

# 在 Oracle® Solaris 11.1 中使用固定网络配置连接系统

版权所有 © 2011, 2012, Oracle 和/或其附属公司。保留所有权利。

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

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

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

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

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

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

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

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

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

# 目录

---

前言 .....	7
<b>1 固定网络配置概述 .....</b>	<b>9</b>
什么是固定网络配置? .....	9
配置文件管理网络配置的亮点 .....	10
网络配置工具 .....	11
dladm 命令 .....	12
ipadm 命令 .....	13
<b>2 配置系统网络 .....</b>	<b>15</b>
配置网络 (任务列表) .....	15
▼ SPARC: 如何确保每个接口的 MAC 地址是唯一的 .....	16
▼ 如何更改系统上的活动 NCP .....	17
▼ 如何配置 IP 接口 .....	18
其他网络配置和管理任务 .....	22
<b>3 使用数据链路 .....</b>	<b>23</b>
基本 dladm 命令 .....	23
显示有关数据链路的常规信息 (dladm) .....	23
显示系统的数据链路 (dladm show-link) .....	24
显示数据链路的物理属性 (dladm show-phys) .....	24
删除数据链路 (dladm delete-phys) .....	25
重命名数据链路 (dladm rename-link) .....	25
定制数据链路属性 .....	25
数据链路属性概述 .....	26
启用巨型帧支持 .....	26
修改链路速度参数 .....	27

设置数据链路上的 STREAMS 模块 .....	27
将 e1000g 驱动程序设置为使用直接内存访问绑定 .....	28
手动设置中断率 .....	29
获取有关数据链路属性的状态信息 .....	29
使用 dladm 命令的其他配置任务 .....	30
▼如何在系统中切换主接口 .....	31
▼如何使用动态重新配置更换网络接口卡 .....	32
<b>4 使用 IP 接口 .....</b>	<b>35</b>
基本 ipadm 命令 .....	35
删除 IP 接口配置 (ipadm delete-ip) .....	35
禁用 IP 接口配置 (ipadm disable-ip) .....	36
删除接口的地址 (ipadm delete-addr) .....	36
设置 IP 接口属性 .....	37
启用包转发 .....	37
设置 IP 地址属性 .....	38
设置 TCP/IP 协议属性 .....	39
全局启用包转发 .....	39
设置特权端口 .....	40
在多宿主主机上实现对称路由 .....	41
实施通信拥塞控制 .....	42
更改 TCP 接收缓冲区大小 .....	43
监视 IP 接口和地址 .....	44
获取有关 IP 接口的常规信息 .....	45
获取有关 IP 接口的信息 .....	45
获取有关 IP 接口属性的信息 .....	46
获取有关 IP 地址的信息 .....	47
获取有关 IP 地址属性的信息 .....	48
<b>5 在运行 Oracle Solaris 的手提电脑上配置无线联网 .....</b>	<b>49</b>
WiFi 通信任务列表 .....	49
▼如何连接到 WiFi 网络 .....	50
▼如何监视 WiFi 链路 .....	53
安全 WiFi 通信 .....	54
▼如何设置加密的 WiFi 网络连接 .....	55

**A** 对应关系比较：ifconfig 和 ipadm 命令 ..... 57

**B** 对应关系比较：nfd 和 ipadm 命令 ..... 59

索引 ..... 63



# 前言

---

欢迎阅读《在 Oracle Solaris 11.1 中使用固定网络配置连接系统》。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。

---

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

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

---

## 目标读者

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

## 获取 Oracle 支持

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

## 印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

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

## 命令中的 shell 提示符示例

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

表 P-2 shell 提示符

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

# 固定网络配置概述

---

在运行 Oracle Solaris 11 的系统上，网络配置可以是反应性的，也可以是固定的，具体取决于系统上活动的网络配置文件 (network configuration profile, NCP)。有关反应性和固定网络配置的概述，请参见《Oracle Solaris 11 联网介绍》。有关如何创建和配置 NCP 的详细信息，请参见《在 Oracle Solaris 11.1 中使用反应性网络配置连接系统》。

本章提供固定网络配置的概述，其中包含以下主题：

- 第 9 页中的“什么是固定网络配置？”
- 第 10 页中的“配置文件管理网络配置的亮点”
- 第 11 页中的“网络配置工具”

## 什么是固定网络配置？

在 Oracle Solaris 11 中，网络配置是由网络配置文件 (network configuration profile, NCP) 管理的。特定系统中有效 NCP 的类型决定该系统的网络配置。如果 NCP 是反应性的，则将动态实现该系统上的网络配置。如果 NCP 是固定的，则将静态实现网络配置。

固定网络配置是指在系统上实例化特定网络设置的配置模式。与反应性网络配置模式不同，固定配置模式中的实例化配置无论系统的网络环境是否发生更改都始终保持不变。如果该环境中发生更改（如添加接口），您必须手动重新配置系统的网络设置，从而使系统能够适应新环境。

---

注 - 不要将固定网络配置与仅配置静态 IP 地址相混淆。在固定网络配置中，您可以向接口指定 DHCP 地址。同样，在反应性网络配置中，您可以创建接口配置了静态 IP 地址的 NCP。因此，固定网络配置的范围更广泛，具体指的是系统的网络配置根据系统环境中的更改而进行更改的能力。

---

下表比较了两种网络配置模式。

功能	反应性网络配置	固定网络配置
自动适应系统网络环境中的更改	通过多个可配置的 NCP 来支持	不支持； 需要根据需求手动重新配置
系统上有效的 NCP 类型	反应性 (Automatic 或其他一些用户创建的 NCP)	固定 (DefaultFixed)
多个 NCP	支持 (但每次只能有一个 NCP 处于活动状态)	不支持
用户创建的 NCP	支持	只存在一个固定 NCP (DefaultFixed)， 它是由系统生成的。但是， DefaultFixed 的内容完全由用户决定。

下面几节将更为详细地介绍配置文件管理的网络配置和用于网络配置的工具。

## 配置文件管理网络配置的亮点

在 Oracle Solaris 11 中，网络配置基于配置文件。系统的网络配置是由 NCP 和相应的位置配置文件管理的。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 的详细信息，请参见《[在 Oracle Solaris 11.1 中使用反应性网络配置连接系统](#)》。

---

注 - 对于网络配置，基本配置文件类型包括 NCP、位置配置文件、外部网络修饰器 (external network modifier, ENM) 以及无线局域网 (wireless local area network, WLAN)。在这些类型中，主配置文件是 NCP。在整个本文档中，除非另有说明，否则术语**配置文件**均指 NCP。

---

基于配置文件的网络配置的亮点如下：

- 在同一时刻，只有一对 NCP 和位置配置文件可以处于活动状态来管理系统的网络配置。系统中所有其他现有 NCP 均不工作。
- 活动的 NCP 既可以是**反应性的**，也可以是**固定的**。反应性配置文件用于监视网络配置以适应系统网络环境中的变化。固定配置文件用于实例化（而不是监视）网络配置。
- 如果活动 NCP 是反应性的，则系统的网络配置是自适应的。如果活动 NCP 是固定的，则系统的网络配置保持不变。
- NCP 的不同属性的值构成一个策略，控制配置文件如何管理网络配置。
- 对 NCP 属性的更改将立即实现为新的属性值。这些新值将成为管理网络配置的配置文件的策略的一部分。

如果您的系统配置为固定网络，则管理其网络配置的活动 NCP 为 `DefaultFixed`。该配置文件是由 OS 生成的，并且是系统上唯一的固定配置文件。一个系统不支持多个固定配置文件。

`DefaultFixed` NCP 的属性反映在 `DefaultFixed` NCP 处于活动状态时创建或修改的永久性配置。

## 网络配置工具

在 Oracle Solaris 11 中有以下四个网络命令可用于配置网络：

- `netcfg` 命令
- `netadm` 命令
- `dladm` 命令
- `ipadm` 命令

`netcfg` 和 `netadm` 命令用于管理系统上的反应性网络配置。您可以使用 `netcfg` 命令来创建和配置实现反应性网络配置的文件：NCP、位置配置文件、ENM 和 WLAN。但是，在使用固定网络配置的系统上，您只能使用 `netcfg` 命令来查看 `DefaultFixed` 配置文件。`netadm` 命令用于管理系统上的所有配置文件，特别是列出系统的网络配置文件并将一个活动 NCP 替换为另一个。

`dladm` 和 `ipadm` 命令分别用于配置数据链路和 IP 接口。这些命令将创建永久性配置，并应用于使用这些命令时在系统上处于活动状态的配置文件。

例如，如果数据链路 `net0` 的特定最大传输单元 (maximum transmission unit, MTU) 配置为 1200，并且活动 NCP 是 `Automatic`，则该 MTU 值将成为 `Automatic` NCP 中 `net0` 的永久性值。假设您激活名为 `myncp` 的第二个 NCP。如果您发出 `dladm` 命令来设置其他值的 MTU，则该值将应用于 `myncp`。因此，在不同的配置文件中，`net0` 可以有不同的 MTU 值。因此，`dladm` 和 `ipadm` 命令还可以用于间接配置配置文件。

当您使用 `dladm` 或 `ipadm` 命令配置数据链路和 IP 接口时，请注意以下使用范围：

- 这两个命令仅配置活动配置文件的数据链路和 IP 接口。要配置该配置文件的其他属性（如设置缺省路由），请使用 `netcfg` 命令来配置引用缺省路由的配置文件属性。或者，您可以使用 `routadm` 命令直接在系统路由表上设置缺省路由。如果是后一种情况，则配置将应用于系统上处于活动状态的任何配置文件。
- 您可以在任何反应性配置文件上使用 `dladm` 和 `ipadm` 命令，前提是该配置文件处于活动状态。但是，您不能使用 `netcfg` 命令来配置 `DefaultFixed` 配置文件，该配置文件是系统的唯一固定配置文件。您只能使用 `netadm` 和 `netcfg` 命令来查看 `DefaultFixed` 配置文件的属性，但是不能配置它们。

`dladm` 和 `ipadm` 命令在活动配置文件（无论是反应性配置文件还是固定配置文件）上是有效的。因此，在使用这些命令之前，必须确保以下事项：

- 知道在系统上处于活动状态的配置文件，以确保您对正确的目标配置文件做出更改。

- 知道目标配置文件是反应性的还是固定的，以避免在使用命令后导致意外的配置行为。反应性配置文件管理网络配置的方式与固定配置文件不同。因此，实现更改时，这两个配置文件的行为也不同。

以下几节将详细介绍 `dladm` 和 `ipadm` 命令。

## dladm 命令

使用 `dladm` 命令配置数据链路。只要链路的网络驱动程序已转换为 GLDv3 驱动程序配置框架，如 `e1000g`，就可以使用 `dladm` 命令定制数据链路属性。要确认您的特定驱动程序是否支持此功能，请参阅驱动程序的手册页。

GLDv3 驱动程序配置框架的完整实现以下列几种方式增强了网络接口卡 (Network Interface Card, NIC) 驱动程序的配置：

- 配置网络驱动程序属性只需一个命令接口，即 `dladm` 命令。
- 对任何属性均使用统一的语法：`dladm subcommand properties datalink`。
- `dladm` 命令对驱动程序的公共属性和专用属性均适用。
- 对特定驱动程序使用 `dladm` 命令不会中断相似类型的其他 NIC 的网络连接。因此，您可以动态地配置数据链路属性。
- 数据链路配置值存储在一个 `dladm` 系统信息库中，即便在重新引导系统后仍然存在。

要在配置数据链路时利用这些优点，您应该使用 `dladm` 作为配置工具而不是早期发行版中惯用的工具，例如 `ndd` 命令。

有关 `dladm` 命令的更多详细信息，请参阅 [dladm\(1M\)](#) 手册页。要获取与 `dladm` 命令一起使用的子命令的列表，请键入以下内容：

```
# dladm help
The following subcommands are supported:
Bridge      : add-bridge      create-bridge   delete-bridge
              modify-bridge  remove-bridge   show-bridge
Etherstub   : create-etherstub delete-etherstub show-etherstub
IB          : create-part   delete-part     show-ib         show-part
IP tunnel   : create-iptun  delete-iptun   modify-iptun    show-iptun
Link Aggregation: add-aggr      create-aggr    delete-aggr
              modify-aggr   remove-aggr    show-aggr
Link        : rename-link  reset-linkprop set-linkprop
              show-link    show-linkprop
Secure Object : create-secobj  delete-secobj  show-secobj
VLAN        : create-vlan  delete-vlan    modify-vlan      show-vlan
VNIC        : create-vnic  delete-vnic    modify-vnic      show-vnic
Wifi        : connect-wifi disconnect-wifi scan-wifi        show-wifi
Miscellaneous : delete-phys  show-ether     show-phys        show-usage
For more info, run: dladm help <subcommand>.
```

要在数据链路上使用 `dladm` 命令，请参见第 3 章，[使用数据链路](#)。

## ipadm 命令

Oracle Solaris 中的改进已超越了传统工具的功能，可以有效地管理网络配置的各个方面。例如，`ifconfig` 命令已成为用来配置网络接口的惯用工具。但是，此命令不能实现持久性配置。随着时间的推移，`ifconfig` 添加了网络管理方面的功能，从而得到了增强。然而，该命令同时也变得复杂起来，不便于使用。

接口配置和管理的另一个问题是没有简单的工具来管理 TCP/IP 属性或可调参数。因此，`ndd` 命令被指定为定制工具。然而，与 `ifconfig` 命令一样，`ndd` 也不能实现持久性配置。以前，可以通过编辑引导脚本模拟网络方案的持久性配置。随着 Oracle Solaris 引入服务管理工具 (service management facility, SMF)，使用此类替代方法可能风险很高，因为管理 SMF 相关项相当复杂，尤其是在升级到 Oracle Solaris 安装时。

