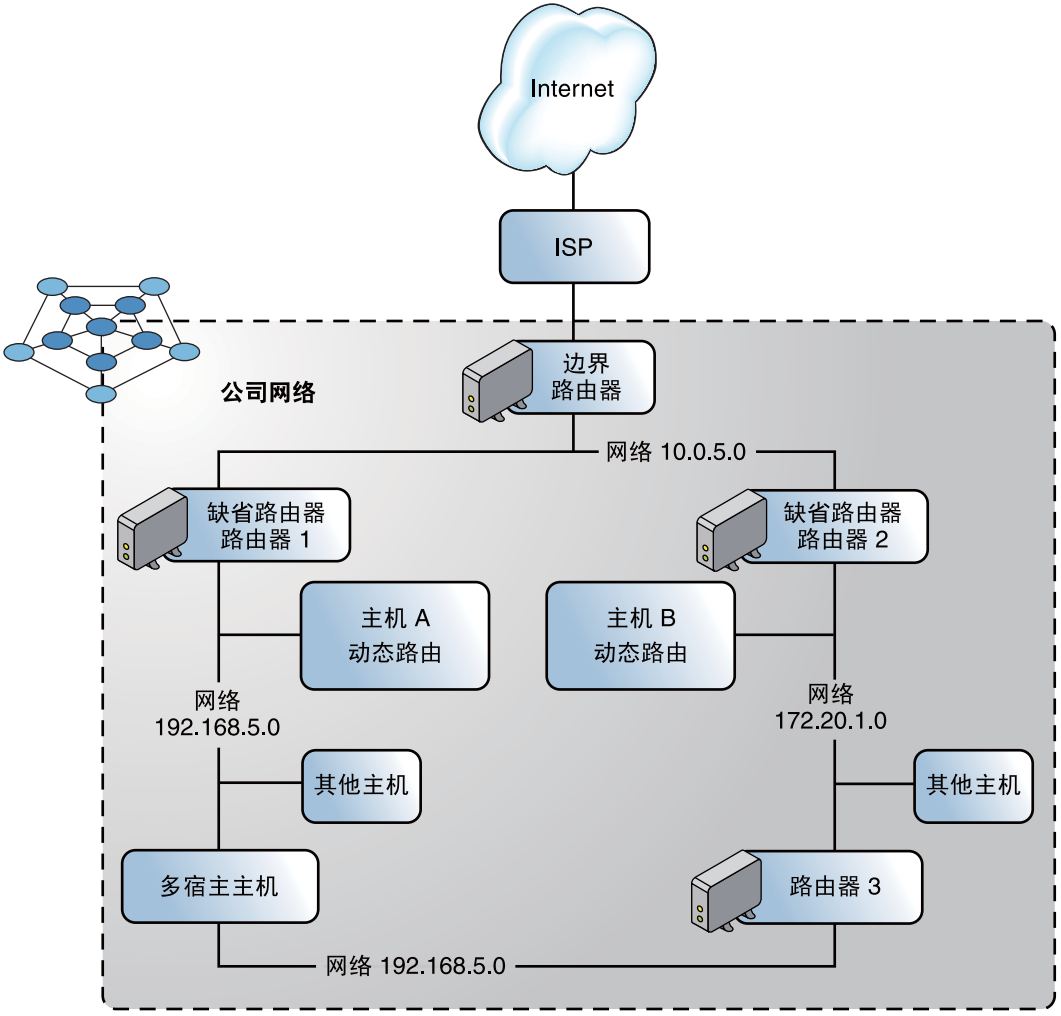


图 3–1 展示了一个 AS，它分成了三个本地网络 `10.0.5.0`、`172.16.1.0` 和 `192.168.5.0`。该网络由以下类型的系统组成：

- 路由器使用路由协议管理网络包如何从其源定向或路由至本地网络中的目标，或者定向或路由至外部网络。有关 Oracle Solaris 支持的路由协议的信息，请参见第 113 页中的 “Oracle Solaris 中的路由协议表”。

  路由器的类型划分如下：

  - **边界路由器**用于将本地网络（如 `10.0.5.0`）以外部方式连接到服务提供商。

  - **缺省路由器**用于管理本地网络中的包路由，其本身可能包含多个本地网络。例如，在图 3–1 中，路由器 1 充当 192.168.5 的缺省路由器。同时，路由器 1 还连接到 10.0.5.0 内部网络。路由器 2 的接口连接到 10.0.5.0 和 172.16.1.0 内部网络。
  - **包转发路由器**用于转发内部网络之间的包，但不运行路由协议。在图 3–1 中，路由器 3 是一个包转发路由器，具有到 172.16.1 和 192.168.5 网络的连接。
- 客户机系统
  - 多宿主系统或具有多个 NIC 的系统。在 Oracle Solaris 中，这些系统缺省情况下可以将包转发到同一网络段中的其他系统。
  - 单接口系统依赖于本地路由器进行包转发和接收配置信息。

设置系统配置模式

本节介绍了将系统设置为以**本地文件模式**或**网络客户机模式**运行的过程。以本地文件模式运行时，系统会从本地目录中的文件获取所有 TCP/IP 配置信息。在网络客户机模式下，配置信息通过远程网络配置服务器提供给网络中的所有系统。

通常，网络中的服务器是以本地文件模式运行的，例如以下服务器：

- 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
- 路由器

客户机可在任意一种模式下运行。因此，在网络中，您可以组合这些用于配置不同系统的模式，如下图所示。

图 3–2 IPv4 网络拓扑方案中的系统

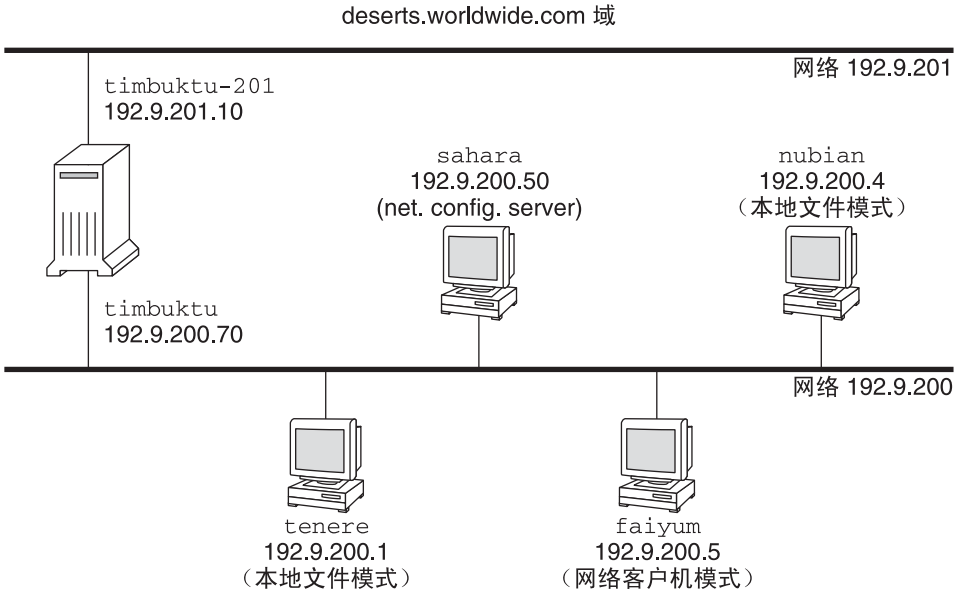


图 3–2 显示了 192.9.200 网络中的系统。

- 所有系统均属于组织域 deserts.worldwide.com。
- sahara 为配置服务器。该服务器在本地文件模式下运行，其中 TCP/IP 配置信息是从系统的本地磁盘获取的。

  注 – 如果您将客户机配置为在网络客户机模式下运行，则必须至少配置一个网络配置服务器，用于为这些客户机提供配置信息。

- tenere、nubian 和 faiyum 为网络中的客户机。tenere 和 nubian 在本地文件模式下运行。无论 faiyum 的本地磁盘如何，该系统都会配置为以网络客户机模式运行。
- timbuktu 配置为路由器，因此在本地文件模式下运行。该系统包括两个 NIC，每一个 NIC 都有各自的已配置 IP 接口。第一个 IP 接口名为 timbuktu 并连接到网络 192.9.200。第二个 IP 接口名为 timbuktu-201 并连接到网络 192.9.201。

▼ 如何以本地文件模式配置系统

使用以下过程配置任何要以本地文件模式运行的系统。

1. **使用指定的 IP 地址配置系统的 IP 接口。**

   有关过程，请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

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

3. **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。

   此文件还必须包含缺省路由器的名称和路由器的 IP 地址。

   a. **（可选）为安装后添加到系统的任何网络接口添加 IP 地址和对应名称。**

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

4. **将系统的全限定域指定为 nis/domain SMF 服务的属性。**

   例如，可以将 deserts.worldwide.com 指定为 nis/domain SMF 服务的 domainname 属性的值，如下所示：

   ```
   # domainname domainname
   ```

   该步骤将影响持久性更改。

5. **在 /etc/defaultrouter 文件中键入路由器的名称。**

6. **添加网络掩码信息（如果适用）。**

   注 – 如果您使用的是 DHCP 服务，请跳过此步骤。

   a. **在 /etc/inet/netmasks 文件中键入网络号和网络掩码。**

      要创建项，请使用格式 *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 1–1 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. **在交换机的 SMF 属性中更改网络掩码查找顺序，以便首先搜索本地文件，然后刷新实例。**

      ```
      # svccfg -s name-service/switch setprop config/host = astring: '"files nis"'
      # svccfg -s name-service/switch:default refresh
      ```

7. **重新引导系统。**

▼ 如何以网络客户机模式配置系统

对要在网络客户机模式下配置的每个主机执行以下过程。

开始之前 网络客户机从网络配置服务器接收其配置信息。因此，在将系统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。