`ipadm` 命令的引入最终会取代 `ifconfig` 命令来进行接口配置。此命令还将取代 `ndd` 命令来配置协议属性。

作为一个接口配置工具，`ipadm` 命令具有以下优点：

- 它管理 IP 接口和 IP 地址更有效率，因为它是专用于 IP 接口管理的工具，而 `ifconfig` 命令除了用于接口配置外，还用于其他目的。
- 它实现了持久性接口和地址配置。

有关 `ifconfig` 选项及其等效 `ipadm` 子命令的列表，请参见附录 A，[对应关系比较：ifconfig 和 ipadm 命令](#)。

作为设置协议属性的工具，`ipadm` 命令相较于 `ndd` 命令，具有以下优点：

- 它可以为以下协议设置临时或持久性属性：IP、地址解析协议 (Address Resolution Protocol, ARP)、流控制传输协议 (Stream Control Transmission Protocol, SCTP)、Internet 控制消息协议 (Internet Control Messaging Protocol, ICMP) 以及上层协议，如 TCP 和用户数据报协议 (User Datagram Protocol, UDP)。
- 它提供了有关每个 TCP/IP 属性的信息，如属性的当前和缺省值以及可能值的范围。因此，更容易获得调试信息。
- 它还遵循一致的命令语法，因此更易于使用。

有关 `ndd` 选项及其等效 `ipadm` 子命令的列表，请参见附录 B，[对应关系比较：ndd 和 ipadm 命令](#)。

有关 `ipadm` 命令的更多详细信息，请参阅 [ipadm\(1M\)](#) 手册页。要列出可以与 `ipadm` 一起使用的子命令，请键入以下内容：

```
# ipadm help
The following subcommands are supported:
Address          : create-addr  delete-addr  disable-addr
                  down-addr      enable-addr  refresh-addr
                  reset-addrprop set-addrprop show-addr
                  show-addrprop up-addr
```

```
Interface          : disable-if   enable-if   reset-ifprop
                   : set-ifprop   show-if    show-ifprop
IP interface       : create-ip   delete-ip
IPMP interface     : add-ipmp    create-ipmp delete-ipmp
                   : remove-ipmp
Protocol property  : reset-prop  set-prop   show-prop
VNI interface      : create-vni  delete-vni
For more info, run: ipadm help <subcommand>.
```

## 配置系统网络

---

本章提供了您在使用固定网络配置的系统上配置 IP 接口所需遵循的过程。本章包含以下主题：

- 第 15 页中的“配置网络（任务列表）”
- 第 22 页中的“其他网络配置和管理任务”

### 配置网络（任务列表）

本节介绍了 IP 接口的基本配置过程。下表介绍了配置任务，并提供了与这些任务相对应的过程的链接。

任务	说明	参考
将系统配置为支持唯一的 MAC 地址。	将基于 SPARC 的系统配置为允许接口使用唯一 MAC 地址。	第 16 页中的“SPARC: 如何确保每个接口的 MAC 地址是唯一的”
确定在系统上处于活动状态的 NCP。	显示系统上的活动 NCP 并启用 DefaultFixed。	第 17 页中的“如何更改系统上的活动 NCP”
通过使用 ipadm 命令执行基本 IP 接口配置。	创建一个 IP 接口并为此接口指定有效 IP 地址（静态或 DHCP）。	第 18 页中的“如何配置 IP 接口”
定制数据链路。	通过设置链路属性进一步定制数据链路。	第 25 页中的“定制数据链路属性”
定制 IP 接口。	通过设置接口属性进一步定制 IP 接口。	第 37 页中的“设置 IP 接口属性”
定制 IP 地址。	通过设置地址属性进一步定制 IP 地址。	第 38 页中的“设置 IP 地址属性”
定制协议。	通过设置协议属性进一步定制协议。	第 39 页中的“设置 TCP/IP 协议属性”

任务	说明	参考
配置无线网络。	使用无线网络将手提电脑连接到网络。	第 5 章, 在运行 Oracle Solaris 的手提电脑上配置无线联网

## ▼ SPARC: 如何确保每个接口的 MAC 地址是唯一的

每个基于 SPARC 的系统都有一个系统范围的 MAC 地址, 缺省情况下所有接口都使用该地址。但是, 某些应用程序要求主机上的每个接口都有唯一的 MAC 地址。类似的, 某些类型的接口配置 (如链路聚合和 IP 多路径 (IP multipathing, IPMP)) 也要求接口必须具有自己的 MAC 地址。

EEPROM 参数 `local-mac-address?` 确定基于 SPARC 的系统上的所有接口是使用系统范围的 MAC 地址还是使用自己的唯一 MAC 地址。下面的过程说明如何使用 `eeprom` 命令检查 `local-mac-address?` 的当前值以及更改该值 (如有必要)。

### 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 确定系统中的所有接口当前是否都使用系统范围的 MAC 地址。

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

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

### 3 如有必要, 请按如下所示更改 `local-mac-address?` 的值:

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

在步骤 6 中重新引导系统时, 具有出厂 MAC 地址的接口将使用这些出厂设置, 而不是系统范围的 MAC 地址。不具有出厂 MAC 地址的接口将继续使用系统范围的 MAC 地址。

### 4 检查系统中所有接口的 MAC 地址。

查找多个接口具有相同 MAC 地址的情况。在此示例中, 有两个接口使用系统范围的 MAC 地址 `8:0:20:0:0:1`。

```
# dladm show-linkprop -p mac-address
LINK  PROPERTY  PERM VALUE          DEFAULT          POSSIBLE
net0   mac-address rw  8:0:20:0:0:1     8:0:20:0:0:1    --
net1   mac-address rw  8:0:20:0:0:1     8:0:20:0:0:1    --
net3   mac-address rw  0:14:4f:45:c:2d  0:14:4f:45:c:2d --
```

注 – 仅当有两个或两个以上的网络接口具有相同 MAC 地址时继续执行下一步。否则，前进到最后一步。

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

```
# dladm set-linkprop -p mac-address=mac-address interface
```

对于上一步的示例，需要用本地管理的 MAC 地址配置 net0 和 net1。例如，要用本地管理的 MAC 地址 06:05:04:03:02 重新配置 net0，您应输入以命令：

```
# dladm set-linkprop -p mac-address=06:05:04:03:02 net0
```

有关此命令的更多信息，请参阅 [dladm\(1M\)](#) 手册页。

- 6 重新引导系统。

## ▼ 如何更改系统上的活动 NCP

系统上启用的 NCP 类型决定系统的网络配置是反应性的还是固定的。具有反应性配置的系统与具有固定网络配置的系统行为不同。本书中的所有过程创建的永久性配置均应用于活动 NCP。因此，在执行任何过程之前，您必须了解哪个 NCP 处于活动状态，以便将配置应用于正确的配置文件。这样，在完成过程后，系统的网络配置便可按预期操作。

- 1 列出系统上的配置文件。

```
# 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。

例如，假设您的系统网络配置是反应性的。如果您希望将使用本书中的过程创建的配置应用于 DefaultFixed NCP，请键入以下命令：

```
# netadm enable -p ncp defaultfixed
```



**注意** - 切换活动配置文件时，会删除现有的网络配置并创建新的配置。在新的活动 NCP 中，会排除在以前的活动 NCP 上实现的任何持久性配置。

## ▼ 如何配置 IP 接口

以下过程提供了可用于配置系统 IP 接口的基本步骤。

**开始之前** 检查在系统上处于活动状态的 NCP 以确保您将配置应用于正确的配置文件。

- 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”。

- 创建接口。

```
# ipadm create-interface-class interface
```

*interface-class* 指您可以创建的以下三类接口之一：

- IP 接口。此接口类是您执行网络配置时最常创建的接口类。要创建此接口类，请使用 `create-ip` 子命令。
- STREAMS 虚拟网络接口驱动程序 (VNI 接口)。要创建此接口类，请使用 `create-vni` 子命令。有关 VNI 设备或接口的更多信息，请参见 [vni\(7d\)](#) 手册页。
- IPMP 接口。此接口用于配置 IPMP 组。要创建此接口类，请使用 `create-ipmp` 子命令。有关 IPMP 组的更多信息，请参见《管理 Oracle Solaris 11.1 网络性能》中的第 5 章“IPMP 介绍”。

*interface* 指接口的名称。该名称与用于创建接口的数据链路的名称相同。要了解系统上的数据链路，请使用 `dladm show-link` 命令。

- 通过选择以下命令之一配置具有有效 IP 地址的 IP 接口。

- 要配置静态地址，请键入以下内容：

```
# ipadm create-addr -a address [interface | addrobj]
```

-a *address*

指定要在接口上配置的 IP 地址。

---

注-隧道配置通常需要隧道接口的两个地址：一个本地地址和一个远程地址。有关本地和远程地址以及隧道配置的信息，请参见《配置和管理 Oracle Solaris 11.1 网络》中的第 6 章“配置 IP 隧道”。

---

对于数字 IP 地址，请使用 CIDR 表示法。如果您不使用 CIDR 表示法，则会根据 *name-service/switch* 服务中为 *netmask* 列出的序列或通过使用有类地址语义计算网络掩码。

您也可以选择指定主机名，而非数字 IP 地址。如果在 */etc/hosts* 文件中为主机名定义了相应的数字 IP 地址，则使用该主机名是有效的。如果未在文件中定义数字 IP 地址，则该数值可通过使用在 *name-service/switch* 服务中为 *host* 指定的解析器顺序唯一获取。如果给定的主机名存在多项，则会生成错误。

---

注-在引导过程中，先创建 IP 地址，再使命名服务联机。因此，您必须确保在 */etc/hosts* 文件中定义用于网络配置的任何主机名。

---

[*interface* | *addrobj*]

在 Oracle Solaris 中，每个地址由相应的地址对象标识，并由 *addrobj* 在命令中表示。对于此地址的任何后续配置，您可以参考地址对象而非实际 IP 地址。例如，您可以键入 *ipadm show-addr addrobj* 或 *ipadm delete-addr addrobj*。要自动创建地址对象名称，请指定 *interface* 的接口名称。否则，请直接提供地址对象名称。

- 如果您指定接口名称，则地址对象将以 *interface/address-family* 格式自动命名。*Address family* 是 IPv4 地址的 *v4* 或 IPv6 地址的 *v6*。同一接口上的多个地址的地址对象名称附加有字母，如 *net0/v4*、*net0/v4a*、*net0/v4b*、*net0/v6*、*net0/v6a* 等。
  - 如果您手动命名 *addrobj* 的地址对象，则必须使用格式 *interface/user-specified-string*。*User-specified-string* 表示由字母数字字符组成的字符串，该字符串以字母开头且最大长度为 32 个字符。例如，您可以命名地址对象 *net0/static*、*net0/static1*、*net1/private* 等。
- 要配置非静态地址，请键入以下内容：

```
# ipadm create-addr -T address-type [interface | addrobj]
```

其中 *address-type* 为 `dhcp` 或 `addrconf`。Addrconf 指自动生成的 IPv6 地址。

有关 `[interface | addrobj]` 的更完整的说明，请参阅有关创建静态地址的先前描述。

#### 4 (可选) 显示有关新配置的 IP 接口的信息。

您可以根据要检查的信息使用以下命令：

```
# ipadm [interface]
```

如果您未指定 *interface*，则会显示系统中所有接口的信息。

有关 `ipadm show-*` 子命令的输出的更多信息，请参见第 44 页中的“监视 IP 接口和地址”。

#### 5 如果您要配置使用主机名的静态 IP 地址，请在 `/etc/hosts` 文件中添加 IP 地址对应的各项。

此文件中的项由 IP 地址和其对应的主机名组成。

---

注 – 如果您要配置 DHCP 地址，则不需要更新 `/etc/hosts` 文件。

---

#### 6 定义缺省路由。

```
# route -p add default address
```

您可以使用 `netstat -r` 命令验证路由表的内容。

有关管理路由的详细信息，请参见 `route(1M)` 和 `routadm(1M)` 手册页。另请参见《配置和管理 Oracle Solaris 11.1 网络》中的“路由表和路由类型”。

### 示例 2-1 为网络接口配置静态 IP 地址

本示例介绍如何为接口配置静态 IP 地址。该示例首先在系统上启用 `DefaultFixed` NCP，从而允许您使用 `dladm` 和 `ipadm` 命令进行固定网络配置。

```
# netadm enable -p ncp DefaultFixed

# dladm show-phys
LINK      MEDIA      STATE      SPEED      DUPLEX      DEVICE
net3      Ethernet   up         100Mb      full        bge3

# dladm show-link
LINK      CLASS      MTU        STATE      BRIDGE      OVER
net3      phys       1500      up         --          --

# ipadm create-ip net3
# ipadm create-addr -a 192.168.84.3/24 net3
ipadm: net3/v4

# ipadm
NAME      CLASS/TYPE  STATE      UNDER      ADDR
lo0       loopback   ok         --          --
```

```

    lo/v4      static  ok    --    127.0.0.1/8
net3    ip      ok    --    --
net3/v4    static  ok    --    192.168.84.3/24

# vi /etc/hosts
# Internet host table
# 127.0.0.1      localhost
10.0.0.14      myhost
192.168.84.3   campus01

# route -p add default 192.168.84.1
# netstat -r
Routing Table: IPv4
  Destination          Gateway                Flags  Ref    Use    Interface
-----
default                some.machine.com      UG      2     10466
192.168.84.0          192.168.84.3         U        3     1810  net3
localhost              localhost              UH      2      12    lo0

Routing Table: IPv6
  Destination/Mask      Gateway                Flags  Ref    Use    If
-----
solaris                 solaris                UH      2     156   lo0

```

请注意，如果 campus01 已在 /etc/hosts 文件中定义，可以在指定以下地址时使用主机名：

```

# ipadm create-addr -a campus01 net3
ipadm: net3/v4

```

## 示例 2-2 自动使用 IP 地址配置网络接口

在此示例中，IP 接口配置为从 DHCP 服务器接收其地址。

```

# dladm show-phys
LINK    MEDIA    STATE    SPEED    DUPLEX    DEVICE
net3    Ethernet up        100Mb    full     bge3

# dladm show-link
LINK    CLASS    MTU    STATE    BRIDGE    OVER
net3    phys     1500   up        --        --

# ipadm create-ip net3
# ipadm create-addr -T dhcp net3
ipadm: net3v4

# ipadm
NAME    CLASS/TYPE    STATE    UNDER    ADDR
lo0     loopback     ok      --        --
  lo/v4      static       ok      --        127.0.0.1/8
net3    ip           ok      --        --
  net3/v4    dhcp        ok      --        10.0.1.13/24

```

## 其他网络配置和管理任务

本书介绍了将您的系统连接到网络的基本网络配置。具体来说，这些信息重点描述系统数据链路和接口的配置。可以执行有关网络的其他书籍中介绍的其他网络配置和管理任务。假定您的系统已配置为固定网络配置，您可以参阅有关其他任务的以下书籍：

- 要将系统配置为路由器、网络配置服务器等，请参见《[配置和管理 Oracle Solaris 11.1 网络](#)》。
- 要执行高级数据链路和 IP 接口配置，请参见《[管理 Oracle Solaris 11.1 网络性能](#)》。
- 要构建基本配置并提高网络性能（如配置链路聚合、IPMP 组等），请参见《[管理 Oracle Solaris 11.1 网络性能](#)》。
- 要建立网络安全性，请参见《[在 Oracle Solaris 11.1 中保护网络安全](#)》。
- 要实现网络虚拟化，请参见《[在 Oracle Solaris 11.1 中使用虚拟网络](#)》。

该库中还提供了专门研究特定网络领域（如 DHCP、名称服务等）的其他书籍。

## 使用数据链路

---

本章讨论 `dladm` 命令并说明了如何对数据链路使用该命令来显示其当前配置、更改其属性的缺省值，或者从系统中删除数据链路。本章包含以下主题：

- 第 23 页中的“基本 `dladm` 命令”
- 第 25 页中的“定制数据链路属性”
- 第 30 页中的“使用 `dladm` 命令的其他配置任务”

### 基本 `dladm` 命令

本节介绍您可能会对系统数据链路经常使用的基本 `dladm` 命令。除本节中列出的子命令以外，还支持其他 `dladm` 子命令。有关其他子命令，请参见 `dladm(1M)` 手册页。

---

注 - 除了显示数据链路信息的 `dladm` 子命令外，其他所有子命令均需要先删除数据链路上现有的所有接口配置。要删除 IP 接口配置，请参见第 35 页中的“删除 IP 接口配置 (`ipadm delete-ip`)”。

---

### 显示有关数据链路的常规信息 (`dladm`)

如果是由自身使用，`dladm` 命令将显示有关系统数据链路的常规信息，包括其类、状态和底层物理链路。

```
# dladm
LINK      CLASS    MTU     STATE   OVER
net0     phys    1500   unknown --
net1     phys    1500    up     --
net2     phys    1500   unknown --
net3     phys    1500   unknown --
net4     phys    1500    up     --
aggr0    aggr    1500    up     net1,net4
```

数据链路可以是物理链路以外的其他类，例如链路聚合、虚拟 LAN (virtual LAN, VLAN) 或虚拟 NIC (virtual NIC, VNIC)。这些其他数据链路也包含在 `dladm` 命令显示的缺省信息中。例如，输出显示了在物理数据链路 `net1` 和 `net4` 上配置的链路聚合 `aggr0`。

有关链路聚合和 VLAN 的信息，请参见《管理 Oracle Solaris 11.1 网络性能》。有关 VNIC 的信息，请参见《在 Oracle Solaris 11.1 中使用虚拟网络》。

## 显示系统的数据链路 (`dladm show-link`)

使用 `dladm show-link` 可显示系统上的数据链路。系统包含的数据链路与已安装的 NIC 一样多。可以将选项与此命令一起使用来定制获取的信息。例如，使用 `-P` 选项可包含有关数据链路的持久性配置信息。基于该命令提供的信息，您可以进行进一步的网络配置。例如，您可以确定系统上 NIC 的数量，然后可以选择要使用的数据链路，在该数据链路上，您可以配置 IP 接口。

发出该命令时，将显示类似于以下内容的信息：

```
# dladm show-link -P
LINK      CLASS    OVER
net0      phys     --
net1      phys     --
net2      phys     --
```

该示例显示了系统拥有三个与对应物理 NIC 直接相关联的数据链路。不存在任何特殊数据链路，如在 `phys` 类下的数据链路上配置的聚合或虚拟 NIC。

## 显示数据链路的物理属性 (`dladm show-phys`)

使用 `dladm show-phys` 可获取与物理 NIC（与系统数据链路相关联）相关的系统数据链路的信息。如果不使用任何选项，该命令将显示类似于以下内容的信息：

```
# dladm show-phys
LINK      MEDIA      STATE    SPEED    DUPLEX    DEVICE
net0      Ethernet   up       100Mb    full      e1000g0
net1      Ethernet   down     0Mb      --        nge0
net2      Ethernet   up       100Mb    full      bge0
net3      Infiniband --        0Mb      --        ibd0
```

除了其他详细信息，输出还显示与具有通用链路名称的数据链路相关联的物理 NIC。例如，`net0` 是 NIC `e1000g0` 的数据链路名称。要查看有关为数据链路设置的标志的信息，请使用 `-P` 选项。例如，标记有 `r` 的数据链路意味着其底层 NIC 已被移除。

该命令的另一个有用选项是 `-L`，该选项显示每个数据链路的物理位置。位置确定数据链路的实例编号，如 `net0`、`net1` 等。

```
# dladm show-phys -L
LINK      DEVICE      LOCATION
net0      bge0        MB
net2      ibp0        MB/RISER0/PCIE0/PORT1
net3      ibp1        MB/RISER0/PCIE0/PORT2
net4      eoib2       MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

## 删除数据链路 (dladm delete-phys)

使用 `dladm delete-phys` 可从系统中删除数据链路。

删除数据链路与移除物理 NIC 没有紧密联系。例如，从系统中移除物理 NIC。与该 NIC 相关的数据链路配置将保留下来，因为软件层不再绑定到硬件层，如《[Oracle Solaris 11 联网介绍](#)》中的“[Oracle Solaris 中的网络栈](#)”中所述。因此，您仍然可以通过将该数据链路的名称指定给其他 NIC 的关联链路来在其他底层物理 NIC 中使用该数据链路配置。

如果您要分离 NIC 而不替换它，且不再需要其数据链路配置，则可以删除该数据链路，如下所示：

```
# dladm delete-phys datalink
```

---

**提示** – 要确认数据链路的 NIC 是否已移除，请使用 `dladm show-phys -P` 命令。

---

## 重命名数据链路 (dladm rename-link)

使用 `dladm rename-link` 可重命名数据链路。在 Oracle Solaris 11 系统上，OS 会自动提供所有数据链路的通用名称。《[Oracle Solaris 11 联网介绍](#)》中的“[缺省通用链路名称](#)”中介绍了通用数据链路名称。

缺省情况下，这些通用名称使用命名格式 `netn`，如 `net0`、`net1`、`net2` 等。由于 OS 管理这些名称，因此您无法在管理任务的常规过程中重命名数据链路。有关需要更改链路名称的过程，请参见第 31 页中的“[如何在系统中切换主接口](#)”。

## 定制数据链路属性

除了执行基本的数据链路配置，您还可以使用 `dladm` 命令设置数据链路属性并根据您的网络要求对其进行定制。

有以下三个用于数据链路属性的 `dladm` 子命令：

- `dladm show-linkprop [-p property] [datalink]` 显示数据链路属性及属性的当前值。如果不使用 `-p property` 选项，则会列出数据链路的所有属性。如果不指定数据链路，则会列出所有数据链路的所有属性。

- `dladm set-linkprop -p property =value datalink` 为数据链路的属性赋值。
- `dladm reset-linkprop -p property datalink` 将特定属性重置为其缺省值。

## 数据链路属性概述

可定制的数据链路属性取决于特定 NIC 驱动程序支持的属性。可通过使用 `dladm` 命令配置的数据链路属性分为两个类别：

- **公共属性**：可应用于给定介质类型的任何驱动程序，例如链路速度、以太网的自动协商，或可应用于数据链路所有驱动程序的最大传输单元 (`maximum transmission unit, MTU`) 大小。
- **专用属性**：仅属于给定介质类型的 NIC 驱动程序的特定子集。这些属性可能是特定于该子集的，因为它们与驱动程序相关联的硬件密切相关，或者与驱动程序实现本身的详细信息（如与调试相关的可调参数）密切相关。

链路属性通常具有缺省值。但是，在某些网络情况下，可能需要您更改特定属性值。例如，NIC 可能与不能正确执行自动协商的旧交换机进行通信。或者，一个交换机可能已配置为支持巨型帧。或者，可能需要针对特定驱动程序修改用于控制数据包传输或数据包接收的驱动程序特定属性。以下各节有选择地介绍了部分属性并解释了如何更改它们的值以便在网络环境中起作用。

## 启用巨型帧支持

MTU 定义协议可从系统中传输的最大包的大小。缺省情况下，大部分 NIC 驱动程序将 MTU 大小定义为 1500。但是，如果巨型帧要遍历网络，缺省值将不够用。要支持巨型帧，MTU 大小至少需要为 9000。

要更改 MTU 大小的缺省值，请键入以下命令：

```
# dladm set-linkprop -p mtu=new-size datalink
```

更改 MTU 大小后，您可以在数据链路上重新配置一个 IP 接口

以下示例显示了启用对巨型帧的支持的步骤。此示例假定您已删除了数据链路上现有的所有 IP 接口配置。

```
# dladm show-linkprop -p mtu net1
LINK    PROPERTY  VALUE    DEFAULT  POSSIBLE
net1    mtu       1500    1500    --
# dladm set-linkprop -p mtu=9000 net1
# dladm show-link web1
LINK    CLASS    MTU     STATE    BRIDGE    OVER
web1    phys     9000    up       --        --
```

## 修改链路速度参数

大多数网络设置包含多个具有不同速度能力的系统。每个系统均会将速度能力通告给网络中的其他系统，网络通知每个系统如何传输和接收网络通信。以下成对的数据链路属性控制系统通告的速度能力：

- `adv_10gfdx_cap/en_10gfdx_cap`
- `adv_1000fdx_cap/en_1000fdx_cap`
- `adv_1000hdx_cap/en_1000hdx_cap`
- `adv_100fdx_cap/en_100fdx_cap`
- `adv_100hdx_cap/en_100hdx_cap`
- `adv_10fdx_cap/en_10fdx_cap`
- `adv_10hdx_cap/en_10hdx_cap`

每个链路速度能力均涉及一对属性：通告速度 (`adv_*_cap`) 和支持通告速度 (`en_*_cap`)。此外，还将为全双工和半双工能力提供数据链路速度信息，如属性名称中的 `*fdx*` 和 `*hdx*` 所指定。通告速度属性为只读属性，指示特定数据链路速度是否已通告。您可以通过设置相应的 `en_*_cap` 属性来确定特定数据链路速度是否已通告。

缺省情况下，通告数据链路的所有速度和双工能力。但是，可能会存在以下情况：新系统正在与旧系统通信且自动协商已禁用或不受支持。要启用这两个系统间的通信，较旧系统与较新系统之间的通告速度可能需要更改为较低的值。系统的 GB 能力可能需要关闭，并且只通告较低的速度能力。在这种情况下，您可以为全双工能力和半双工能力键入以下内容。

```
# dladm set-linkprop -p en_1000fdx_cap=0 datalink
# dladm set-linkprop -p en_1000hdx_cap=0 datalink
```

该命令为全双工能力和半双工能力关闭对系统的 GB 能力的通告。

要显示这些属性的新值，请使用 `dladm show-linkprop` 命令。

```
# dladm show-linkprop -p adv_10gfdx_cap datalink
# dladm show-linkprop -p adv_1000hdx_cap datalink
```

通常情况下，给定的支持速度属性和相应的通告属性的值是相同的。不过，如果 NIC 支持一些高级功能（如电源管理），这些功能可能对主机和其链路伙伴之间实际通告的位设置限制。例如，使用电源管理时，`adv_*_cap` 属性的设置可能只是 `en_*_cap` 属性的设置的子集。

## 设置数据链路上的 STREAMS 模块

当数据链路处于打开状态时，可以设置将最多 8 个 STREAMS 模块推送到流。这些模块通常由第三方网络软件（如虚拟专用网络 (virtual private networks, VPN) 和防火墙）使用。软件供应商会提供有关此类网络软件的文档。

要在特定数据链路上推送的模块的列表是由 `autopush` 属性控制的。而 `autopush` 属性的值是通过使用 `dladm set-linkprop` 子命令设置的。

也可以使用单独的 `autopush` 命令基于每个驱动程序将模块推送到数据链路的流。该命令使用为每个驱动程序设置的配置文件，且此配置文件会就要推送的模块通知命令。然而，驱动程序始终绑定到 NIC。如果移除了数据链路的底层 NIC，则链路的 `autopush` 属性信息也会丢失。

因此，基于此目的，`dladm` 命令比 `autopush` 命令更可取。如果特定数据链路的 `autopush` 配置的每驱动程序类型和每链路类型都存在，则使用通过 `dladm set-linkprop` 设置的每链路信息，而忽略每驱动程序信息。

要在数据链路处于打开状态时将模块推送到 STREAMS，您可以使用相同的 `dladm set-linkprop` 命令为 `autopush` 属性指定模块。例如，要将 `vpnmod` 和 `bufmod` 模块推送到 `net0` 链路上，应键入：

```
# dladm set-linkprop -p autopush=vpnmod.bufmod net0
```