1. **成为管理员。**

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

2. **使用指定的 IP 地址配置系统的 IP 接口。**

   有关过程，请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

3. **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
   ```

4. **删除指定给 `nis/domain` SMF 服务的 `domainname` 属性的值。**

   ```
   # domainname ''
   ```

   该步骤将影响持久性更改。

5. **确保客户机的 `name-service/switch` 服务中的搜索路径反映了网络的相同服务要求。**

▼ 如何设置网络配置服务器

用于设置安装服务器和引导服务器的信息可在《安装 Oracle Solaris 11.1 系统》中找到。

1. **成为管理员。**

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

2. **按照如下所示打开 `in.tftpd` 守护进程：**

   a. **导航到指定的网络配置服务器的根目录 (`/`)：**

   b. **创建 `/tftpboot` 目录：**

      ```
      # mkdir /tftpboot
      ```

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

c. **创建指向目录的符号链接。**

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

3 **在 /etc/inetd.conf 文件中添加 tftp 行。**

该行应显示如下：

```
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。

4 **在 /etc/hosts 数据库上，添加网络中所有客户机的主机名和 IP 地址。**

5 **在 /etc/ethers 数据库上，为网络中以网络客户机模式运行的每个系统创建项。**

此数据库中的项使用以下格式：

MAC Address *host name* *#comment*

有关更多信息，请参见 ethers(4) 手册页。

6 **在 /etc/bootparams 数据库上，为网络中以网络客户机模式运行的每个系统创建项。**

有关编辑此数据库的信息，请参见 bootparams(4) 手册页。

7 **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
```

8 **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《在 Oracle Solaris 11.1 中管理服务和故障》中的第 1 章 “管理服务（概述）”。

配置 IPv4 路由器

路由器用于提供两个或多个网络之间的接口。因此，您必须为路由器的每个物理网络接口指定唯一的名称和 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。如果某个单接口系统充当 PPP 链路上的一个端点，则可以将该系统配置为路由器，如《在 Oracle Solaris 11.1 中使用 UUCP 和 PPP 管理串行网络》中的“规划拨号 PPP 链路”中所述。

▼ 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 在将路由器以物理方式安装到网络上后，请将路由器配置为以本地文件模式运行，如第 34 页中的 “如何以本地文件模式配置系统”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. **成为管理员。**

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

2. **配置系统中 NIC 上的 IP 接口。**

   有关配置 IP 接口的详细步骤，请参见《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

   确保每个 IP 接口都配置有系统将路由其包的网络的 IP 地址。因此，如果系统为 `192.168.5.0` 和 `10.0.5.0` 网络提供服务，则必须为每个网络配置一个 NIC。

注意 – 如果您要配置 IPv4 路由器以使用 DHCP，则必须完全了解 DHCP 管理。

3. **将每个接口的主机名和 IP 地址添加到 `/etc/inet/hosts` 文件中。**

   例如，假定您为路由器 1 的两个接口指定的名称分别为 `krakatoa` 和 `krakatoa-1`。`/etc/inet/hosts` 文件中的项将如下所示：

   ```
   192.168.5.1      krakatoa        #interface for network 192.168.5.0
   10.0.5.1         krakatoa-1      #interface for network 10.0.5.0
   ```

4. **执行其余步骤，将此路由器配置为在本地文件模式下运行。**

   请参见第 34 页中的 “如何以本地文件模式配置系统”。

5. **如果路由器连接到具有子网的任何网络，请将网络号和网络掩码添加到 `/etc/inet/netmasks` 文件中。**

   例如，对于传统的 IPv4 地址表示法（如 `192.168.5.0`），应键入：

   ```
   192.168.5.0     255.255.255.0
   ```

6 **在路由器上启用 IPv4 包转发。**

```
# ipadm set-prop -p forwarding=on ipv4
```

7 **（可选的）启动路由协议。**

使用以下命令语法之一：

- `# routeadm -e ipv4-routing -u`
- `# svcadm enable route:default`

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

当您启动路由协议时，路由守护进程 `/usr/sbin/in.routed` 将自动更新路由表（该过程称为**动态路由**）。有关路由类型的更多信息，请参见第 40 页中的 “路由表和路由类型”。有关 `routeadm` 命令的信息，请参见 `routeadm(1M)` 手册页。

示例 3–1 配置网络的缺省路由器

此示例基于图 3–1。路由器 2 包含两个有线网络连接，一个是与网络 `172.16.1.0` 的连接，另一个是与网络 `10.0.5.0` 的连接。此示例说明如何配置路由器 2，使其成为 `172.16.1.0` 网络的缺省路由器。此示例还假定路由器 2 已配置为在本地文件模式下工作，如第 34 页中的 “如何以本地文件模式配置系统”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。

```
# dladm show-link
LINK     CLASS     MTU     STATE   BRIDGE   OVER
net0     phys      1500    up      --       --
net1     phys      1500    up      --       --
net2     phys      1500    up      --       --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           172.16.1.10/24
```

只有 `net0` 配置有 IP 地址。要将路由器 2 设置为缺省路由器，应将 `net1` 接口以物理方式连接到 `10.0.5.0` 网络。

```
# ipadm create-ip net1
# ipadm create-addr -a 10.0.5.10/24 net1
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           172.16.1.10/24
net1/v4           static   ok           10.0.5.10/24
```

接下来，应使用有关新配置的接口和与该接口相连的网络的信息，更新以下网络数据库：

```
# vi /etc/inet/hosts
127.0.0.1       localhost
172.16.1.10        router2        #interface for network 172.16.1
```

```
10.0.5.10          router2-out    #interface for network 10.0.5
# vi /etc/inet/netmasks
172.16.1.0     255.255.255.0
10.0.5.0       255.255.255.0
```

最后，启用包转发以及 in.routed 路由守护进程。

```
# ipadm set-prop -p forwarding=on ipv4
# 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 172.16.1.0 的缺省路由器配置尚未完成。您需要执行以下操作：

- 修改 172.16.1.0 网络中的每个主机，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 45 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 40 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**。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，可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              172.16.1.10          UG        1    532   net0
224.0.0.0            10.0.5.100           U         1      0   net1
10.0.0.0             10.0.5.100           U         1      0   net1
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），以路由网络通信流量以及更新表中的路由信息。对于**静态路由**，路由信息是通过 route 命令手动维护的。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。

路由类型	最适用于
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 3–1 所示的 AS 将静态路由和动态路由组合在一起。

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。如果您需要这些功能，请使用 IPMP，如《管理 Oracle Solaris 11.1 网络性能》中的第 5 章 “IPMP 介绍”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   net0
224.0.0.0            198.168.5.10          U      1  0       net0
default              192.168.5.10          UG     1  91908
127.0.0.1            127.0.0.1             UH     1  811302   lo0
```

2 成为管理员。

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

3 （可选的）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。
`-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 3–2 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图 3–1 的路由器 2。AS 的边界路由器 10.0.5.150 需要静态路由。

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              172.16.1.10          UG        1    249 ce0
224.0.0.0            172.16.1.10          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 172.16.1.10 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              172.16.1.10          UG        1    249 ce0
224.0.0.0            172.16.1.10          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《管理 Oracle Solaris 11.1 网络性能》中的第 5 章 “IPMP 介绍”。

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：

- 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《Oracle Solaris 11.1 管理：安全服务》中的“防火墙系统”或第三方防火墙的相应文档。

▼ 如何创建多宿主主机

1 成为管理员。

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

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

请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

3 如果启用了包转发，请将此服务禁用。

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY      PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding    rw   on           --           off          on,off

ipadm set-prop -p forwarding=off ipv4
```

4 （可选的）为多宿主主机打开动态路由。

使用以下命令语法之一：

- `# routeadm -e ipv4-routing -u`
- `# svcadm enable route:default`

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

示例 3–3 配置多宿主主机

以下示例说明如何配置图 3–1 中所示的多宿主主机。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
LINK     CLASS     MTU     STATE   BRIDGE   OVER
net0     phys      1500    up      --       --
net1     phys      1500    up      --       --

# ipadm show-addr
ADDROBJ        TYPE     STATE        ADDR
lo0/v4         static   ok           127.0.0.1/8
net0/v4        static   ok           192.168.5.82/24
```

dladm show-link 命令报告 hostc 具有两个数据链路。但是，只有 net0 配置有 IP 地址。要将 hostc 配置为多宿主主机，应使用同一 192.168.5.0 网络中的 IP 地址配置 net1。确保 net1 的底层物理 NIC 已实际连接到网络。

```
# ipadm create-ip net1
# ipadm create-addr static -a 192.168.5.85/24 net1
# ipadm show-addr
ADDROBJ        TYPE     STATE        ADDR
lo0/v4         static   ok           127.0.0.1/8
net0/v4        static   ok           192.168.5.82/24
net1/v4        static   ok           192.168.5.85/24
```

接下来，将 net1 接口添加到 /etc/hosts 数据库：

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

接下来，关闭包转发（如果此服务正在 hostc 上运行）：

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on        --           off          on,off

# ipadm set-prop -p forwarding=off ipv4

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

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

routeadm 命令报告当前启用了通过 in.routed 守护进程的动态路由。

为单接口系统配置路由

单接口系统可以使用静态路由或动态路由进行配置。对于静态路由，主机必须依赖于缺省路由器的服务来获取路由信息。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

也可以使用以下过程在多宿主主机上配置静态路由。

1. **成为管理员。**

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

2. **使用系统所属网络的 IP 地址配置系统的 IP 接口。**

   有关说明，请参见《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

3. **使用文本编辑器创建 `/etc/defaultrouter` 文件，或添加系统将使用的路由器的 IP 地址以修改该文件。**

4. **在本地 `/etc/inet/hosts` 文件中添加缺省路由器的项。**

5. **确保路由已关闭。**

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

          Routing services   "route:default ripng:default"

# svcadm disable route:default
```

6. **确保包转发已关闭。**

```
# # ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on        --           off          on,off

# ipadm set-prop -p forwarding=off ipv4
```

示例 3–4 在单接口系统上配置静态路由

以下示例说明如何为图 3–1 所示的 `172.16.1.0` 网络中的单接口系统 `hostb` 配置静态路由。`hostb` 需要将路由器 2 用作其缺省路由器。此示例假定您已配置了系统的 IP 接口。

首先，您应使用管理员权限登录 hostb。接下来，确定系统中是否存在 /etc/defaultrouter 文件：

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

# vi /etc/defaultrouter
172.16.1.10
```

IP 地址 172.16.1.10 属于路由器 2。

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

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on          --           off          on,off

# ipadm set-prop -p forwarding=off ipv4

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

          Routing services   "route:default ripng:default"

# svcadm disable route:default
```

▼ 如何在单接口系统上启用动态路由

使用路由协议的动态路由是管理系统上的路由的最简便方法。

1. **成为管理员。**

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

2. **使用系统所属网络的 IP 地址配置系统的 IP 接口。**

   有关说明，请参见《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

3. **删除 /etc/defaultrouter 文件中的任何项。**

   /etc/defaultrouter 文件为空会强制系统使用动态路由。

4 **确保包转发已禁用。**

```
# ipadm set-prop -p forwarding=off ipv4
```

5 **在系统上启用路由协议。**

使用以下命令之一：

- **# routeadm -e ipv4-routing -u**
- **# svcadm enable route:default**

示例 3–5 在单接口系统上运行动态路由

以下示例说明如何为图 3–1 所示的网络 192.168.5.0 中的单接口系统 hosta 配置动态路由。该系统将路由器 1 用作其缺省路由器。此示例假定您已配置了系统的 IP 接口。

首先，您应使用管理员权限登录 hosta。如果系统上存在 /etc/defaultrouter 文件，接下来您需要将该文件删除：

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

# rm defaultrouter

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

          Routing services   "route:default ripng:default"

# svcadm enable route:default

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on          --           off          on,off

# ipadm set-prop -p forwarding=off ipv4
```

将子网添加到网络

如果要将不使用子网的网络更改为使用子网的网络，请执行以下列表中的任务。该列表假定您已准备好子网架构。

- 将含有新子网号的 IP 地址指定给子网所属的系统。

  有关参考信息，请参见《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

- 将正确的 IP 地址和网络掩码添加到每个系统的 /etc/netmasks 文件中。

- 使用正确的 IP 地址修正每个系统的 `/etc/inet/hosts` 文件，以便与主机名相符。
- 重新引导子网中的所有系统。

以下过程与子网密切相关。如果您实现子网划分的时间远迟于初始配置网络而未划分子网后的时间，请执行以下过程实现更改。

▼ 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《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 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。

1. **成为管理员。**

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

2. **使用 `ipadm` 命令修改 IP 地址。**

   您不能使用 `ipadm` 命令直接修改 IP 地址。您应该首先删除表示想要修改的 IP 地址的地址对象。然后使用相同的地址对象名称来指定一个新地址。

   ```
   # ipadm delete-addr addrobj
   # ipadm create-addr -a IP-address  interface
   ```

3. **如果适用，修改 `system/identity:node` SMF 服务中的主机名项：**

   ```
   # hostname newhostname
   ```

   该步骤将影响持久性更改。

4. **如果子网掩码已更改，请修改 `/etc/netmasks` 文件中的子网项。**

5. **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。**

6. **重新引导系统。**

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

示例 3–6 更改 IP 地址和主机名

此示例说明如何更改主机名、主网络接口的 IP 地址和子网掩码。主网络接口 `net0` 的 IP 地址从 `10.0.0.14` 更改为 `192.168.34.100`。

```
# ipadm show-addr
ADDROBJ       TYPE     STATE   ADDR
lo0/v4        static   ok      127.0.0.1/8
net0/v4       static   ok      10.0.0.14/24

# ipadm delete-addr net0/v4
# ipadm create-addr -a 192.168.34.100/24 net0
# hostname mynewhostname

# ipadm show-addr
ADDROBJ          TYPE     STATE   ADDR
lo0/v4           static   ok      127.0.0.1/8
net0/v4          static   ok      192.168.34.100/24

# hostname
mynewhostname
```

另请参见 要更改主网络接口以外的接口的 IP 地址，请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

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《在 Oracle Solaris 11.1 中管理服务和故障》中的第 1 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 110 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 **成为管理员。**

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

2 **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《在 Oracle Solaris 11.1 中管理服务和故障》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 3–7 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《在 Oracle Solaris 11.1 中管理 sendmail 服务》中的“sendmail 版本 8.12 支持 TCP 包装”中所述。

1. **成为管理员。**

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

2. **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 `/usr/sfw/man` 目录中找到。

◆ ◆ ◆ 第 4 章

在网络上启用 IPv6

本章包含有关在网络上启用 IPv6 的任务。本章主要包含以下主题：

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：

- `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。

▼ 如何针对 IPv6 配置系统

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 126 页中的 “自动配置过程”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 58 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。

1 **使用相应命令配置 IP 接口。**

请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“如何配置 IP 接口”。

注 – 指定 IP 地址时，请确保使用正确的选项指定 IPv6 地址：

```
# ipadm create-addr -T addrconf interface
```

要添加更多地址，请使用以下语法：

```
# ipadm create-addr -a ipv6-address interface
```

2 **启动 IPv6 守护进程 `in.ndpd`。**

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

3 **（可选）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 60 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 115 页中的“ndpd.conf 配置文件”。

5 **（可选）要显示 IP 接口的状态及其 IPv6 配置，请键入以下命令：**

```
# ipadm show-addr
```

示例 4–1 在安装之后启用 IPv6 接口

此示例说明如何在 `net0` 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。

```
# ipadm show-addr
ADDROBJ    TYPE     STATE   ADDR
lo0/v4     static   ok      127.0.0.1/8
net0/v4    static   ok      172.16.27.74/24
```

目前仅为该系统配置了 `net0` 接口。请按如下所示在该接口上启用 IPv6：

```
# ipadm create-addr -T addrconf net0
# ipadm create-addr -a 2001:db8:3c4d:15:203/64 net0
# /usr/lib/inet/in.ndpd

# ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      172.16.27.74/24
net0/v6       addrconf   ok      fe80::203:baff:fe13:14e1/10
lo0/v6        static     ok      ::1/128
net0/v6a      static     ok      2001:db8:3c4d:15:203/64

# route -p add -inet6 default fe80::203:baff:fe13:14e1
```

接下来的步骤

- 要将 IPv6 节点配置为路由器，请转至第 58 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 57 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 64 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 62 页中的 “配置 IPv6 标记”中所述。

1. **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。要在服务器上关闭接口的地址自动配置，该文件必须具有以下内容:

   interface `StatelessAddrConf false`

   要关闭所有接口的地址自动配置，请使用以下项：

   ```
   ifdefault StatelessAddrConf false
   ```

   有关 `/etc/inet/ndpd.conf` 的详细信息，请参阅 `ndpd.conf(4)` 手册页和第 115 页中的 “ndpd.conf 配置文件”。

2. **使用所做更改更新 IPv6 守护进程。**

   ```
   # pkill -HUP in.ndpd
   ```

配置 IPv6 路由器

本节介绍配置 IPv6 路由器的任务。根据站点要求，可能需要仅执行选定的任务。

▼ 如何配置启用了 IPv6 的路由器

以下过程假定您已为系统进行了 IPv6 配置。有关过程，请参阅第 55 页中的 “配置 IPv6 接口”。

1 **在路由器的所有接口上配置 IPv6 包转发。**

```
# ipadm set-prop -p forwarding=on ipv6
```

2 **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：

- 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。

3 **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 115 页中的 “ndpd.conf 配置文件”和 `ndpd.conf(4)` 手册页。

4 **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。

5 **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 net0 和 net1 通告站点前缀 2001:0db8:3c4d::/48。

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on

if net0 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:15::0/64 net0

if net1 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:16::0/64 net1
```

6 **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 4–2 ipadm show-addr 输出显示 IPv6 接口

以下示例显示了 ipadm show-addr 命令的输出，在完成第 58 页中的 “配置 IPv6 路由器”过程之后将看到这类输出。

```
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      172.16.15.232/24
net1/v4       static     ok      172.16.16.220/24
net0/v6       addrconf   ok      fe80::203:baff:fe11:b115/10
lo0/v6        static     ok      ::1/128
net0/v6a      static     ok      2001:db8:3c4d:15:203:baff:fe11:b115/64
net1/v6       addrconf   ok      fe80::203:baff:fe11:b116/10
net1/v6a      static     ok      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。包含地址对象名称（如 *interface*/v6）的项显示该接口的链路本地地址。包含地址对象名称（如 *interface*/v6）的项显示全局 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。请注意，标识 v6add 是随机定义的字符串。您可以定义其他字符串以构成地址对象名称的第二部分，只要 *interface* 反映将在其上创建 IPv6 地址的接口，例如 net0/mystring、net0/ipv6addr 等。

另请参见

- 要从 IPv6 网络拓扑中标识的路由器配置任何隧道，请参阅第 98 页中的 “使用 dladm 命令进行隧道配置和管理”。
- 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 60 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 64 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 115 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。在大多数情况下，您应当对启用了 IPv6 的接口使用地址自动配置。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。

您需要按以下顺序执行三个常规任务：

1. 关闭 IPv6 地址自动配置。请参见第 57 页中的 “如何关闭 IPv6 地址自动配置”。
2. 为主机创建临时地址。请参见第 61 页中的 “如何配置临时地址”。
3. 为接口 ID 配置 IPv6 标记。请参见第 63 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" (`http://www.ietf.org/rfc/rfc3041.txt?number=3041`)（IPv6 中用于无状态地址自动配置的专用扩展）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 **如有必要，请在主机的接口上启用 IPv6。**

请参阅第 55 页中的 “如何针对 IPv6 配置系统”。

2 **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`**

3 **（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。

4 **（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 87 页中的 “管理缺省地址选择”。

---

5 **（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。

6 **更改 `in.ndpd` 守护进程的配置。**

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

7 **通过运行 `ipadm show-addr` 命令来验证是否创建了临时地址，如示例 4–4 中所示。**

该命令输出显示临时地址的 `CURRENT` 字段中的 `t` 标志。

示例 4–3 `/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 4–4 显示启用了临时地址的 `ipadm show-addr` 命令输出

此示例显示了创建临时地址之后 `ipadm show-addr` 命令的输出。请注意，样例输出中仅包括 IPv6 相关信息。