## 将 e1000g 驱动程序设置为使用直接内存访问绑定

本节和下一节显示如何配置专用属性。这两节适用于特定于 `e1000g` 驱动程序的属性。不过，这两节中的常规信息也适用于配置其他 NIC 驱动程序的专用属性。

大容量通信，例如文件传输，通常涉及通过网络的大数据包的协商。在此类情况下，您可以通过将 `e1000g` 驱动程序配置为自动使用直接内存访问 (Direct Memory Access, DMA) 绑定（其中为数据包段大小定义阈值），从该驱动程序获得更好的性能。如果一个段大小超过阈值，则使用 DMA 绑定传输包。如果一个段大小在阈值内，则使用 `bcopy` 模式，其中段数据被复制到预分配的传输缓冲区。

```
# dladm set-linkprop -p _tx_bcopy_threshold=value datalink
```

对于此属性，阈值的有效值范围为 60 到 2048。

---

注 - 所有数据链路均自动命名为通用名称。必须确保该专用属性是在底层 NIC 为 `e1000g` 的数据链路上配置的。设置该属性前使用 `dladm show-phys` 验证。

与配置公共属性一样，也必须先删除所有 IP 接口，然后才能修改专用属性值。

---

您可能会执行类似于以下内容的步骤：

```
# dladm show-phys
LINK  MEDIA   STATE  SPEED  DUPLEX  DEVICE
net0  Ethernet up     100Mb  full   nge0
net1  Ethernet up     100Mb  full   e1000g0

# dladm set-linkprop -p _tx_bcopy_threshold=1024 net1
```

## 手动设置中断率

控制由 e1000g 驱动程序提供中断的速率的属性也会影响网络和系统性能。通常，通过为每个数据包生成一个中断来将网络数据包传送到栈的上层。进而，缺省情况下，中断率由内核中的 GLD 层自动调整。但是，这种模式可能并非对所有的网络通信流量情况都是适宜的。有关此问题的讨论，请参阅 1996 年 USENIX 技术会议上发表的文档 (<http://www.stanford.edu/class/cs240/readings/mogul.pdf>)。因此，在某些情况下，必须手动设置中断率以获得更好的性能。

要定义中断率，需要设置以下属性：

- `_intr_throttling_rate` 确定中断断言之间的延迟，而不管网络通信流量情况如何。
- `_intr_adaptive` 确定是否启用中断限制率的自动调优。缺省情况下，此属性处于启用状态。

首先关闭中断限制率的自动调优。然后，手动设置中断限制率属性。

假设您拥有一个基于 x86 的系统，且该系统的 e1000g NIC 的中断限制率需要修改。进一步假设 e1000g0 的数据链路名称为 net1。应键入以下命令。

```
# dladm set-linkprop -p _intr_adaptive=0 net1
# dladm set-linkprop -p _intr-throttling_rate=1024 net1
```

## 获取有关数据链路属性的状态信息

要获取有关数据链路属性的信息，可以使用以下任一命令：

- `dladm show-linkprop [-p property] [datalink]`
- `dladm show-ether datalink`

### 显示数据链路属性 (dladm show-linkprop)

此方法在第 25 页中的“定制数据链路属性”中有所介绍。要显示数据链路属性的完整列表，请键入该命令，而不指定属性。例如：

```
# dladm show-linkprop net1
LINK      PROPERTY      VALUE      DEFAULT      POSSIBLE
net1      speed          1000      --           --
net1      autopush      --         --           --
net1      zone          --         --           --
net1      duplex        half       --           half,full
net1      state         unknown    up           up,down
net1      adv_autoneg_cap 1          1            1,0
net1      mtu           1500      1500        --
net1      flowctrl      no         bi           no,tx,rx,bi
net1      adv_1000fdx_cap 1          1            1,0
net1      en_1000fdx_cap 1          1            1,0
net1      adv_1000hdx_cap 1          1            1,0
```

```

net1      en_1000hdx_cap      1      1      1,0
net1      adv_100fdx_cap     0      0      1,0
net1      en_100fdx_cap     0      0      1,0
net1      adv_100hdx_cap     0      0      1,0
net1      en_100hdx_cap     0      0      1,0
net1      adv_10fdx_cap     0      0      1,0
net1      en_10fdx_cap     0      0      1,0
net1      adv_10hdx_cap     0      0      1,0
net1      en_10hdx_cap     0      0      1,0

```

## 显示以太网属性值 (dladm show-ether)

如果没有随 `dladm show-ether` 命令使用任何选项，则只显示数据链路的当前以太网属性值。要获取缺省情况下提供的信息以外的更多信息，请使用 `-x` 选项。以下是如何使用该命令的示例：

```

# dladm show-ether -x net1
LINK      PTYPE      STATE      AUTO      SPEED-DUPLEX      PAUSE
net1      current    up         yes       1G-f             both
--        capable    --        yes       1G-fh,100M-fh,10M-fh  both
--        adv        --        yes       100M-fh,10M-fh    both
--        peeradv   --        yes       100M-f,10M-f     both

```

如果使用 `-x` 选项，该命令还显示指定链路的内置能力，以及主机和链路伙伴之间当前通告的能力。以下介绍上一示例中显示的信息：

- 对于以太网设备的当前状态，该链路处于活动状态并在全双工模式下以每秒 1 GB 速度工作。其自动协商功能已启用并采用双向流量控制，即主机和链路伙伴都可以发送和接收 PAUSE 帧。此信息显示在输出的第一行中。
- 随后的行显示有关数据链路速度能力、已通告的实际数据链路速度以及来自对等方系统的信息，如下所述：
  - 列出以太网设备的功能。协商类型可以设置为自动。此外，该设备可以支持的速度有每秒 1 GB、每秒 100 MB 和每秒 10 MB（全双工和半双工）。同样，主机和链路伙伴之间可以双向接收或发送 PAUSE 帧。
  - `net1` 的功能按如下所示通告：自动协商、速度双工和 PAUSE 帧的流控制。
  - 同样，`net1` 的链路或对等伙伴通告以下功能：自动协商、速度双工和 PAUSE 帧的流控制。

## 使用 dladm 命令的其他配置任务

本节介绍因使用 `dladm` 命令而简化的其他配置过程，如切换主接口或执行动态重新配置 (dynamic reconfiguration, DR)。

## ▼ 如何在系统中切换主接口

更改系统的主接口时，也需要重命名数据链路。以下过程基于下列系统配置：

- 系统拥有两个数据链路：net0 和 net1。
- 底层 NIC 分别是 e1000g0 和 nge0。
- 在 net0 上配置了一个 IP 接口。该 IP 接口总是使用底层数据链路的名称。

系统的主接口是基于实例编号为零 (0) 的 net0。主接口配置在 e1000g0 上。以下步骤将引导您在 nge0 上进行数据链路配置以实现主接口配置。

### 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 显示系统数据链路的物理属性。

```
# dladm show-phys
```

### 3 删除主 IP 接口。

```
# ipadm delete-ip interface
```

---

注 – 有关 ipadm 命令的更多信息，请参见第 4 章，使用 IP 接口以及 ipadm(1M) 手册页。

---

### 4 将主链路的名称替换为系统中其他数据链路均未使用的名称。

```
# dladm rename-link primary-link unused-name
```

### 5 将主链路的名称分配给指定要成为主设备的数据链路。

```
# dladm rename-link new-link primary-link
```

## 示例 3-1 切换主接口

以下示例合并了在系统中更改主接口这一过程中的所有步骤。在示例末尾，在 e1000g0 上配置的主接口将替换为在 nge0 上配置的接口。将主链路切换到其他 NIC 后，可以在新 NIC 的数据链路上配置接口。

```
# dladm show-phys
LINK MEDIA STATE SPEED DUPLEX DEVICE
net0 Ethernet up 100Mb full e1000g0
net1 Ethernet up 100Mb full nge0

# ipadm delete-ip net0
# dladm rename-link net0 oldnet0
# dladm rename-link net1 net0

# ipadm create-ip net0
# ipadm create-addr -a 192.168.10.10/24 net0
```

```

ipadm: net0/v4

# dladm show-phys
LINK      MEDIA      STATE    SPEED    DUPLEX    DEVICE
oldnet0   Ethernet  up       1000     full     e1000g0
net0      Ethernet  up       1000     full     nge0

```

## ▼ 如何使用动态重新配置更换网络接口卡

此过程仅适用于支持动态重新配置 (Dynamic Reconfiguration, DR) 的系统。该过程专指完成 DR 后的配置步骤。在 Oracle Solaris 11 中，不再需要在完成 DR 后重新配置网络链路。相反，您只需将已删除 NIC 的链路配置传输至更换 NIC。

该过程不对执行 DR 本身的步骤进行详细描述。有关这些步骤，请参考系统文档。

有关 DR 的介绍，请参见《Oracle Solaris 11.1 管理：设备和文件系统》中的第 4 章“动态配置设备（任务）”。

**开始之前** 执行 DR 的过程因系统类型而异。确保您首先完成以下操作：

- 确保您的系统支持 DR。
  - 参考描述您系统上的 DR 的相应手册。  
要查找有关 Oracle 提供的 Sun 服务器上的 DR 的当前文档，请在 <http://www.oracle.com/technetwork/indexes/documentation/index.html> 中搜索动态重新配置。
- 有关在 Oracle Solaris Cluster 环境中执行 DR 的信息，请参见《Oracle Solaris Cluster 系统管理指南》。

### 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 （可选的）显示有关数据链路的物理属性及其在系统上的各自位置的信息。

```
# dladm show-phys -L
```

有关由 dladm show-phys -L 显示的信息类型的更多信息，请参阅 [dladm\(1M\)](#) 手册页。

### 3 按照系统文档中所述执行 DR。

请参见您系统的 DR 文档以执行此步骤。

安装更换 NIC 后，继续执行下一步骤。

### 4 根据具体情况，执行以下步骤之一：

- 如果您将更换 NIC 插入旧 NIC 原来所在的插槽，则继续执行步骤 5。

如果新 NIC 使用旧 NIC 先前占用的位置，则新 NIC 会继承旧 NIC 的链路名称和配置。

- 如果您将更换 NIC 插入其他插槽，且新 NIC 需要继承已删除 NIC 的数据链路配置，请键入：

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

*new-datalink* 指的是位于其他插槽（不同于移除旧 NIC 的位置）中的更换 NIC 的数据链路。

*old-datalink* 指的是与移除的旧 NIC 相关联的数据链路名称。

---

注 – 在此方案中，已从中移除旧 NIC 的插槽必须留空。

---

例如，插槽 1 中的 NIC 已移除，且在插槽 2 中插入了新 NIC。插槽 1 中未插入 NIC。假定插槽 1 中的数据链路是 `net0`，插槽 2 中的数据链路是 `net1`。对于要继承旧 NIC 数据链路配置的新 NIC 的数据链路，应键入：

```
# dladm rename-link net1 net0
```

- 5 通过使新 NIC 的资源变为可供 Oracle Solaris 使用，完成 DR 过程。

例如，您可以使用 `cfgadm` 命令配置 NIC。有关更多信息，请参见 [cfgadm\(1M\)](#) 手册页。

- 6 （可选的）显示链路信息。

可以使用 `dladm show-phys` 或 `dladm show-link` 显示有关数据链路的信息。

### 示例 3-2 通过安装新的网卡执行动态重新配置

本示例显示如何使用 `e1000g` 卡更换具有链路名称 `net0` 的 `bge` 卡。在 `e1000g` 连接到系统后，`net0` 的链路配置从 `bge` 传输到 `e1000g`。

```
# dladm show-phys -L
LINK    DEVICE    LOCATION
net0    bge0     MB
net1    ibp0     MB/RISER0/PCIE0/PORT1
net2    ibp1     MB/RISER0/PCIE0/PORT2
net3    eoib2    MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

管理员执行 DR 特定步骤，如使用 `cfgadm` 移除 `bge`，然后在其位置安装 `e1000g`。安装该卡后，`e1000g0` 的数据链路会自动采用名称 `net0` 并继承链路配置。

```
# dladm show-phys -L
LINK    DEVICE    LOCATION
net0    e1000g0  MB
net1    ibp0     MB/RISER0/PCIE0/PORT1
net2    ibp1     MB/RISER0/PCIE0/PORT2
net3    eoib2    MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

```
# dladm show-link
LINK    CLASS    MTU     STATE   OVER
net0    phys     9600    up      ---
net1    phys     1500    down    ---
net2    phys     1500    down    --
net3    phys     1500    down    ---
```

## 使用 IP 接口

---

本章讨论 `ipadm` 命令并说明了如何对 IP 接口使用该命令。有关 `ipadm` 命令及其优点的概述，请参见第 13 页中的“`ipadm` 命令”。

本章包含以下主题：

- 第 35 页中的“基本 `ipadm` 命令”
- 第 37 页中的“设置 IP 接口属性”
- 第 38 页中的“设置 IP 地址属性”
- 第 39 页中的“设置 TCP/IP 协议属性”
- 第 44 页中的“监视 IP 接口和地址”

### 基本 `ipadm` 命令

在第 18 页中的“如何配置 IP 接口”中，介绍了以下三个基本 `ipadm` 子命令：

- `ipadm`
- `ipadm create-ip`
- `ipadm create-addr`

本节选择性地介绍了 `ipadm` 命令对 IP 接口的其他用法。该列表并不完整。有关 `ipadm` 命令及所有可能的子命令和选项的完整说明，请参见 [ipadm\(1M\)](#) 手册页。

### 删除 IP 接口配置 (`ipadm delete-ip`)

使用此命令可删除数据链路上已配置的 IP 接口。此命令在执行某些数据链路配置时尤为重要。例如，如果在某个数据链路上配置了 IP 接口，则重命名该数据链路将失败。必须先发出 `ipadm delete-ip`，然后才能重命名数据链路。

通常，此命令与其他 `ipadm` 和 `dladm` 子命令一起使用，例如更改系统的主接口。该任务将要求您删除接口、重命名链路，然后在重命名的数据链路上重新配置接口。序列如下所示：

```
# ipadm delete-ip interface
# dladm rename-link old-name new-name
# ipadm create-ip interface
# ipadm create-address parameters
```