```
# ipadm show-addr -o all
ADDROBJ    TYPE     STATE CURRENT PERSISTENT ADDR
lo0/v6     static   ok    U----   ---        ::1/128
net0/v6    addrconf ok    U----   ---        fe80::a00:20ff:feb9:4c54/10
net0/v6a   static   ok    U----   ---        2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
net0/?     addrconf ok    U--t-   ---        2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

请注意，对于地址对象 `net0/?`，`t` 标志在 `CURRENT` 字段下设置。该标志指示对应地址具有临时接口 ID。

另请参见

- 要为 IPv6 地址设置名称服务支持，请参见第 65 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 63 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 5 章，管理 TCP/IP 网络。

配置 IPv6 标记

IPv6 地址的 64 位接口 ID 也称为**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 确认要为其配置标记的接口存在并且未在该接口上配置 IPv6 地址。

注 – 确保该接口未配置 IPv6 地址。

```
# ipadm show-if
IFNAME   CLASS      STATE   ACTIVE   OVER
lo0      loopback   ok       yes     ---
net0     ip         ok       yes     ---

# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
```

此输出显示存在未配置 IPv6 地址的网络接口 `net0`。

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，标记格式如下：*xxxx*:*xxxx*:*xxxx*:*xxxx*。

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 `ipadm` 命令：

```
# ipadm create-addr -T addrconf -i interface-ID interface
```

例如，可使用以下命令配置接口 `net0` 的标记：

```
# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
```

注 – 为地址对象创建标记之后，无法再修改该标记。

---

4 使用所做更改更新 IPv6 守护进程。

```
# pkill -HUP in.ndpd
```

示例 4–5 在 IPv6 接口上配置用户指定的标记

以下示例显示为 net0 配置了 IPv6 地址和标记。

```
# ipadm show-if
IFNAME   CLASS      STATE    ACTIVE     OVER
lo0      loopback   ok       yes        ---
net0     ip         ok       yes        ---

# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8

# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
# pkill -HUP in.ndpd
# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v6       static     ok      ::1/128
net0/v6      addrconf   ok      fe80::1a:2b:3c:4d/10
net0/v6a     addrconf   ok      2002:a08:39f0:1:1a:2b:3c:4d/64
```

配置标记之后，地址对象 net0/v6 同时具有链路本地地址和为其接口 ID 配置的地址 1a:2b:3c:4d。请注意，创建 net0/v6 之后，无法再针对该接口修改此标记。

另请参见

- 要使用服务器的 IPv6 地址更新名称服务，请参见第 65 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 5 章，管理 TCP/IP 网络。

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

此过程提供在网络服务器上启用 IPv6 的常规步骤。某些步骤根据要实现 IPv6 的方式的不同可能有所不同。

1. **在服务器的 IP 接口上启用 IPv6。**

   有关过程，请参阅第 55 页中的 “配置 IPv6 接口”。

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 58 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 126 页中的 “自动配置过程”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 61 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 63 页中的 “如何配置用户指定的 IPv6 标记”。

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 的完整详细信息，请参阅《在 Oracle Solaris 11.1 中使用命名和目录服务》。

▼ 如何向 DNS 中添加 IPv6 地址

1. **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*

2. **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《在 Oracle Solaris 11.1 中使用命名和目录服务》。

示例 4–6 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.0.6.5.2.9.0.0.0.0.0.0.0.2.0.0.0 \
    IN        PTR        vallejo.Eng.apex.COM.
```

▼ 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 4–7 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 4–8 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 .***byname*

  此时会显示有关指定 *hostname* 的信息。

◆ ◆ ◆ 第 5 章

管理 TCP/IP 网络

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《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“监视 IP 接口和地址”。

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，请参阅以下各章以获取更多信息：

- 要计划 IPv6 实现，请参阅第 2 章，使用 IPv6 地址的注意事项。
- 要配置 IPv6 和创建双栈网络环境，请参阅第 4 章，在网络上启用 IPv6。

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
按协议显示统计信息。	监视特定系统上网络协议的性能。	第 70 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 74 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 73 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 75 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 和 traceroute 命令的输出。创建名为 inet_type 文件。在该文件中设置 DEFAULT_IP 变量。	第 78 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 84 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 81 页中的 “如何跟踪所有路由”

注 – 要监视网络接口，请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“监视 IP 接口和地址”。

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 5–1 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 5–2 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 5–3 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 5–4 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 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
net0  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
net0  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 5–5 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 5–6 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 5–7 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 5–8 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  net0
10.0.0.14            myhost               U         1      0  net0
default              distantrouter        UG        1      2  net0
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      net0:1
fe80::/10              fe80::1a2b:3c4d:5e6f:12a2           U     1   23     net0
ff00::/8               fe80::1a2b:3c4d:5e6f:12a2           U     1   0      net0
default                fe80::1a2b:3c4d:5e6f:12a2           UG    1   0      net0
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 5–9 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
```

包丢失统计信息显示主机是否丢弃了包。如果 `ping` 失败，检查 `ipadm` 和 `netstat` 命令报告的网络状态。请参阅《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》中的“监视 IP 接口和地址”和第 70 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` 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 – `netstat` 命令中的 `-f` 标志将覆盖 `inet_type` 文件中设置的值。

示例 5–10 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：

```
% ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      10.46.86.54/24
lo0/v6        static     ok      ::1/128
net0/v6       addrconf   ok      fe80::a00:fe73:56a8/10
net0/v6add    static     ok      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：

```
% ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      10.46.86.54/24
```

▼ 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。

- **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 5–11 `in.routed` 守护进程的网络日志

以下示例显示由第 79 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 net0 #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  net0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  net0  <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. **启动对 in.ndpd 守护进程的跟踪。**

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

2. **根据需要按 Ctrl-C 组合键终止跟踪。**

示例 5–12 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 net0
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 net0
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 5–13 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 5–14 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. **列显有关连接到系统的接口的信息。**

   ```
   # ipadm show-if
   ```

   snoop 命令通常使用第一个非回送设备，通常为主网络接口。

2. **键入不带参数的 snoop 开始捕获包，如示例 5–15 所示。**

3. **使用 Ctrl-C 组合键停止此进程。**

示例 5–15 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/net (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)
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 **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。

2 **检查 snoop 输出捕获文件。**

```
# snoop -i filename
```

示例 5–16 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 **键入带有选项的 snoop 并将输出保存到文件。**

3 **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 (http://www.ietf.org/rfc/rfc1761.txt?number=1761)（RFC 1761，Snoop 版本 2 包捕获文件格式）。

▼ 如何监视 IPv6 网络通信

您可以使用 snoop 命令来仅显示 IPv6 包。

- **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 5–17 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
```

使用 IP 层设备监视包

Oracle Solaris 中引入了用于增强 IP 观察功能的 IP 层设备。通过这些设备，可以访问具有与系统网络接口关联的地址的所有包。这些地址包括本地地址以及位于非回送接口或逻辑接口上的地址。可观察的通信流量可以是 IPv4 和 IPv6 地址。因此，可以监视以系统为目标的所有通信流量。通信流量可以是回送 IP 通信流量、来自远程计算机的包、要从系统发送的包或转发的所有通信流量。

使用 IP 层设备，全局区域的管理员可以监视区域之间以及区域内的通信流量。非全局区域的管理员也可以观察由该区域发送和接收的通信流量。

要监视 IP 层上的通信流量，请将一个新选项 -I 添加到 snoop 命令。此选项对该命令指定将使用新 IP 层设备，而不是底层链路层设备以显示通信流量数据。

▼ 如何检查 IP 层上的包

1. **如有必要，列显有关连接到系统的接口的信息。**

   ```
   # ipadm show-if
   ```

2. **捕获特定接口上的 IP 通信流量。**

   ```
   # snoop -I interface [-V | -v]
   ```

检查包的示例

所有示例都基于下列系统配置：

```
# ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      192.68.25.5/24
lo0/?         static     ok      127.0.0.1/8
net0/?        static     ok      172.0.0.3/24
net0/?        static     ok      172.0.0.1/24
lo0/?         static     ok      127.0.0.1/8
```

假设有两个区域 sandbox 和 toybox，它们使用以下 IP 地址：

- sandbox – 172.0.0.3
- toybox – 172.0.0.1

您可以对系统上的不同接口发出 snoop -I 命令。显示的包信息取决于您是全局区域的管理员还是非全局区域的管理员。

示例 5–18 回送接口上的通信流量

```
# snoop -I lo0
Using device ipnet/lo0 (promiscuous mode)
   localhost -> localhost    ICMP Echo request (ID: 5550 Sequence number: 0)
   localhost -> localhost    ICMP Echo reply (ID: 5550 Sequence number: 0)
```

要生成详细输出，请使用 -v 选项。

```
# snoop -v -I lo0
Using device ipnet/lo0 (promiscuous mode)
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 1 arrived at 10:40:33.68506
IPNET:  Packet size = 108 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 4
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 0
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
...
```

为了支持观察 IP 层上的包，引入了优先于要观察的包的新 ipnet 数据包头。将同时指示源和目标 ID。ID 为“0”指示将从全局区域生成通信流量。

示例 5–19 本地区域的 net0 设备中的包流

```
# snoop -I net0
Using device ipnet/net0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=62117 Syn Seq=195630514 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=62117 S=22 Syn Ack=195630515 Seq=195794440 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=62117 Ack=195794441 Seq=195630515 Len=0 Win=49152
sandbox -> toybox TCP D=62117 S=22 Push Ack=195630515 Seq=195794441 Len=20 Win=491
```

示例 5–19 本地区域的 net0 设备中的包流 （续）

该输出显示了系统内不同区域中发生的通信。您可以查看与 net0 IP 地址关联的所有包，包括从本地传递到其他区域的包。如果生成详细输出，则可以查看包流中涉及的区域。

```
# snoop -I net0 -v port 22
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 5 arrived at 15:16:50.85262
IPNET:  Packet size = 64 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 0
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 1
IPNET:
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:         .... ..0. = not ECN capable transport
IP:         .... ...0 = no ECN congestion experienced
IP:   Total length = 40 bytes
IP:   Identification = 22629
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 0000
IP:   Source address = 172.0.0.1, 172.0.0.1
IP:   Destination address = 172.0.0.3, 172.0.0.3
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 46919
TCP:  Destination port = 22
TCP:  Sequence number = 3295338550
TCP:  Acknowledgement number = 3295417957
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        0... .... = No ECN congestion window reduced
TCP:        .0.. .... = No ECN echo
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP         .... 0... = No push
TCP         .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 49152
```

示例 5–19 本地区域的 net0 设备中的包流 （续）

```
TCP:  Checksum = 0x0014
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
```

ipnet 数据包头指示该包是从全局区域 (ID 0) 发送至沙箱 (ID 1) 的。

示例 5–20 通过标识区域观察通信流量

```
# snoop -I hme0 sandboxsnoop -I net0 sandbox
Using device ipnet/hme0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=61658 Syn Seq=374055417 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=61658 S=22 Syn Ack=374055418 Seq=374124525 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=61658 Ack=374124526 Seq=374055418 Len=0 Win=49152
#
```

通过标识区域来观察包的功能在具有多个区域的系统中非常有用。当前，只能使用区域 ID 标识区域。不支持将 snoop 与区域名称一起使用。

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 **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
```

2 **备份缺省地址策略表的副本。**

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

3 **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。

4 **将已修改的策略表加载到内核。**

```
ipaddrsel -f /etc/inet/ipaddrsel.conf
```

5 **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 **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
```

2 **根据需要在 *filename* 中编辑策略表。**

3 **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 6 章

6 配置 IP 隧道

本章包含 IP 隧道的说明以及在 Oracle Solaris 中配置和维护隧道的过程。

IP 隧道概述

当中间网络不支持域中的协议时，IP 隧道提供了一种在这些域之间传输数据包的方式。例如，由于引入了 IPv6 协议，在大多数网络使用 IPv4 协议的环境中，IPv6 网络需要一种方式来与其边界外部通信。使用隧道使通信变得可能。IP 隧道在可使用 IP 访问的两个节点之间提供了一个虚拟链路。因此，可使用该链路在 IPv4 网络上传输 IPv6 包，从而在两个 IPv6 站点之间实现 IPv6 通信。

Oracle Solaris 11 中的 IP 隧道管理

在此 Oracle Solaris 发行版中，隧道管理已修订为与网络数据链路管理的新模型一致。现在，将使用新的 `dladm` 子命令创建和配置隧道。隧道现在还可以使用新管理模型的其他数据链路功能。例如，通过支持以管理方式选择的名称，可以为隧道指定有意义的名称。有关 `dladm` 子命令的更多信息，请参见 `dladm(1M)` 手册页。

隧道类型

隧道连接涉及将 IP 包封装到其他包中。通过此封装操作，包可以通过不支持包协议的中间网络到达其目标。

根据包封装的类型，隧道会有所不同。Oracle Solaris 支持以下类型的隧道：

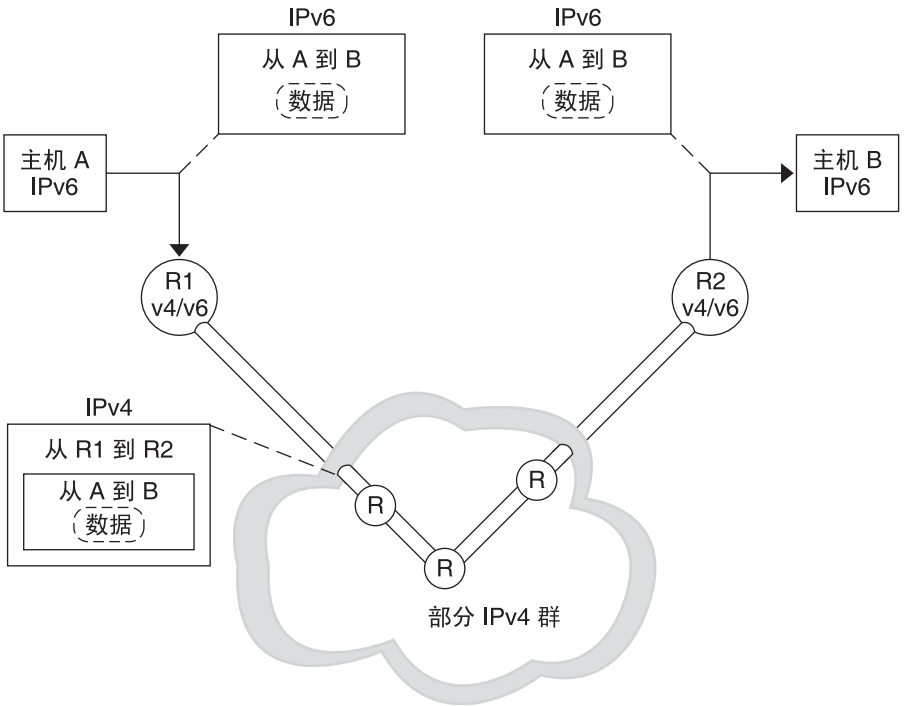
- *IPv4* **隧道**－IPv4 或 IPv6 包将封装在 IPv4 数据包头中并发送到预配置的单播 IPv4 目标。为了更具体地说明流经隧道的包，IPv4 隧道也称为 *IPv4 over IPv4* **隧道**或 *IPv6 over IPv4* **隧道**。

- *IPv6* **隧道**－IPv4 或 IPv6 包将封装在 IPv6 数据包头中并发送到预配置的单播 IPv6 目标。为了更具体地说明流经隧道的包，IPv6 隧道也称为 *IPv4 over IPv6* **隧道**或 *IPv6 over IPv6* **隧道**。
- *6to4* **隧道**－IPv6 包将封装在 IPv4 数据包头中并发送到自动按包确定的 IPv4 目标。确定的依据基于 6to4 协议中定义的算法。

IPv6 和 IPv4 的组合网络环境中的隧道

具有 IPv6 域的大多数站点通过遍历 IPv4 网络与其他 IPv6 域通信，这比仅 IPv6 网络更普遍。下图说明两个 IPv6 主机之间通过 IPv4 路由器（在该图中用 "R" 指示）的隧道连接机制。

图 6–1 IPv6 隧道连接机制



在图中，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。

IPv6 包封装于 IPv4 包内。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。包通过隧道传输到目标边界路由器，将在该路由器中对包取消封装，然后，路由器将单独的 IPv6 包转发到目标节点。

6to4 隧道

Oracle Solaris 包括 6to4 隧道作为首选中间方法，以实现从 IPv4 到 IPv6 寻址的转换。通过 6to4 隧道，隔离的 IPv6 站点可以通过不支持 IPv6 的 IPv4 网络跨自动隧道进行通信。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 主题的参考信息：

- 6to4 隧道的拓扑
- 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

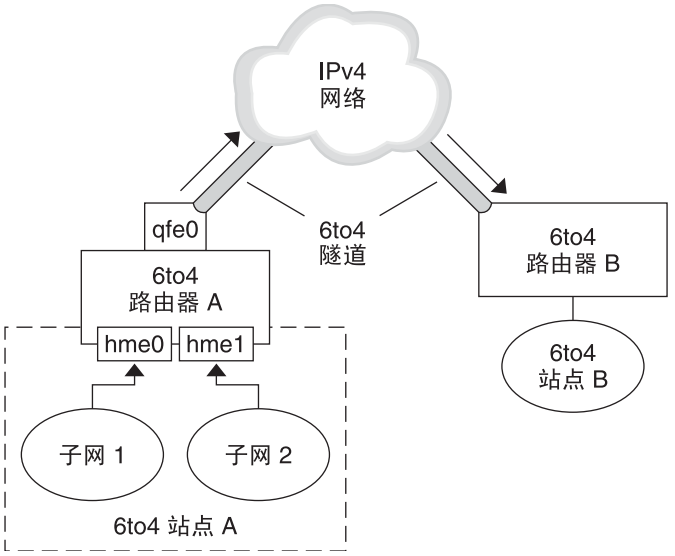
下表介绍用来配置 6to4 隧道的额外任务，以及获得额外有用信息的资源。

任务或详细信息	参考
用来配置 6to4 隧道的任务	第 102 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" (`http://www.ietf.org/rfc/rfc3056.txt`)（RFC 3056，通过 IPv4 云连接 IPv6 域）
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)`
6to4 安全问题	Security Considerations for 6to4 (`http://www.ietf.org/rfc/rfc3964.txt`)（6to4 安全注意事项）

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 6–2 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 6–2 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。

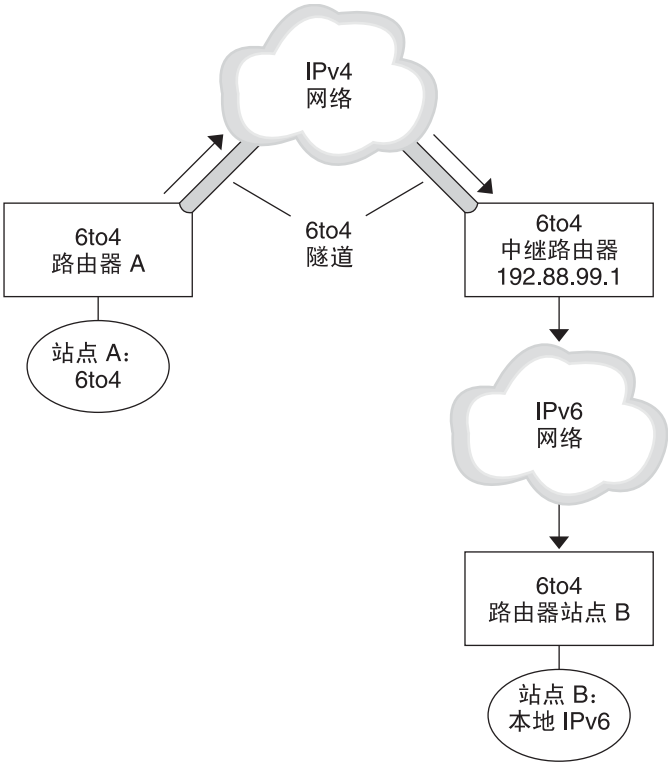
1. 6to4 站点 A 的子网 1 上的主机发送传输请求，6to4 站点 B 上的主机作为目标。每个数据包头中均有 6to4 派生的源地址和 6to4 派生的目标地址。