请参见第 25 页中的“重命名数据链路 (`dladm rename-link`)”中有关更改主接口的示例。要在重命名数据链路后重新配置 IP 接口，请参见第 18 页中的“如何配置 IP 接口”。

## 禁用 IP 接口配置 (`ipadm disable-ip`)

缺省情况下，当您使用 `ipadm create-ip` 创建接口时，IP 接口会标记为 UP 并成为活动配置的一部分。您可以从活动配置中删除接口，无需使用 `ipadm disable-ip` 子命令销毁其配置。此命令将特定接口标记为 DOWN。

```
# ipadm disable-ip interface
```

要使 IP 接口正常运行并使其标志变为 UP，应键入：

```
# ipadm enable-ip interface
```

---

**提示** – 要显示接口的状态，请使用 `ipadm`。请参见第 45 页中的“获取有关 IP 接口的信息”。

---

## 删除接口的地址 (`ipadm delete-addr`)

此命令删除 IP 接口的特定地址配置。如果您要更改特定接口的 IP 地址，此命令很有用。您必须先删除原始地址配置，然后才能指定新的地址配置。应执行以下常规步骤：

```
# ipadm delete-addr addrobj
# ipadm create-addr parameters
```

有关为接口创建 IP 地址的示例，请参见第 18 页中的“如何配置 IP 接口”。

---

**注** – 一个接口可以有多个地址。每个地址由地址对象标识。要确保您删除的地址正确，必须知道地址对象。使用 `ipadm show-addr` 子命令可显示系统上的接口地址。有关地址对象的说明，请参见第 18 页中的“如何配置 IP 接口”。有关显示地址的更多信息，请参见第 47 页中的“获取有关 IP 地址的信息”。

---

## 设置 IP 接口属性

本节介绍如何使用 `ipadm` 命令设置选定的 IP 接口属性。

IP 接口与数据链路一样，具有您可以为特定网络环境定制的属性。对于每个接口，存在两组属性，一组适用于 IPv4 协议，另一组适用于 IPv6 协议。某些属性（如 MTU）对数据链路和 IP 接口都适用。因此，您可以有一个针对数据链路的 MTU 值和另一个针对在该链路上配置的接口的 MTU 值。而且，在该 IP 接口上，您可以有适用于 IPv4 和 IPv6 数据包的不同 MTU 值。

有以下三个用于设置 IP 接口属性的 `ipadm` 子命令：

- `ipadm show-ifprop -p property interface` 子命令，用于显示 IP 接口的属性及其当前值。如果不使用 `-p property` 选项，则会列出 IP 接口的所有属性。如果没有指定 IP 接口，则会列出所有 IP 接口的所有属性。
- `ipadm set-ifprop -p property =value interface` 子命令，用于为 IP 接口的属性指定值。
- `ipadm reset-ifprop -p property interface` 子命令将特定属性重置为其缺省值。

## 启用包转发

在网络中，一个主机可以接收发送到另一个主机系统的数据包。通过在作为接收方的本地系统中启用数据包转发，该系统可以将数据包转发到目标主机。缺省情况下，IP 转发被禁用。

包转发功能由同时可以在 IP 接口和 TCP/IP 协议上设置的属性管理。如果希望可以选择包转发方式，则启用 IP 接口上的包转发。例如，您的系统可能有多个 NIC。其中部分 NIC 连接到外部网络，其他 NIC 连接到专用网络。因此，您只对部分接口（而不是所有接口）启用数据包转发。

通过设置 TCP/IP 协议的属性，还可以全局性地在系统中启用包转发。请参见第 39 页中的“全局启用包转发”。

---

注 – IP 接口的 `forwarding` 属性与协议的该属性不是互斥的。您可以同时为接口和协议设置该属性。例如，可以在协议上全局性地启用包转发，然后在系统上定制每个 IP 接口的包转发。因此，尽管全局启用包转发，但系统仍可选择包转发。

---

要在 IP 接口上启用包转发，请使用以下命令：

```
# ipadm set-ifprop forwarding=on [-m protocol-version] interface
```

其中 `protocol-version` 为 IPv4 或 IPv6。必须对 IPv4 和 IPv6 包分别发出此命令。

以下是如何在系统上仅启用 IPv4 包转发的示例：

```
# ipadm show-ifprop -p forwarding net0
IFNAME  PROPERTY      PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net0    forwarding    ipv4   rw    off      off         off      on,off
net0    forwarding    ipv6   rw    off      --         off      on,off

# ipadm set-ifprop -p forwarding=on -m ipv4 net0
# ipadm show-ifprop net0
IFNAME  PROPERTY      PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
...
net0    forwarding    ipv4   rw    on       on         off      on,off
...
```

## 设置 IP 地址属性

`ipadm` 命令允许您在将 IP 地址分配给接口后设置特定于这些地址的属性。通过设置这些属性，您可以确定以下内容：

- 网络掩码长度
- IP 地址是否可用作传出数据包的源地址
- 地址是属于全局区域还是非全局区域
- 地址是否为专用地址

处理 IP 地址属性时可以使用以下 `ipadm` 子命令：

- `ipadm show-addrprop [-p property] [addrobj]` 子命令，用于显示地址属性，具体取决于使用的选项。  
要列出所有 IP 地址的属性，请勿指定属性或地址对象。要列出所有 IP 地址的单个属性的值，请仅指定该属性。要列出特定地址对象的所有属性，请仅指定该地址对象。
- `ipadm set-addrprop -p property =value addrobj` 子命令，用于为地址属性指定值。请注意，一次只能设置一个地址属性。
- `ipadm reset-addrprop -p property addrobj` 子命令，用于恢复地址属性的所有缺省值。

---

注 - 如果要更改特定接口的 IP 地址，请勿使用 `set-addressprop` 子命令，而是删除地址对象并使用新的 IP 地址创建新的地址对象。请参见第 36 页中的“删除接口的地址 (`ipadm delete-addr`)”。

---

例如，假设您要更改 IP 地址的网络掩码。该 IP 地址是在 IP 接口 `net3` 上配置的，由地址对象名称 `net3/v4` 标识。以下命令显示如何修改网络掩码：

```
# ipadm show-addr
ADDROBJ  TYPE      STATE  ADDR
lo0/?    static   ok     127.0.0.1/8
net3/v4   static   ok     192.168.84.3/24
```

```
# ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net3/v4 prefixlen rw 24 24 24 1-30,32

# ipadm set-addrprop -p prefixlen=8 net3/v4
# ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net3/v4 prefixlen rw 8 24 24 1-30,32
```

## 设置TCP/IP协议属性

使用 `ipadm` 命令可配置协议属性，也称为**可调参数**。`ipadm` 取代了早期发行版中常用于设置可调参数的 `ndd` 命令。

TCP/IP 属性可以是基于接口的，也可以是全局性的。属性可以应用于特定接口，也可全局性地应用于区域中的所有接口。全局属性在不同的非全局区域中可以有不同的值。有关支持的协议属性的列表，请参阅 [ipadm\(1M\)](#) 手册页。

通常情况下，TCP/IP Internet 协议的缺省值足以使网络正常工作。但是，如果缺省值对您的网络拓扑来说不够用，则可以根据需要定制这些属性。

有以下三个用于设置 TCP/IP 接口属性的 `ipadm` 子命令：

- `ipadm show-prop -p property protocol` 命令，用于显示协议的属性及其当前值。如果不使用 `-p property` 选项，则会列出协议的所有属性。如果不指定协议，则会列出所有协议的所有属性。
- `ipadm set-prop -p property =value protocol` 子命令，用于为 IP 接口的属性指定值。
- `ipadm reset-prop -p property protocol` 子命令将特定协议属性重置为其缺省值。

---

注 – 如果属性可以接收多个值，则使用 `+=` 限定符将多个值分配给属性，如下所示：

```
ipadm set-prop -p property+=value1 [value2 value3 ...].
```

要从属性的一组值中删除一个值，可以使用 `-=` 限定符，如下所示：

```
ipadm set-prop -p property-=value2
```

---

## 全局启用包转发

第 37 页中的“启用包转发”显示了如何在接口上启用包转发。通过设置 IP 接口上的包转发属性，可以有选择性地实现此功能。可以仅在系统的特定接口上启用此属性。

如果要在不考虑 IP 接口数量的情况下在整个系统上启用包转发，则使用协议属性：在协议中，属性名称与 IP 接口中的属性名称相同，均为 `forwarding`。必须分别发出命令才能在 IPv4 和 IPv6 协议上启用包转发。

以下示例显示如何在系统上对所有IPv4和IPv6通信启用包转发：

```
# ipadm show-prop -p forwarding ip
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv4 forwarding rw off -- off on,off
ipv6 forwarding rw off -- off on,off
#
# ipadm set-prop -p forwarding=on ipv4
# ipadm set-prop -p forwarding=on ipv6
#
# ipadm show-prop ip
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv4 forwarding rw on on off on,off
ipv4 ttl rw 255 -- 255 1-255
ipv6 forwarding rw on on off on,off
ipv6 hoplimit rw 255 -- 255 1-255#
```

注 - IP接口的 `forwarding` 属性与协议的该属性不是互斥的。您可以同时为接口和协议设置该属性。例如，可以在协议上全局性地启用包转发，然后在系统上定制每个IP接口的包转发。因此，尽管全局启用包转发，但系统仍可选择包转发。

## 设置特权端口

在传输协议（如TCP、UDP和SCTP）上，端口1-1023是缺省的特权端口，只有使用root权限运行的进程才可以绑定到这些端口。通过使用 `ipadm` 命令，您可以将超出此给定缺省范围的一个端口保留为特权端口。因此，只有root进程可以绑定到该端口。要设置特权端口，请定制以下传输协议属性：

- `smallest_nonpriv_port`—该属性的值指示一般用户可以绑定到的端口号范围。如果指定的端口在此范围内，则可以将其设置为特权端口。使用 `ipadm show-prop` 命令可显示属性的值。
- `extra_priv_ports`—该属性指定哪些端口为特权端口。使用 `ipadm set-prop` 子命令可指定要限制的端口。可以为此属性指定多个值。

例如，假设要将TCP端口3001和3050设置为特权端口，并且仅限root用户访问。`smallest_nonpriv_port`属性指示1024是非特权端口的最小端口号。因此，指定的端口3001和3050可以更改为特权端口。应发出类似于以下内容的命令以继续：

```
# ipadm show-prop -p smallest_nonpriv_port tcp
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp smallest_nonpriv_port rw 1024 -- 1024 1024-32768

# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp extra_priv_ports rw 2049,4045 -- 2049,4045 1-65535

# ipadm set-prop -p extra_priv_ports+=3001 tcp
# ipadm set-prop -p extra_priv_ports+=3050 tcp
# ipadm show-prop -p extra_priv_ports tcp
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
tcp	extra_priv_ports	rw	2049,4045 3001,3050	3001,3050	2049,4045	1-65535

要删除其中一个端口（如 4045）以避免成为特权端口，应键入以下命令：

```
# ipadm set-prop -p extra_priv_ports=-4045 tcp
# ipadm show-prop -p extra_priv_ports tcp
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
tcp	extra_priv_ports	rw	2049,3001 3050	3001,3050	2049,4045	1-65535

## 在多宿主主机上实现对称路由

缺省情况下，具有多个接口的系统（也称为**多宿主主机**）基于路由表中到通信流的目标的最长匹配路由来路由其网络通信流。如果存在多个长度相等的到目标的路由路径，Oracle Solaris 将应用等成本多路径 (Equal-Cost Multi-Path, ECMP) 算法将通信流量分配到这些路由。

在某些情况下，以这种方式分配通信流量并不合适。可能通过多宿主主机上的一个接口发送了 IP 包，但是此主机与该包中的 IP 源地址不位于同一子网上。此外，如果传出数据包是对某传入请求（例如 ICMP 回显请求）的响应，则请求和响应可能不经过相同的接口。这种通信路由配置称为**非对称路由**。如果您的 Internet 服务提供商按照 RFC 3704 (<http://rfc-editor.org/rfc/bcp/bcp84.txt>) 中所述实现准入过滤，则非对称路由配置可能导致提供商丢弃传出数据包。

RFC 3704 旨在限制拒绝服务攻击在 Internet 上扩散。为了与它保持一致，必须将您的网络配置为对称路由。在 Oracle Solaris 中，可以通过使用 IP `hostmodel` 属性满足这一要求。此属性控制通过多宿主主机接收或传送 IP 数据包的行为。

`hostmodel` 属性可以具有以下三个可能值之一：

<code>strong</code>	对应于 RFC 1122 中定义的强端系统 (end system, ES) 模型。此值实现对称路由。
<code>weak</code>	对应于 RFC 1122 中定义的弱 ES 模型。在采用此值的情况下，多宿主主机使用非对称路由。
<code>src-priority</code>	通过使用首选路由配置数据包路由。如果路由表中存在多个目标路由，则使用在其上配置传出数据包的 IP 源地址的接口的路由，就是首选路由。如果不存在这样的路由，则传出数据包将使用到数据包包的 IP 目标的最长匹配路由。

以下示例显示了如何在多宿主主机上实现 IP 包的对称路由。

```
# ipadm set-prop -p hostmodel=strong ip
# ipadm show-prop -p hostmodel ip
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
ipv6	hostmodel	rw	strong	--	weak	strong,