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 6–3 从 6to4 站点到 6to4 中继路由器的隧道



在图 6–3 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 6–3 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。

部署隧道

要正确部署 IP 隧道，需要执行两个主要任务。首先，创建隧道链路。然后，在隧道上配置 IP 接口。本节简要介绍了创建隧道及其对应 IP 接口的要求。

创建隧道的要求

要成功创建隧道，必须遵守以下要求：

- 如果使用主机名而不是字面值 IP 地址，则这些名称必须解析为与隧道类型兼容的有效 IP 地址。
- 所创建的 IPv4 或 IPv6 隧道不得与配置的其他隧道具有相同的隧道源地址和隧道目标地址。
- 所创建的 IPv4 或 IPv6 隧道不得与现有的 6to4 隧道具有相同的隧道源地址。
- 如果创建 6to4 隧道，则该隧道不得与配置的其他隧道具有相同的隧道源地址。

有关在网络中建立隧道的信息，请参阅第 28 页中的 “网络中使用隧道的规划”。

隧道和 IP 接口的要求

每个隧道类型都对隧道上配置的 IP 接口具有特定的 IP 地址要求。下表中汇总了这些要求。

表 6–1 隧道和 IP 接口要求

隧道类型	隧道上允许的 IP 接口	IP 接口要求
IPv4 隧道	IPv4 接口	手动指定本地和远程地址。

表 6–1 隧道和 IP 接口要求 （续）

隧道类型	隧道上允许的 IP 接口	IP 接口要求
	IPv6 接口	发出 ipadm create-addr -T addrconf 命令时，将自动设置本地和远程本地链路地址。有关详细信息，请参见 ipadm(1M) 手册页。
IPv6 隧道	IPv4 接口	手动指定本地和远程地址。
	IPv6 接口	发出 ipadm create-addr -T addrconf 命令时，将自动设置本地和远程本地链路地址。有关详细信息，请参见 ipadm(1M) 手册页。
6to4 隧道	仅限 IPv6 接口	发出 ipadm create-ip 命令时，将自动选择缺省的 IPv6 地址。有关详细信息，请参见 ipadm(1M) 手册页。

您可以通过为 ipadm 命令指定其他 IPv6 地址来覆盖 6to4 隧道的缺省 IPv6 接口地址。

同样，要覆盖自动为 IPv4 或 IPv6 隧道上的 IPv6 接口设置的本地链路地址，可以在隧道的主机文件中指定其他源和目标地址。

使用 dladm 命令进行隧道配置和管理

本节介绍使用 dladm 命令配置隧道的过程。

dladm 子命令

从此 Oracle Solaris 发行版开始，隧道管理将与 IP 接口配置分开。IP 隧道的数据链路方面现在使用 dladm 命令管理。此外，IP 接口配置（包括 IP 隧道接口）现在使用 ipadm 命令执行。

将使用 dladm 的下列子命令配置 IP 隧道：

- create-iptun
- modify-iptun
- show-iptun
- delete-iptun
- set-linkprop

有关 dladm 命令的详细信息，请参阅 dladm(1M) 手册页。

注 – IP 隧道管理与 IPsec 配置密切相关。例如，IPsec 虚拟专用网络 (virtual private network, VPN) 是主要使用 IP 隧道的网络之一。有关 Oracle Solaris 中的安全性的更多信息，请参见《在 Oracle Solaris 11.1 中保护网络安全》中的第 6 章 “IP 安全体系结构（概述）”。要配置 IPsec，请参见《在 Oracle Solaris 11.1 中保护网络安全》中的第 7 章 “配置 IPsec（任务）”。

配置隧道（任务列表）

任务	说明	参考
创建 IP 隧道。	配置要用于网络之间通信的隧道。	第 99 页中的 “如何创建和配置 IP 隧道”
修改隧道的配置。	更改隧道的原始参数，例如隧道的源地址或目标地址。	第 105 页中的 “如何修改 IP 隧道配置”
显示隧道配置。	显示特定隧道或系统的所有 IP 隧道的配置信息。	第 106 页中的 “如何显示 IP 隧道的配置”
删除隧道。	删除隧道配置。	第 108 页中的 “如何删除 IP 隧道”

▼ 如何创建和配置 IP 隧道

1 创建隧道。

```
# dladm create-iptun [-t] -T type -a [local|remote]=addr,... tunnel-link
```

此命令可使用以下选项或参数：

`-t`

创建临时隧道。缺省情况下，该命令将创建一个持久性隧道。

注 – 如果要在隧道中配置一个持久性 IP 接口，则必须创建一个持久性隧道并且不使用 `-t` 选项。

`-T` *type*

指定要创建的隧道的类型。创建所有隧道类型都需要此参数。

`-a [local|remote]=`*address,...*

指定对应于本地地址和远程隧道地址的字面值 IP 地址或主机名。这些地址必须有效并且已在系统中创建。取决于隧道的类型，仅指定一个地址或同时指定本地和远程地址。如果指定本地和远程地址，则必须使用逗号分隔这两个地址。

- IPv4 隧道需要有本地和远程 IPv4 地址才能正常工作。
- IPv6 隧道需要有本地和远程 IPv6 地址才能正常工作。
- 6to4 隧道需要有本地 IPv4 地址才能正常工作。

注 – 对于持久性 IP 隧道数据链路配置，如果对地址使用主机名，这些主机名将保存在配置存储中。在后续系统引导期间，如果名称解析到的 IP 地址不同于创建隧道时使用的 IP 地址，则隧道将获取新配置。

tunnel-link

指定 IP 隧道链路。如果网络链路管理中支持有意义的名称，则隧道名称不再限制为要创建的隧道类型。而是，可以为隧道指定任何通过管理方式选择的名称。隧道名称由字符串和物理连接点 (physical point of attachment, PPA) 编号组成，例如 *mytunnel0*。有关指定有意义名称的管理规则，请参阅《Oracle Solaris 11 联网介绍》中的“有效链路名称的规则”。

如果未指定隧道链路，则会根据以下命名约定自动提供该名称：

- 对于 IPv4 隧道： `ip.tun#`
- 对于 IPv6 隧道： `ip6.tun#`
- 对于 6to4 隧道： `ip.6to4tun#`

是所要创建的隧道类型的最低可用 PPA 编号。

2 （可选）设置跃点限制或封装限制的值。

```
# dladm set-linkprop -p [hoplimit=value] [encaplimit=value] tunnel-link
```

hoplimit　指定 IPv6 上隧道的隧道接口的跃点限制。*hoplimit* 与 IPv4 上隧道的 IPv4 生存时间 (time to live, TTL) 字段等效。

encaplimit　指定包允许的嵌套隧道的级数。此选项仅适用于 IPv6 隧道。

指定包允许的嵌套隧道的级数。此选项仅适用于 IPv6 隧道。

注 – 为 `hoplimit` 和 `encaplimit` 设置的值必须处于可接受的范围内。`hoplimit` 和 `encaplimit` 为隧道链路属性。因此，与其他链路属性一样，这些属性通过相同的 `dladm` 子命令管理。这些子命令为 `dladm set-linkprop`、`dladm reset-linkprop` 和 `dladm show-linkprop`。有关与 `dladm` 命令配合使用以管理链路的其他子命令，请参阅 `dladm(1M)` 手册页。

3 **创建隧道上的 IP 接口。**

```
# ipadm create-ip tunnel-interface
```

其中，*tunnel-interface* 使用与隧道链路相同的名称。

4 **为隧道接口指定本地和远程 IP 地址。**

```
# ipadm create-addr [-t] -a local=address,remote=address  interface
```

`-t`	指示临时 IP 配置而不是隧道上的持久性 IP 配置。如果不使用此选项，则 IP 接口配置为持久性配置。
`-a local=`*address*`,remote=`*address*	指定隧道接口的 IP 地址。需要同时指定源和目标 IP 地址，分别由 `local` 和 `remote` 表示。本地和远程地址可以是 IPv4 或 IPv6 地址。
interface	指定隧道接口。

有关 `ipadm` 命令和配置 IP 接口(包括隧道接口)的不同选项的更多信息，请参见 `ipadm`(1M) 手册页和《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》 。

5 **将隧道配置信息添加到 `/etc/hosts` 文件。**

6 **（可选）验证隧道的 IP 接口配置的状态。**

```
# ipadm show-addr interface
```

示例 6–1 在 IPv4 隧道上创建 IPv6 接口

以下示例说明了如何创建持久性的 IPv6 over IPv4 隧道。

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 private0
# dladm set-linkprop -p hoplimit=200 private0
# ipadm create-ip private0
# ipadm create-addr -T addrconf private0
# ipadm show-addr private/
ADDROBJ       TYPE     STATE   ADDR
private0/v6   static   ok      fe80::a08:392e/10 --> fe80::8191:9a56
```

要添加备用地址，请使用相同语法。例如，可以按如下所示添加全局地址：