```

src-priority,
weak
strong,
src-priority,
weak

```

## 实施通信拥塞控制

当节点发送的包数超过网络可以容纳的量时，通常会出现路由器缓冲区溢出形式的网络拥塞。各种算法通过对发送系统实施控制来防止通信拥塞。下面的算法在 Oracle Solaris 中受支持，可以轻松地添加或直接插入到操作系统中。

下表列出和说明了支持的算法。

算法	Oracle Solaris 名称	说明
NewReno	newreno	Oracle Solaris 中的缺省算法。控制机制包括发送者的拥塞窗口、慢速启动和拥塞避免。
HighSpeed	highspeed	针对高速网络的新 Reno 的最广为人知和最简单的修改版之一。
CUBIC	cubic	Linux 2.6 中当前的缺省算法。将拥塞避免阶段从线性窗口增加更改为 cubic 函数。
Vegas	vegas	基于延迟的经典算法，该算法尝试预测拥塞而不会引起实际包损失。

通过设置以下控制相关的 TCP 属性来启用拥塞控制。尽管下面列出的属性为 TCP 的属性，但由这些属性启用的控制机制也适用于 SCTP 通信。

- `cong_enabled`—包含系统中当前运行的算法的列表（以逗号分隔）。您可以添加或删除相应算法以仅启用要使用的那些算法。此属性可以有多个值。因此，必须使用 `+=` 限定符或 `-=` 限定符，具体取决于希望生效的更改。
- `cong_default`—当应用程序未在套接字选项中显式指定算法时，此算法为缺省使用的算法。当前，`cong_default` 属性的值同时适用于全局和非全局区域。

要将用于实施拥塞控制的算法添加到协议中，请发出以下命令：

```
# ipadm set-prop -p cong_enabled+=algorithm tcp
```

要删除算法，请发出以下命令：

```
# ipadm set-prop -p cong_enabled-=algorithm tcp
```

要替换缺省算法，请发出以下命令：

```
# ipadm set-prop -p cong_default=algorithm tcp
```

注 - 添加或删除算法时不遵循任何序列规则。您可以先删除属性的算法，然后再将其他算法添加到该属性。但是，`cong_default` 属性必须始终具有已定义的算法。

以下示例显示了实施拥塞控制可能需要执行的步骤。在此示例中，将TCP协议的缺省算法从 `newreno` 更改为 `cubic`。然后，从已启用的算法列表中删除 `vegas` 算法。

```
# ipadm show-prop -p cong_default,cong_enabled tcp
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp cong_default rw newreno -- newreno -
tcp cong_enabled rw newreno,cubic, -- newreno newreno,cubic,
highspeed, highspped,vegas
vegas

# ipadm set-prop -p cong_enabled=vegas tcp
# ipadm set-prop -p cong_default=cubic tcp

# ipadm show-prop -p cong_default,cong_enabled tcp
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp cong_default rw cubic -- newreno -
tcp cong_enabled rw newreno,cubic, -- newreno newreno,cubic,
highspeed highspped
```

## 更改TCP接收缓冲区大小

TCP接收缓冲区大小是通过使用TCP属性 `recv_buf` 来设置的，缺省情况下为128 KB。但是，应用程序没有均匀使用可用带宽。因此，连接延迟可能会要求您更改缺省大小。例如，使用Oracle Solaris的安全Shell功能会产生带宽使用开销，因为有其他校验和与加密进程在数据流上执行。因此，可能需要增加缓冲区大小。同样，为使执行批量传输的应用程序能有效使用带宽，也需要进行相同的缓冲区大小调整。

通过估算**带宽延迟乘积** (bandwidth delay product, BDP)，可以计算要使用的正确接收缓冲区大小，如下所示：

$$BDP = \text{available\_bandwidth} * \text{connection-latency}$$

使用 `ping -s host` 可获取连接延迟的值。使用 `iperf` 和 `iperf` 工具可估计带宽使用情况。

合适的接收缓冲区大小接近BDP的值。但请注意，带宽使用情况还取决于各种条件。共享基础结构或争用带宽的应用程序和用户的数量可能会影响该估算值。

要更改缓冲区大小的值，请使用以下语法：

```
# ipadm set-prop -p recv_buf=value tcp
```

以下示例显示了如何将缓冲区大小增加到164 KB

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp  recv_buf  rw  128000  --  128000  2048-1048576
```

```
# ipadm set-prop -p recv_buf=164000 tcp
```

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp  recv_buf  rw  164000  --  164000  2048-1048576
```

最好不要设置缓冲区大小的值，因为首选大小在不同的情况下有所不同。考虑以下针对每种网络在各自特定条件下显示不同 BDP 值的情况：

典型的 1 Gbps LAN，其中缓冲区的缺省大小是 128 KB：

$$\text{BDP} = 128 \text{ MBps} * 0.001 \text{ s} = 128 \text{ kB}$$

理论上的 1Gbps WAN，延迟为 100 毫秒：

$$\text{BDP} = 128 \text{ MBps} * 0.1 \text{ s} = 12.8 \text{ MB}$$

欧洲到美国链路（带宽由 uperf 测量）

$$\text{BDP} = 2.6 \text{ MBps} * 0.175 = 470 \text{ kB}$$

如果无法计算 BDP，请使用以下观测值作为指导：

- 对于通过 LAN 的批量传输，缓冲区大小缺省值 128 KB 已够用。
- 对于大部分 WAN 部署，接收缓冲区大小应在 2 MB 范围内。




---

注意 - 增加 TCP 接收缓冲区大小的同时会增加许多网络应用程序的内存资源占用。

---

## 监视 IP 接口和地址

使用 ipadm 命令可监视并获取有关 IP 接口及其属性的信息。单独使用此命令可显示有关系统上 IP 接口的常规信息。但是，也可以使用子命令来限制要使用以下语法显示的信息：

```
ipadm show-* [other-arguments] [interface]
```

- 要仅获取接口信息，请使用 ipadm show-if。
- 要仅获取地址信息，请使用 ipadm show-addr。
- 要获取有关接口属性的信息，请使用 ipadm show-ifprop。
- 要获取有关地址属性的信息，请使用 ipadm show-addrprop。

本节提供了一些示例，用于说明如何使用 ipadm 子命令获取接口信息。有关通过 ipadm show-\* 命令显示的所有字段的说明，请参阅 [ipadm\(1M\)](#) 手册页。

## 获取有关 IP 接口的常规信息

仅使用 `ipadm` 命令而不附带任何子命令可提供有关所有系统的 IP 接口的缺省信息。例如：

```
# ipadm
NAME          CLASS/TYPE STATE   UNDER ADDR
lo0           loopback  ok      --    --
  lo0/v4      static   ok      --    127.0.0.1/8
  lo0/v6      static   ok      --    ::1/128
net0          ip        ok      --    --
  net0/v4     static   ok      --    10.132.146.233/23
  net0/v4     dhcp    ok      --    10.132.146.234/23
ipmp0        ipmp      degraded --    --
  ipmp0/v6    static   ok      --    2001:db8:1:2::4c08/128
net1         ip        failed  ipmp0 --
  net1/v6     addrconf ok      --    fe80::124:4fff:fe58:1831/10
net2         ip        ok      ipmp0 --
  net2/v6     addrconf ok      --    fe80::214:4fff:fe58:1832/10
iptun0       ip        ok      --    --
  iptun0/v4   static   ok      --    172.16.111.5->172.16.223.75
  iptun0/v6   static   ok      --    fe80::10:5->fe80::223:75
  iptun0/v6a  static   ok      --    2001:db8:1a0:7::10:5->2001:db8:7a82:64::223:75
```

样例输出提供以下信息：

- IP 接口。
- 每个接口的类。
- 每个接口的状态。
- 接口的状态可以是“独立”IP 接口，也可以是底层接口（对于另一种接口配置类型）。在该示例中，`net1` 和 `net2` 是 `ipmp0` 的底层接口，如 `UNDER` 列所指示。
- 与接口相关联的地址对象。地址对象标识特定 IP 地址。这些地址对象列在 `NAME` 标题下并进行了缩进，以便与接口名称进行区分。
- IP 地址的类型，已在 `CLASS/TYPE` 标题下进行了缩进，可以是 `static`、`dhcp` 等。
- 实际地址，在 `ADDRESS` 列下列出。

因此，通过 `ipadm` 命令可以全面了解系统的接口。

## 获取有关 IP 接口的信息

有关 IP 接口的信息，请使用 `ipadm show-if [interface]` 子命令。如果您不指定接口，则会获得系统上所有接口的信息。

命令输出中各字段的含义如下：

`IFNAME` 指显示其信息的接口。

`CLASS` 指接口的类，可以是四项之一：

- ip 指 IP 接口
- ipmp 指 IPMP 接口
- vni 指虚拟接口
- loopback 指回送接口，它是自动创建的。除了回送接口，您可以手动创建其余 3 个接口类。

STATE 指接口的状态，可以是以下状态之一：ok、offline、failed、down 或 disabled。

状态 failed（故障）适用于 IPMP 组，可以指出现故障而无法承载通信流量的数据链路或 IP 接口。如果 IP 接口属于 IPMP 组，则 IPMP 接口可以继续使用组中其他活动 IP 接口接收和发送通信流量。

状态 down（关闭）指由管理员切换为脱机模式的 IP 接口。

状态 disable（已禁用）指通过使用 ipadm disable-if 命令取消激活的 IP 接口。

ACTIVE 指示接口是否用于承载通信，并且设置为 yes 或 no。

OVER 仅适用于 IPMP 接口类，指构成 IPMP 接口或组的底层接口。

以下是命令提供的信息的示例：

```
# ipadm show-if
IFNAME      CLASS      STATE      ACTIVE      OVER
lo0         loopback  ok         yes         --
net0        ip         ok         yes         --
net1        ip         ok         yes         --
tun0        ip         ok         yes         --
```

## 获取有关 IP 接口属性的信息

使用 ipadm show-ifprop [*interface*] 命令可获取有关 IP 接口属性的信息。如果不指定属性或接口，则提供有关系统上所有 IP 接口的所有属性的信息。

命令输出中各字段的含义如下：

- IFNAME 指显示其信息的 IP 接口。
- PROPERTY 指接口的属性。一个接口可以有几个属性。
- PROTO 指属性所适用的协议，可以是 IPv4 或 IPv6。
- PERM 指给定属性的允许的权限，可以是只读、只写或读写。
- CURRENT 指活动配置中属性的当前值。
- PERSISTENT 指系统重新引导时重新应用的属性值。

DEFAULT	指示指定属性的缺省值。
POSSIBLE	指可以分配给指定属性的值的列表。对于数字值，显示一个可接受值的范围。

注 – 如果任何字段值是未知的，例如，一个接口不支持正在请求其信息的属性，则值显示为问号(?)。

以下是 `ipadm show-ifprop` 子命令提供的信息的示例：

```
# ipadm show-ifprop -p mtu net1
IFNAME  PROPERTY  PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net1    mtu       ipv4   rw    1500     --          1500     68-1500
net1    mtu       ipv6   rw    1500     --          1500     1280-1500
```

## 获取有关 IP 地址的信息

有关 IP 地址的信息，请使用 `ipadm show-addr [interface]` 子命令。如果不指定接口，则提供有关系统上所有 IP 地址的信息。

命令输出中各字段的含义如下：

ADDROBJ	指定列出其 IP 地址的地址对象。
TYPE	指示 IP 地址为 <code>static</code> 、 <code>dhcp</code> 或 <code>addrconf</code> 。 <code>addrconf</code> 值指示地址是使用无状态还是有状态地址配置获得的。
STATE	介绍了活动配置中地址对象的状态。有关这些值的完整列表，请参见 <a href="#">ipadm(1M)</a> 手册页。
ADDR	指定在接口上配置的 IP 地址。地址可以是 IPv4 或 IPv6 地址。隧道接口同时显示本地地址和远程地址。

有关隧道的更多信息，请参见《[配置和管理 Oracle Solaris 11.1 网络](#)》中的第 6 章“配置 IP 隧道”。

以下是 `ipadm show-addr` 子命令提供的信息的示例：

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
net0/v4      static    ok         192.168.84.3/24
tun0/v4      static    ok         172.16.134.1-->172.16.134.2
```

如果使用命令指定一个接口并且该接口具有多个地址，则显示类似于以下内容的信息：

```
# ipadm show-addr net0
ADDROBJ      TYPE      STATE      ADDR
net0/v4      static    ok         192.168.84.3/24
net0/v4a     static    ok         10.0.1.1/24
net0/v4bc    static    ok         172.16.10.1
```

列出为 *interface/?* 的地址对象指示地址在接口上由不使用 libipadm API 的应用程序配置。此类应用程序不受 ipadm 命令控制，该命令要求地址对象名称使用的格式为 *interface/ user-defined-string*。有关指定 IP 地址的示例，请参见第 18 页中的“如何配置 IP 接口”。

## 获取有关 IP 地址属性的信息

有关 IP 地址属性的信息，请参见 ipadm show-addrprop [*addrobj*] 子命令。要列出所有属性，请省略 *addrobj* 选项。要列出所有 IP 地址的单个属性，请仅指定该属性。要列出特定地址的所有属性，请仅指定 *addrobj* 选项。

命令输出中各字段的含义如下：

- ADDROBJ            指列出其属性的地址对象。
- PROPERTY          指地址对象的属性。一个地址对象可以有几个属性。
- PERM              指给定属性的允许的权限，可以是只读、只写或读写。
- CURRENT          指当前配置中属性的实际值。
- PERSISTENT        指系统重新引导时重新应用的属性值。
- DEFAULT          指示指定属性的缺省值。
- POSSIBLE          指可以分配给指定属性的值的列表。对于数字值，显示一个可接受值的范围。

以下是 ipadm show-addrprop 子命令提供的信息的示例：

```
# ipadm show-addrprop net1/v4
ADDROBJ  PROPERTY  PERM  CURRENT          PERSISTENT  DEFAULT          POSSIBLE
net1/v4  broadcast r-    192.168.84.255  --          192.168.84.255  --
net1/v4  deprecated rw     off            --          off             on,off
net1/v4  prefixlen rw     24            24          24              1-30,32
net1/v4  private   rw     off            --          off             on,off
net1/v4  transmit  rw     on             --          on              on,off
net1/v4  zone      rw     global         --          global          --
```