```
# ipadm create-addr -a local=2001:db8:4728::1, \
remote=2001:db8:4728::2 private0
# ipadm show-addr private0/
ADDROBJ          TYPE       STATE   ADDR
private0/v6      addrconf   ok      fe80::a08:392e/10 --> fe80::8191:9a56
private0/v6a     static     ok      2001:db8:4728::1 --> 2001:db8:4728::2
```

请注意，IPv6 地址的前缀 `2001:db8` 为特殊的 IPv6 前缀，专门用于文档示例。

示例 6–2 在 IPv4 隧道上创建 IPv4 接口

以下示例说明了如何创建持久性的 IPv4 over IPv4 隧道。

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 vpn0
# ipadm create-ip vpn0
# ipadm create-addr -a local=10.0.0.1,remote=10.0.0.2 vpn0
# ipadm show-addr
ADDROBJ    TYPE     STATE   ADDR
lo0/v4     static   ok      127.0.0.1
vpn0/v4    static   ok      10.0.0.1-->10.0.0.2
```

您可以进一步配置 IPsec 策略来为流经此隧道的包提供安全连接。有关 IPsec 配置的信息，请参见《在 Oracle Solaris 11.1 中保护网络安全》中的第 7 章 “配置 IPsec（任务）”。

示例 6–3 在 IPv6 隧道上创建 IPv6 接口

以下示例说明了如何创建持久性的 IPv6 over IPv6 隧道。

```
# dladm create-iptun -T ipv6 -a local=2001:db8:feed::1234,remote=2001:db8:beef::4321 \
tun0
# ipadm create-ip tun0
# ipadm create-addr -T addrconf tun0
# ipadm show-addr
ADDROBJ    TYPE        STATE   ADDR
lo0/v6     static      ok      ::1/128
tun0/v6    addrconf    ok      2001:db8:feed::1234 --> 2001:db8:beef::4321
```

要添加地址，如全局地址或替代本地和远程地址，请按如下所示使用 ipadm 命令：

```
# ipadm create-addr \
-a local=2001:db8::4728:56bc,remote=2001:db8::1428:57ab tun0
# ipadm show-addr tun0
ADDROBJ     TYPE       STATE ADDR
tun0/v6     addrconf   ok    2001:db8:feed::1234 --> 2001:db8:beef::4321
tun0/v6a    static     ok    2001:db8::4728:56bc --> 2001:db8::1428:57ab
```

▼ 如何配置 6to4 隧道

在 6to4 隧道中，6to4 路由器必须充当网络的 6to4 站点中节点的 IPv6 路由器。因此，当配置 6to4 路由器时，还必须在其物理接口上将该路由器配置为 IPv6 路由器。有关 IPv6 路由的更多信息，请参见第 130 页中的 “IPv6 路由”。

1 创建 6to4 隧道。

```
# dladm create-iptun -T 6to4 -a local=address  tunnel-link
```

此命令可使用以下选项或参数：

-a local=*address*	指定隧道本地地址，该地址必须是系统中已存在的有效地址。
tunnel-link	指定 IP 隧道链路。如果网络链路管理中支持有意义的名称，则隧道名称不再限制为要创建的隧道类型。相反，可以为隧道指定任何通过管理方式选择的名称。隧道名称由字符串和 PPA 编号组成，例如 *mytunnel0*。有关指定有意义名称的管理规则，请参阅《Oracle Solaris 11 联网介绍》中的“有效链路名称的规则”。

2 **创建隧道 IP 接口。**

```
# ipadm create-ip tunnel-interface
```

其中，*tunnel-interface* 使用与隧道链路相同的名称。

3 **（可选）为所使用的隧道添加替代 IPv6 地址。**

4 **编辑 /etc/inet/ndpd.conf 文件并添加以下两行以通告 6to4 路由：**

```
if subnet-interface AdvSendAdvertisements 1
IPv6-address subnet-interface
```

第一行指定接收通告的子网。*subnet-interface* 是指子网所连接到的链路。第二行中的 IPv6 地址必须具有 6to4 前缀 2000，该前缀用于 6to4 隧道中的 IPv6 地址。

有关 ndpd.conf 文件的详细信息，请参阅 ndpd.conf(4) 手册页。

5 **启用 IPv6 转发。**

```
# ipadm set-prop -p forwarding=on ipv6
```

6 **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。

7 **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 65 页中的 “针对 IPv6 配置名称服务支持”。

示例 6–4 创建 6to4 隧道

在此示例中，子网接口为 bge0，/etc/inet/ndpd.conf 将在相应步骤中引用该接口。

以下示例说明了如何创建 6to4 隧道。请注意，在 6to4 隧道上只能配置 IPv6 接口。

```
# dladm create-iptun -T 6to4 -a local=192.168.35.10 tun0
# ipadm create-ip tun0
# ipadm show-addr
ADDROBJ       TYPE     STATE   ADDR
lo0/v4        static   ok      127.0.0.1/8
net0/v4       static   ok      192.168.35.10/24
lo0/v6        static   ok      ::1/128
```

```
tun0/_a         static   ok      2002:c0a8:57bc::1/64

# ipadm create-addr -a 2002:c0a8:230a::2/16 tun0
# ipadm create-addr -a 2002:c0a8:230a::3/16 tun0
# ipadm show-addr tun0
ADDROBJ         TYPE     STATE   ADDR
lo0/v4          static   ok      127.0.0.1/8
net0/v4         static   ok      192.168.35.10/24
lo0/v6          static   ok      ::1/128
tun0/_a         static   ok      2002:c0a8:57bc::1/64
tun0/v6         static   ok      2002:c0a8:230a::2/16
tun0/v6a        static   ok      2002:c0a8:230a::3/16

# vi /etc/inet/ndpd.conf
if bge0 AdvSendAdvertisements 1
2002:c0a8:57bc::1/64 bge0

# ipadm set-prop -p forwarding=on ipv6
```

请注意，对于 6to4 隧道，IPv6 地址的前缀为 2002。

▼ 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《Troubleshooting Network Issues》中的"Security Issues When Tunneling to a 6to4 Relay Router"。

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 99 页中的 "如何创建和配置 IP 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 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)（RFC 3068，6to4 中继路由器的任播前缀）。

- 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。

2 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
```

3 （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 6–5 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
```

▼ 如何修改 IP 隧道配置

- **更改隧道的配置。**

```
# dladm modify-iptun -a [local|remote]=addr,... tunnel-link
```

无法修改现有隧道的类型。因此，不允许对此命令使用 -T *type* 选项。只能修改以下隧道参数：

-a [local|remote]=*address,...*

指定对应于本地地址和远程隧道地址的字面值 IP 地址或主机名。取决于隧道的类型，仅指定一个地址或同时指定本地和远程地址。如果指定本地和远程地址，则必须使用逗号分隔这两个地址。

- IPv4 隧道需要有本地和远程 IPv4 地址才能正常工作。
- IPv6 隧道需要有本地和远程 IPv6 地址才能正常工作。
- 6to4 隧道需要有本地 IPv4 地址才能正常工作。

对于持久性 IP 隧道数据链路配置，如果对地址使用主机名，这些主机名将保存在配置存储中。在后续系统引导期间，如果名称解析到的 IP 地址不同于创建隧道时使用的 IP 地址，则隧道将获取新配置。

如果要更改隧道的本地和远程地址，请确保这些地址与要修改的隧道类型一致。

注 – 如果要更改隧道链路的名称，不要使用 modify-iptun 子命令，应使用 dladm rename-link。

```
# dladm rename-link old-tunnel-link new-tunnel-link
```

类似地，不要使用 modify-iptun 命令更改隧道属性，例如 hoplimit 或 encaplimit，应使用 dladm set-linkprop 命令设置这些属性的值。

示例 6–6 修改隧道的地址和属性

此示例由两个过程组成。首先，临时更改 IPv4 隧道 vpn0 的本地和远程地址。以后重新引导系统时，隧道将恢复为使用原始地址。第二个过程将 vpn0 的 hoplimit 更改为 60。

```
# dladm modify-iptun -t -a local=10.8.48.149,remote=192.1.2.3 vpn0

# dladm set-linkprop -p hoplimit=60 vpn0
```

▼ 如何显示 IP 隧道的配置

- **显示 IP 隧道的配置。**

```
# dladm show-iptun [-p] -o fields [tunnel-link]
```

可以将以下选项与该命令配合使用：

`-p` 显示计算机可解析格式的信息。此参数是可选的。

`-o` *fields* 显示提供特定隧道信息的选定字段。

tunnel-link 指定要显示其配置信息的隧道。此参数是可选的。如果省略隧道名称，该命令将显示有关系统中所有隧道的信息。

示例 6–7 显示有关所有隧道的信息

在以下示例中，系统中仅存在一个隧道。

```
# dladm show-iptun
LINK     TYPE     FLAGS     LOCAL          REMOTE
tun0     6to4     --        192.168.35.10  --
vpn0     ipv4     --        10.8.48.149    192.1.2.3
```

示例 6–8 显示计算机可解析格式的选定字段

在以下示例中，仅显示包含隧道信息的特定字段。

```
# dladm show-iptun -p -o link,type,local
tun0:6to4:192.168.35.10
vpn0:ipv4:10.8.48.149
```

▼ 如何显示 IP 隧道的属性

- **显示隧道链路的属性。**

  # dladm show-linkprop [-c] [-o *fields***] [***tunnel-link***]**

  可以将以下选项与该命令配合使用：

  `-c` 显示计算机可解析格式的信息。此参数是可选的。

  `-o` *fields* 显示提供有关链路属性的特定信息的选定字段。

  tunnel-link 指定要显示其属性信息的隧道。此参数是可选的。如果省略隧道名称，该命令将显示有关系统中所有隧道的信息。

示例 6–9 显示隧道的属性

以下示例说明如何显示隧道的所有链路属性。

```
# dladm show-linkprop tun0
LINK     PROPERTY     PERM     VALUE     DEFAULT     POSSIBLE
tun0     autopush     --       --        --          --
```

```
tun0      zone          rw       --         --          --
tun0      state         r-       up         up          up,down
tun0      mtu           r-       65515      --          576-65495
tun0      maxbw         rw       --         --          --
tun0      cpus          rw       --         --          --
tun0      priority      rw       high       high        low,medium,high
tun0      hoplimit      rw       64         64          1-255
```