# 在运行 Oracle Solaris 的手提电脑上配置无线联网

IEEE 802.11 规范定义局域网的无线通信。这些规范及其描述的网络统称为 *WiFi*，该术语是 Wi-Fi 联盟贸易组注册的商标。提供商和潜在客户都可以轻松配置 WiFi 网络。因此，WiFi 网络日益普及，在世界各地广泛使用。WiFi 网络与手提电话、电视和收音机使用相同的无线电波技术。

注 - Oracle Solaris 不提供配置 WiFi 服务器或访问点的功能。

本章包含以下主题：

- 第 49 页中的“WiFi 通信任务列表”
- 第 54 页中的“安全 WiFi 通信”

## WiFi 通信任务列表

任务	说明	参考
连接到 WiFi 网络	设置和建立与本地 WiFi 网络的通信。	第 50 页中的“如何连接到 WiFi 网络”
监视 WiFi 链路上的通信。	使用标准的 Oracle Solaris 联网工具检查 WiFi 链路的状况。	第 53 页中的“如何监视 WiFi 链路”
建立安全的 WiFi 通信。	创建有线对等保密 (Wired Equivalent Privacy, WEP) 密钥并使用该密钥建立与安全 WiFi 网络的连接。	第 55 页中的“如何设置加密的 WiFi 网络连接”

## ▼ 如何连接到 WiFi 网络

开始之前 执行以下步骤将手提电脑连接到 WiFi 网络。

### 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 显示数据链路的物理属性。

```
# dladm show-phys
LINK          MEDIA          STATE  SPEED  DUPLEX  DEVICE
net0          Ethernet      up     1500   full    ath0
net1          Ethernet      up     1500   full    e1000g0
```

在本示例中，输出指示两个链路可用。通过设备 `ath0` 的 `Net0` 链路支持 WiFi 通信。`e1000g0` 链路用于将系统连接到有线网络。

### 3 配置 WiFi 接口。

使用以下步骤配置接口：

#### a. 创建支持 WiFi 的接口：

```
# ipadm create-ip net0
```

#### b. 验证链路已激活：

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

### 4 检查可用网络。

```
# dladm scan-wifi
LINK  ESSID          BSSID/IBSSID  SEC  STRENGTH  MODE  SPEED
net0  ofc             00:0e:38:49:01:d0  none  good      g     54Mb
net0  home           00:0e:38:49:02:f0  none  very weak g     54Mb
net0  linksys        00:0d:ed:a5:47:e0  none  very good g     54Mb
```

`scan-wifi` 命令的示例输出显示有关在当前位置可用的 WiFi 网络的信息。输出中的信息包括：

LINK 请参阅 WiFi 连接中使用的链路名称。

ESSID 请参阅扩展服务集 ID。ESSID 是 WiFi 网络的名称，该名称可由特定无线网络的管理员随机指定。

BSSID/IBSSID 请参阅基本服务集 ID，即特定 ESSID 的唯一标识符。BSSID 是服务于具有特定 ESSID 的网络的附近访问点的 48 位 MAC 地址。

SEC	请参阅访问网络所需的安全类型。值是 none（无）或 WEP。有关 WEP 的信息，请参阅第 54 页中的“安全 WiFi 通信”。
STRENGTH	请参阅您位置处可用的 WiFi 网络发出的无线电信号的强度。
MODE	请参阅网络运行的 802.11 协议的版本。模式有 a、b、g 或这些模式的组合。
SPEED	请参阅特定网络的速度（以兆位/秒为单位）。

## 5 连接到 WiFi 网络。

执行以下任一操作：

- 连接到信号最强的不安全 WiFi 网络。

```
# dladm connect-wifi
```

- 通过指定其 ESSID 连接到不安全的网络。

```
# dladm connect-wifi -e ESSID
```

dladm 的 connect-wifi 子命令有几个用于连接到 WiFi 网络的更多选项。有关完整的详细信息，请参阅 [dladm\(1M\)](#) 手册页。

## 6 配置接口的 IP 地址。

执行以下任一操作：

- 从 DHCP 服务器获取 IP 地址。

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

如果 WiFi 网络不支持 DHCP，您将收到以下消息：

```
ipadm: interface: interface does not exist or cannot be managed using DHCP
```

- 配置静态 IP 地址：

如果您的系统有专用的 IP 地址，请使用此选项。

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

## 7 检查系统所连接到的 WiFi 网络的状态。

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE  SPEED
net0      connected   ofc        none    very good g      36Mb
```

在此示例中，输出指示系统现已连接到 ofc 网络。先前步骤 4 中的 scan-wifi 输出指示 ofc 在可用网络中信号最强。dladm connect-wifi 命令会自动选择信号最强的 WiFi 网络，除非您直接指定其他网络。

## 8 通过 WiFi 网络访问 Internet。

根据系统所连接到的网络，执行以下任一操作：

- 如果访问点提供免费服务，您现在可以运行浏览器或您选择的应用程序。
- 如果访问点位于需要付费的商业 WiFi 网络中，请遵循当前位置提供的说明进行操作。通常情况下，您需要运行浏览器、提供密钥和向网络提供商提供信用卡信息。

## 9 结束会话。

执行以下操作之一：

- 终止 WiFi 会话，但保持系统运行。
 

```
# dladm disconnect-wifi
```
- 当前正在运行多个会话时，终止特定的 WiFi 会话。

```
# dladm disconnect-wifi link
```

其中，*link* 代表要用于会话的接口。

- 在 WiFi 会话正在运行时，正常关闭系统。

```
# shutdown -g0 -i5
```

在通过 `shutdown` 命令关闭系统之前，您不需要显式断开 WiFi 会话连接。

### 示例 5-1 连接到特定 WiFi 网络

以下示例组合了要将 Oracle Solaris 手提电脑连接到无线网络所需执行的各个步骤。该示例还显示了如何强制系统连接到特定的首选无线网络，而不是允许 OS 随机选择无线网络。在该示例中，假定您已在手提电脑上配置静态 IP 地址 10.192.16.3/24。该示例从确定 WiFi 链路的可用性开始。

```
# dladm show-phys
LINK          MEDIA          STATE  SPEED  DUPLEX  DEVICE
net0          Ethernet      up     1500   full    ath0
net1          Ethernet      up     1500   full    e1000g0

# ipadm create-ip net0
IFNAME      CLASS  STATE  ACTIVE  OVER
lo0         loopback  ok     yes     --
net0        ip       ok     yes     --

# dladm scan-wifi
LINK  ESSID          BSSID/IBSSID  SEC  STRENGTH  MODE  SPEED
net0  wifi-a         00:0e:38:49:01:d0  none  weak      g     54Mb
net0  wifi-b         00:0e:38:49:02:f0  none  very weak g     54Mb
net0  ofc-net        00:0d:ed:a5:47:e0  wep   very good g     54Mb
net0  citinet        00:40:96:2a:56:b5  none  good      b     11Mb

# dladm connect-wifi -e citinet

# ipadm create-addr -a 10.192.16.3/24 net0
ipadm: net0/v4
# ipadm show-addr net0
ADDROBJ      TYPE  STATE  ADDR
```

```
net0/v4          static  ok          10.192.16.3/24

# dladm show-wifi
LINK            STATUS      ESSID       SEC          STRENGTH     MODE        SPEED
net0            connected  citinet     none         good         g           11Mb
```

运行浏览器或其他应用程序，开始通过 WiFi 网络进行工作。

```
# firefox
```

此时将显示 Firefox 浏览器的主页。

终止会话但使手提电脑保持运行。

```
# dladm disconnect-wifi
# dladm show-wifi
LINK            STATUS      ESSID       SEC          STRENGTH     MODE        SPEED
net0            disconnected --          --          --          --          --
```

show-wifi 的输出确认您已从 WiFi 网络断开 net0 链路的连接。

## ▼ 如何监视 WiFi 链路

此过程介绍如何通过标准联网工具监视 WiFi 链路的状况以及如何通过 linkprop 子命令更改选择的链路属性。

### 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 连接到 WiFi 网络，如第 50 页中的“如何连接到 WiFi 网络”中所述。

### 3 查看链路的属性。

使用以下语法：

```
# dladm show-linkprop link
```

例如，您将使用以下语法来显示通过 net0 无线链路建立的连接的状态：

```
# dladm show-linkprop net0
...
PROPERTY      VALUE          DEFAULT        POSSIBLE
channel       5              --             --
powermode     off            off            off,fast,max
radio         ?              on             on,off
speed        36            --             1,2,5.5,6,9,11,12,18,24,36,48,54
...
```

### 4 为链路设置固定的速度。



注意 – Oracle Solaris 自动为 WiFi 连接选择最佳速度。修改链路的初始速度可能导致性能降低，或导致无法建立特定 WiFi 连接。

您可以将链路速度修改为 `show-linkprop` 输出中列出的速度值之一。

```
# dladm set-linkprop -p speed=value link
```

## 5 检查通过链路的包流。

```
# netstat -I net0 -i 5
      input  net0      output      input (Total)      output
packets errs  packets errs  colls  packets errs  packets errs  colls
317    0    106    0    0    2905    0    571    0    0
14     0     0     0    0     20     0     0     0    0
7      0     0     0    0     16     0     1     0    0
5      0     0     0    0     9      0     0     0    0
304    0    10     0    0     631    0    316    0    0
338    0     9     0    0     722    0    381    0    0
294    0     7     0    0     670    0    371    0    0
306    0     5     0    0     649    0    338    0    0
289    0     5     0    0     597    0    301    0    0
```

### 示例 5-2 设置链路的速度

本示例说明在连接到 WiFi 网络之后如何设置链路的速度。

```
# dladm show-linkprop -p speed net0
PROPERTY      VALUE      DEFAULT      POSSIBLE
speed         24         --           1,2,5,6,9,11,12,18,24,36,48,54
# dladm set-linkprop -p speed=36 net0
```

```
# dladm show-linkprop -p speed net0
PROPERTY      VALUE      DEFAULT      POSSIBLE
speed         36         --           1,2,5,6,9,11,12,18,24,36,48,54
```

## 安全 WiFi 通信

无线电波技术使 WiFi 网络随时可用，而且经常供许多位置的用户免费访问。因此，连接到 WiFi 网络可能会带来安全风险。但是，某些类型的 WiFi 连接比较安全：

- 连接到专用的、限制访问的 WiFi 网络
  - 专用网络（如公司或大学建立的内联网）要求只有提供正确的安全质询的用户才能访问网络。潜在用户在连接序列或通过安全 VPN 登录到网络的过程中必须提供密钥。
- 加密与 WiFi 网络的连接

可以使用安全密钥对您的系统与 WiFi 网络之间的通信加密。您的 WiFi 网络访问点必须是您的家庭或办公室中具有安全密钥生成功能的路由器。在创建安全连接之前，您的系统和路由器首先建立密钥，然后共享密钥。

`dladm` 命令可以使用有线对等保密 (Wired Equivalent Privacy, WEP) 密钥对通过访问点的连接加密。WEP 协议在针对无线连接的 IEEE 802.11 规范中定义。有关 `dladm` 命令的与 WEP 相关的选项的完整详细信息，请参阅 [dladm\(1M\)](#) 手册页。

## ▼ 如何设置加密的 WiFi 网络连接

下一过程说明如何设置系统与家用路由器之间的安全通信。许多家用无线和有线路由器具有加密功能，可以生成安全密钥。

**开始之前** 如果您要连接到自己家的无线网络，请确保已配置路由器且已生成 WEP 密钥。按照路由器制造商的文档生成和保存密钥配置。

### 1 成为管理员。

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

### 2 创建一个包含 WEP 密钥的安全对象。

在系统上打开一个终端窗口并键入以下内容：

```
# dladm create-secobj -c wep keyname
```

其中 *keyname* 代表您要给予密钥的名称。

### 3 将 WEP 密钥的值提供给安全对象。

然后，`create-secobj` 子命令运行一个请求密钥值的脚本。

```
provide value for keyname: 5-or-13-byte key
confirm value for keyname: Retype key
```

此值就是路由器生成的密钥。该脚本接受一个包含 5 个字节或 13 个字节的字符串，它是密钥值的 ASCII 或十六进制表示形式。

### 4 查看您刚刚创建的密钥的内容。

```
# dladm show-secobj
OBJECT          CLASS
keyname         wep
```

其中 *keyname* 是安全对象的名称。

### 5 建立与 WiFi 网络的加密连接。

```
# dladm connect-wifi -e network -k keyname interface
```

## 6 验证连接是安全的。

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE  SPEED
net0      connected   wifi-1     wep      good      g     11Mb
```

SEC 下的 wep 值指示连接的 WEP 加密已就绪。

### 示例 5-3 设置加密的 WiFi 通信

本示例假定您已经完成以下内容：

- 将您的系统连接到了可以创建 WEP 密钥的家用路由器
- 按照路由器制造商的文档创建了 WEP 密钥
- 保存了密钥，以便使用它在您的系统上创建安全对象

创建安全对象。

```
# dladm create-secobj -c wep mykey
provide value for mykey: *****
confirm value for mkey: *****
```

当您提供由路由器生成的 WEP 密钥时，您键入的值显示为星号。

```
# dladm show-secobj
OBJECT      CLASS
mykey      wep
# dladm connect-wifi -e citinet -k mykey net0
```

以上命令通过使用安全对象 mykey 建立到 WiFi 网络 citinet 的加密连接。

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE  SPEED
net0      connected   citinet    wep      good      g     36Mb
```

此输出验证您已通过 WEP 加密连接到 citinet。

## 对应关系比较：ifconfig 和 ipadm 命令

在配置网络接口方面，ipadm 命令已取代 ifconfig 命令。尽管 ifconfig 命令在 Oracle Solaris 11 中仍起作用，但是 ipadm 命令是网络配置的首选工具。不过，某些 ifconfig 选项在 ipadm 子命令中没有等效项。下表选择性地列出了 ifconfig 命令的部分命令选项及其在 ipadm 命令中的等效项。

注 - 此表未提供 ipadm 选项的完整列表。有关完整列表，请参见 [ipadm\(1M\)](#) 手册页。

表 A-1 ifconfig 和 ipadm 命令之间的语法对应关系

ifconfig 命令	ipadm 命令
plumb/unplumb	ipadm create-ip ipadm create-vni ipadm create-ipmp ipadm enable-addr ipadm delete-ip ipadm delete-vni ipadm delete-ipmp ipadm disable-addr
[address[/prefix-length] [dest-address]] [addif address[ prefix-length]] [removeif address[ prefix-length]][netmask mask][destination dest-address]{auto-dhcp dhcp}[primary][wait seconds]extend   release   start	ipadm create-addr ipadm create-addr -T dhcp ipadm create-addr -T addrconf ipadm delete-addr ipadm refresh-addr

表 A-1 ifconfig 和 ipadm 命令之间的语法对应关系 (续)

ifconfig 命令	ipadm 命令
[deprecated   -deprecated] [preferred   -preferred] [private   -private] [zone <i>zonename</i>   -zones   -all-zones][xmit   -xmit]	ipadm set-addprop ipadm reset-addprop ipadm show-addprop
up	ipadm up-addr
down	ipadm down-addr
[metric <i>n</i> ] [mtu <i>n</i> ] [nud   -nud] [arp   -arp] [usesrc [ <i>name</i>   none] [router   -router]	ipadm set-ifprop ipadm show-ifprop ipadm reset-ifprop
[ipmp] [group [ <i>name</i>   ""]] standby   -standby] [failover   -failover]	ipadm create-ipmp ipadm delete-ipmp  ipadm add-ipmp ipadm remove-ipmp  ipadm set-ifprop -p [standby] [group]
[ <i>interface</i> ] [-a]	ipadm  ipadm show-if  ipadm show-addr
[ <i>tdst tunnel-dest-addr</i> ] [ <i>tsrc tunnel-srcs-addr</i> ] [encaplimit <i>n</i> ] -encaplimit] [thoplimit <i>n</i> ]	dladm *-iptun 命令集。有关更多详细信息，请参见 <a href="#">dladm(1M)</a> 手册页和《配置和管理 Oracle Solaris 11.1 网络》中的“使用 dladm 命令进行隧道配置和管理”。
[auth_algs <i>authentication algorithm</i> ] [encr_algs <i>encryption algorithm</i> ] [encr_auth_algs <i>encryption authentication algorithm</i> ]	ipseccnf 有关详细信息，请参见 <a href="#">ipseccnf(1M)</a> 手册页和《在 Oracle Solaris 11.1 中保护网络安全》中的第 7 章“配置 IPsec (任务)”。
[auth_revprop] [ether [ <i>address</i> ]] [index <i>if-index</i> ] [subnet <i>subnet-address</i> ] [broadcast <i>broadcast-address</i> ] [token <i>address /prefix-length</i> ]  DHCP 选项 -inform, ping, release, status, drop	当前未提供等效子命令。
[modlist] [modinsert <i>mod_name@pos</i> ] [modremove <i>mod_name@pos</i> ]	当前未提供等效子命令。

## 对应关系比较：nnd 和 ipadm 命令

---

在定制网络参数或可调参数方面，ipadm 命令已取代 nnd 命令。尽管 nnd 命令在 Oracle Solaris 11 中仍起作用，但是 ipadm 命令是定制网络参数的首选工具。不过，某些 nnd 选项没有等效的 ipadm 子命令。下表选择性地列出了 nnd 命令的部分命令选项及其在 ipadm 命令中的等效项。

---

注 - 此表未提供 ipadm 选项的完整列表。有关完整列表，请参见 [ipadm\(1M\)](#) 手册页。

---

表 B-1 nnd 和 ipadm 命令之间的语法对应关系：检索属性

nnd 命令	ipadm 命令
<pre>bash-3.2# nnd -get /dev/ip ? ip_def_ttl      (read and write) ip6_def_hops    (read and write) ip_forward_directed_broadcasts                 (read and write) ip_forwarding   (read and write) ... ...</pre>	<pre>bash-3.2# ipadm show-prop ip PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4 forwarding  rw    off      --        off         on,off ipv4 ttl         rw    255     --        255         1-255 ipv6 forwarding  rw    off      --        off         on,off ipv6 hoplimit    rw    255     --        255         1-255 ...</pre>
<pre>bash-3.2# nnd -get /dev/ip \ ip_def_ttl 100 bash-3.2# nnd -get /dev/ip \ ip6_def_hops 255</pre>	<pre>bash-3.2# ipadm show-prop -p ttl,hoplimit ip PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4 ttl         rw    255     --        255         1-255 ipv6 hoplimit    rw    255     --        255         1-255</pre>
<pre>bash-3.2# nnd -get /dev/tcp ? tcp_cwnd_max    (read and write) tcp_strong_iss  (read and write) tcp_time_wait_interval                 (read and write) tcp_tstamp_always (read and write) tcp_tstamp_if_wscale                 (read and write) ... ...</pre>	<pre>bash-3.2# ipadm show-prop tcp PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE tcp ecn         rw    passive --        passive  never,passive, active tcp extra_      rw    2049    2049,4045  2049,4045  1-65535 priv_ports tcp largest_    rw    65535  --        65535     1024-65535 anon_port tcp recv_      rw    128000 --        128000    2048-1073741824 maxbuf tcp sack       rw    active --        active    never,passive, active tcp send_      rw    49152  --        49152     4096-1073741824 maxbuf tcp smallest_  rw    32768  --        32768     1024-65535 anon_port tcp smallest_  rw    1024   --        1024      1024-32768 nonpriv_port ... ... ...</pre>
<pre>bash-3.2# nnd -get /dev/tcp ecn 1</pre>	<pre>tcp send_      rw    49152  --        49152     4096-1073741824 maxbuf</pre>
<pre>bash-3.2# nnd -get /dev/tcp sack 2</pre>	<pre>tcp smallest_  rw    32768  --        32768     1024-65535 anon_port tcp smallest_  rw    1024   --        1024      1024-32768 nonpriv_port ... ... ... bash-3.2# ipadm show-prop -p ecn,sack tcp PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE tcp ecn         rw    passive --        passive  never,passive,active tcp sack       rw    active  --        active    never,passive,active</pre>

表 B-2 nnd 和 ipadm 命令之间的语法对应关系：设置属性

nnd 命令	ipadm 命令
<pre>bash-3.2# nnd -set /dev/ip \ ip_def_ttl 64 bash-3.2# nnd -get /dev/ip \ ip_def_ttl 64</pre>	<pre>bash-3.2# ipadm set-prop -p ttl=64 ipv4 bash-3.2# ipadm show-prop -p ttl ip PROTO PROPERTY FAMILY PERM VALUE DEFAULT POSSIBLE ip    ttl      inet   rw   64    255    1-255 PROTO PROPERTY PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4  ttl      rw     64       64          255     1-255 bash-3.2# ipadm reset-prop -p ttl ip bash-3.2# ipadm show-prop -p ttl ip PROTO PROPERTY PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4  ttl      rw     255     255          255     1-255</pre>



# 索引

---

## A

autopush 属性, 27

## B

BSSID, 50

## C

cfgadm 命令, 33

CIDR 表示法, 18

## D

DefaultFixed NCP, 11

DHCP, 19

dladm 命令, 11-14, 23-25

connect-wifi, 51

delete-phys, 25

help, 12

rename-link, 25

reset-linkprop, 25-30

scan-wifi, 50

set-linkprop, 25-30

show-ether, 29-30, 30

show-link, 24

show-linkprop, 29-30, 53

show-phys, 24-25

show-wifi, 51

## E

ECMP, 41-42

ESSID, 50

/etc/hosts 文件, 19

## G

GLDv3, 12

## I

ICMP, 13

ifconfig 命令, 13

和 ipadm 命令, 57-58

IP 地址

DHCP, 19

IPv4 和 IPv6, 19

包转发, 37-38, 39-40

本地和远程, 18

监视, 44-48

静态, 18

删除, 36

属性, 38, 48

IP 多路径接口 (IPMP), 18-21

IP 接口

IP 地址, 47-48, 48

IPMP 接口, 18-21

TCP/IP 协议属性, 39-44

VNI 接口, 18-21

WiFi, 50

**IP 接口 (续)**

- 创建和检测, 18–21
  - 地址属性, 38
  - 分配 IP 地址, 18
  - 更改 IP 地址, 36
  - 更改主接口, 35–36
  - 监视, 44–48
  - 接口属性, 46–47
  - 禁用和启用, 36
  - 配置, 20
  - 启用包转发, 37–38, 39–40
  - 删除 IP 地址, 36
  - 删除接口配置, 35–36
  - 设置接口属性, 37
  - 特权端口, 40
  - 显示
    - IP 地址, 47–48
    - 常规信息, 20, 45
    - 地址属性, 48
    - 接口, 45–46
    - 接口属性, 46–47
    - 协议属性, 39
  - 显示接口属性, 37
  - 验证 MAC 地址的唯一性, 16–17
- IP 隧道, 18**
- 本地和远程地址, 18
- ipadm 命令, 11–14, 35–48**
- create-addr, 18
  - create-ip, 18–21
  - delete-addr, 36
  - delete-ip, 35–36
  - disable-ip, 36
  - help, 13
  - ifconfig 命令比较, 57–58
  - ndd 命令比较, 59–61
  - set-addrprop, 38
  - set-ifprop, 37
  - set-prop, 39–44
  - show-addr, 47–48
  - show-addrprop, 38, 48
  - show-if, 45–46
  - show-ifprop, 37, 46–47
  - show-prop, 39–44

**M**

- MAC 地址, 验证唯一性, 16–17
- MTU, 26

**N**

- name-service/switch 服务, 19
- NCP, 请参见网络配置文件
- ndd 命令, 13
  - 和 ipadm 命令, 59–61
- netadm 命令, 11–14, 17–18
- netcfg 命令, 11–14
- netstat 命令, 检查通过 WiFi 链路的包流, 54
- NIC 驱动程序, 26

**R**

- route 命令, 20

**S**

- SCTP, 13
- STREAMS 模块, 和数据链路, 27–28

**T**

- TCP 接收缓冲区大小, 43–44

**U**

- UDP, 13
- USENIX, 29

**W**

- WiFi
  - IEEE 802.11 规范, 49
  - WiFi 配置示例, 52
  - 安全的 WiFi 链路, 54

**WiFi (续)**

- 定义, 49
  - 对连接进行加密, 55
  - 基本服务集 ID (Basic Service Set ID, BSSID), 50
  - 加密的通信示例, 56
  - 监视链路, 53
  - 检查包流, 54
  - 扩展服务集 ID (Extended Service Set ID, ESSID), 50
  - 连接到 WiFi 网络, 50, 51, 52
  - 示例, 设置链路速度, 54
- WLAN, 10

**安**

- 安全注意事项, WiFi, 54

**半**

- 半双工, 27

**包**

- 包转发
  - 在接口上, 37-38
  - 在协议上, 39-40

**本**

- 本地地址, 18

**带**

- 带宽延迟乘积 (Bandwidth Delay Product, BDP), 43-44

**地**

- 地址对象, 19

- 地址解析协议 (Address Resolution Protocol, ARP), 13

**电**

- 电源管理, 27

**动**

- 动态重新配置 (Dynamic Reconfiguration, DR), 替换 NIC, 32

**对**

- 对称路由, 41-42

**多**

- 多宿主主机, 41-42

**反**

- 反应性网络配置, 9-14
  - ENM, 10
  - WLAN, 10
  - 位置配置文件, 10

**服**

- 服务管理工具 (service management facility, SMF), 13

**固**

- 固定网络配置, 9-14
  - 静态 IP 地址, 9

## 缓

缓冲区, 28

## 巨

巨型帧, 启用支持, 26

## 链

链路聚合, 23-24

链路名称, 25

链路速度, 27

## 配

配置文件管理的网络配置, 9-10

## 全

全双工, 27

## 数

数据链路

autopush 属性, 27

DMA 绑定, 28

STREAMS 模块, 27-28

VLAN, 23-24

VNIC, 23-24

更改 MTU 大小, 26

公共属性和专用属性, 26

链路聚合, 23-24

链路速度, 27

删除, 25

设置属性, 25-30

通告速度和支持速度, 27

通用名称, 25

物理链路, 24

显示

常规信息, 23-24

数据链路, 显示 (续)

链路, 24

链路属性, 29-30

网络驱动程序属性, 29-30, 30

物理属性, 24-25

系统上的物理位置, 24

以太网参数值, 29-30

在链路上配置 IP 接口, 18-21

中断率, 29

重命名, 25

自动协商, 26

## 特

特权端口, 40

## 通

通用 LAN 驱动程序 (Generic LAN Driver, GLD), 29

## 外

外部网络修饰器 (external network modifier, ENM), 10

## 网

网络接口卡 (network interface card, NIC), 替换, 使用 DR, 32

网络配置工具, 11-14

dladm 命令, 12

ipadm 命令, 13

netadm 命令, 11

netcfg 命令, 11

网络配置文件 (network configuration profile, NCP), 9-14

DefaultFixed, 10, 11

反应性, 10-11

固定, 10-11

活动的 NCP, 9-14

列出 NCP, 17-18

网络配置文件 (network configuration profile, NCP)  
(续)  
切换活动 NCP, 11, 17-18

## 位

位置配置文件, 10

## 无

无线接口, 49

## 协

协议, 属性, 39-44

## 虚

虚拟局域网 (virtual local area network,  
VLAN), 23-24  
虚拟网络接口 (virtual network interface, VNI), 18-21  
虚拟网卡 (VNIC), 23-24  
虚拟专用网络 (virtual private networks, VPN), 27-28

## 以

以太网参数, 29-30

## 拥

拥塞控制, 42-43

## 远

远程地址, 18

## 直

直接内存访问 (Direct Memory Access, DMA), 28

## 中

中断率, 29

## 主

主接口, 切换, 25, 30-34, 35-36

## 自

自动协商, 26