▼ 如何删除 IP 隧道

1 **根据接口类型，使用相应语法取消激活隧道上配置的 IP 接口。**

```
# ipadm delete-ip tunnel-link
```

注 – 要成功删除隧道，隧道上应检测不到 IP 接口。

2 **删除 IP 隧道。**

```
# dladm delete-iptun tunnel-link
```

此命令的唯一选项是 `-t`，将导致隧道被临时删除。重新引导系统后，将恢复该隧道。

示例 6–10 删除为 IPv6 接口配置的 IPv6 隧道

在以下示例中，永久删除了持久性隧道。

```
# ipadm delete-ip ip6.tun0
# dladm delete-iptun ip6.tun0
```

◆ ◆ ◆ 第 7 章

7 IPv4 参考信息

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，

本章包含以下信息：

TCP/IP 配置文件

在网络中，配置信息存储在控制网络运行方式的不同文件和数据库中。本节简要介绍了这些文件。当您对网络实现更改时，有些文件需要进行更新和维护。其他文件需要很少或不需要管理。

`/etc/defaultrouter`	此文件包含与网络直接相连的路由器的 IP 接口名称。系统中存在此文件是可选的。如果此文件存在，则系统配置为支持静态路由。
`/etc/inet/hosts`	此文件包含网络中的 IPv4 地址以及配置了这些地址的对应接口名称。如果您要使用 NIS、DNS 名称服务或 LDAP 目录服务，则主机信息存储在服务器中的其他数据库，例如 `hosts.byname`。有关更多信息，请参见《在 Oracle Solaris 11.1 中使用命名和目录服务》。
`/etc/inet/netmasks`	此文件包含网络号（如 `192.168.0.0`）以及该网络号的网络掩码信息（如 `255.255.255.0`）。在使用 NIS 或 LDAP 的网络中，此信息存储在服务器的网络掩码数据库中。有关更多信息，请参见 `netmasks(4)` 手册页。

/etc/bootparams	此文件包含的参数用于确定配置为以网络客户机模式引导的系统的引导过程。有关更多信息，请参见第 33 页中的 “设置系统配置模式”。此文件是创建名称服务使用的 bootparams 数据库的基础（如果您未使用本地文件模式）。要获取有关此文件的内容和格式的特定信息，请参阅 bootparams(4) 手册页。
/etc/ethers	此文件将主机名与 MAC 地址进行关联。此文件是创建要用于系统配置为网络客户机的网络的 ethers 数据库的基础。有关更多信息，请参见 ethers(4) 手册页。
/etc/inet/networks	此文件将网络名称与网络号进行关联。此外，也可以添加注释，以进一步阐明数据库中的每一项。此文件启用要使用的应用程序，并显示网络名称而非网络号。例如，netstat 程序使用此数据库中的信息生成状态表。通过路由器连接到本地网络的所有子网都必须包含在此文件中。有关更多信息，请参见 networks(4) 手册页。
/etc/inet/protocols	此文件列出了系统中安装的 TCP/IP 协议及其协议编号。此文件很少需要进行管理。有关更多信息，请参见 protocols(4) 手册页。
/etc/inet/services	此文件列出了 TCP 和 UDP 服务的名称及其已知端口号。该列表由调用网络服务的程序使用。通常，此文件不需要进行任何管理。有关更多信息，请参见 services(4) 手册页。

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《在 Oracle Solaris 11.1 中管理服务和故障》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 50 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 110 页中的 “inetd Internet 服务守护进程”。

name-service/switch SMF 服务

name-service/switch SMF 服务可定义配置信息网络数据库的搜索顺序。以前存储在配置文件中的某些网络配置信息（如缺省域）已转换成为此 SMF 服务的属性。此 SMF 服务的属性用于确定系统中名称服务的实现。这些属性列出如下：

```
% svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring             solaris.smf.value.name-service.switch
config/default              astring             files
config/password             astring             "files nis"
config/group                astring             "files nis"
config/host                 astring             "files dns nis"
config/network              astring             "nis [NOTFOUND=return] files"
config/protocol             astring             "nis [NOTFOUND=return] files"
config/rpc                  astring             "nis [NOTFOUND=return] files"
config/ether                astring             "nis [NOTFOUND=return] files"
config/netmask              astring             "files nis"
config/bootparam            astring             "nis [NOTFOUND=return] files"
config/publickey            astring             "nis [NOTFOUND=return] files"
config/netgroup             astring             nis
config/automount            astring             "files nis"
config/alias                astring             "files nis"
config/service              astring             "files nis"
config/printer              astring             "user nis"
config/auth_attr            astring             "files nis"
config/prof_attr            astring             "files nis"
config/project              astring             "files nis"
```

为每个属性设置的值可确定要搜索将影响网络用户的信息（如口令、别名或网络掩码）的名称服务。在此示例中，automount 和 password 属性设置为 files 和 nis。因此，自动挂载信息和口令信息分别通过文件和 NIS 服务获取。

如果您要从一种名称服务更改为另一种名称服务，必须为 name-service/switch SMF 服务设置相应的属性以启用选定的名称服务。

例如，假定您要在网络中使用 LDAP 命名服务。SMF 服务的以下属性需要进行配置：

- config/default 需要设置为使用文件和 LDAP。
- config/host 需要设置为使用文件和 DNS。
- config/netgroup 需要设置为使用 LDAP。
- config/printer 需要设置为使用用户、文件和 LDAP。

因此，您需要键入以下命令以正确设置这些属性。

```
# svccfg -s name-service/switch setprop config/default = astring: '"files ldap"'
# svccfg -s name-service/switch setprop config/host = astring: '"files dns"'
# svccfg -s name-service/switch setprop config/netgroup = astring: '"ldap"'
# svccfg -s name-service/switch setprop config/printer = astring: '"user files ldap"'
# svccfg -s name-service/switch:default refresh
```

有关名称服务切换的完整详细信息，请参阅《在 Oracle Solaris 11.1 中使用命名和目录服务》。

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。

网络数据库的使用方式如下：

- 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 使用包含主机信息的记录。

注 – DNS 引导文件和数据文件不直接对应于网络数据库。

有关网络数据库在 NIS、DNS 和 LDAP 中的对应项的信息，请参阅《在 Oracle Solaris 11.1 中使用命名和目录服务》。

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 7–1 和表 7–2。

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` 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。

Oracle Solaris 中的路由协议表

下表列出了 Oracle Solaris 支持的所有路由协议

表 7–1 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 38 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 45 页中的 “如何在单接口主机上启用静态路由”和第 46 页中的 “如何在单接口系统上启用动态路由”
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 58 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 55 页中的 “配置 IPv6 接口”

下表列出了 Oracle Solaris 同时支持的 Open Source Quagga 路由协议套件。

表 7–2 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。

◆ ◆ ◆ 第 8 章

IPv6 参考信息

本章包含以下有关 Oracle Solaris IPv6 实现的参考信息。

有关配置启用 IPv6 的网络的任务，请参阅第 4 章，在网络上启用 IPv6。有关 IP 隧道的所有信息，请参阅第 6 章，配置 IP 隧道。

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：

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 8–1 /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) (http://www.ietf.org/rfc/rfc2461.txt?number=2461)（RFC 2461，IP 版本 6 (IPv6) 的相邻节点搜索）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 8–2 /etc/inet/ndpd.conf 接口配置变量

变量	缺省值	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 58 页中的 “如何配置启用了 IPv6 的路由器”。

表 8–2 /etc/inet/ndpd.conf 接口配置变量 （续）

变量	缺省值	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 63 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 61 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 61 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 61 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 61 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 8–3 /etc/inet/ndpd.conf 前缀配置变量

变量	缺省值	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。
AdvValidLifetime	2592000 秒	指定所配置的前缀的有效生命周期。

示例 8–1 /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
```

/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 8–4 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 87 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 119 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 8–4 IPv6 地址选择策略表

前缀	优先级	定义
::1/128	50	回送
::/0	40	缺省值
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，您可以使用 `ipadm` 将这些地址标记为首选地址。有关 `ipadm` 命令的更多信息，请参阅 `ipadm(1M)` 手册页。
- 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 95 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 99 页中的 “如何创建和配置 IP 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 8–2 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 8–3 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 8–4 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 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 74 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 84 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 77 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 80 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。

-a	关闭无状态和有状态地址自动配置。
-d	启用调试功能。
-f *config-file*	指定要从中读取配置的文件，而不使用缺省的 /etc/inet/ndpd.conf 文件。
-t	为所有传出和传入包启用包跟踪功能。

in.ndpd 守护进程由 /etc/inet/ndpd.conf 配置文件中设置的参数和 /var/inet/ndpd_state. *interface* 启动文件中任何适用的参数来控制。

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 8–1 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* 指定用于发送或接收 RIPng 包的 UDP 端口号。

-P　禁止使用毒性逆转 (poison reverse)。

-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。

-t　将所有发送和接收的包输出到标准输出。

-v　将对路由表的所有更改（包括时间戳）输出到标准输出。

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《在 Oracle Solaris 11.1 中管理服务和故障》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 50 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《Programming Interfaces Guide》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) (http://www.ietf.org/rfc/rfc2461.txt?number=2461)（RFC 2461，IP 版本 6 (IPv6) 的相邻节点搜索）中所述。

本节讨论相邻节点搜索协议的以下功能：

- 第 126 页中的 “相邻节点搜索功能中的 ICMP 消息”
- 第 126 页中的 “自动配置过程”
- 第 128 页中的 “相邻节点请求和不可访问性”
- 第 128 页中的 “重复地址检测算法”
- 第 129 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。

地址的唯一性

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。

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`。

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)。

索引

N

P

Q

R

S

T

U

V

安

包

本

边

地

丢

动

多

反

服

负

故

互

接

静

空

链

临

路

名

配

前

区

缺