

Oracle® Solaris 11.1 での固定ネット ワーク構成を使用したシステムの接続

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	7
1 固定ネットワーク構成の概要	11
固定ネットワーク構成とは	11
プロファイル管理ネットワーク構成の特徴	12
ネットワーク構成ツール	13
dladm コマンド	15
ipadm コマンド	16
2 ネットワーク用のシステムの構成	19
ネットワークの構成(タスクマップ)	19
▼SPARC: 各インタフェースの MAC アドレスが一意であることを確認する方法 ...	20
▼システム上のアクティブな NCP を変更する方法	21
▼IP インタフェースを構成する方法	23
その他のネットワーク構成および管理タスク	27
3 データリンクの操作	29
基本的な dladm コマンド	29
データリンクに関する一般情報の表示 (dladm)	29
システムのデータリンクの表示 (dladm show-link)	30
データリンクの物理属性の表示 (dladm show-phys)	30
データリンクの削除 (dladm delete-phys)	31
データリンクの名前変更 (dladm rename-link)	31
データリンクプロパティのカスタマイズ	32
データリンクプロパティの概要	32
ジャンボフレームのサポートの有効化	33
リンク速度パラメータの変更	33

データリンク上の STREAMS モジュールの設定	34
ダイレクトメモリアクセスバインディングを使用するための e1000g ドライバ の設定	35
割り込みレートの手動設定	36
データリンクプロパティに関するステータス情報の取得	37
dladm コマンドによるその他の構成タスク	38
▼ システム上のプライマリインタフェースを切り替える方法	38
▼ 動的再構成を使用してネットワークインタフェースカードを交換する方法	40
4 IP インタフェースの操作	43
基本的な ipadm コマンド	43
IP インタフェース構成の削除 (ipadm delete-ip)	43
IP インタフェース構成の無効化 (ipadm disable-ip)	44
インタフェースのアドレスの削除 (ipadm delete-addr)	44
IP インタフェースのプロパティの設定	45
パケット転送の有効化	46
IP アドレスのプロパティの設定	47
TCP/IP プロトコルのプロパティの設定	48
グローバルなパケット転送の有効化	49
特権ポートの設定	50
マルチホームホストに対する対称ルーティングの実装	51
トラフィックの輻輳制御の実装	52
TCP 受信バッファサイズの変更	54
IP インタフェースとアドレスの監視	55
IP インタフェースに関する一般情報の取得	56
IP インタフェースに関する情報の取得	56
IP インタフェースのプロパティに関する情報の取得	58
IP アドレスに関する情報の取得	59
IP アドレスのプロパティに関する情報の取得	60
5 Oracle Solaris を実行するノートパソコン上での無線ネットワークの構成	61
WiFi 通信のタスクマップ	61
▼ WiFi ネットワークに接続する方法	62
▼ WiFi リンクを監視する方法	66
セキュアな WiFi 通信	67

▼ 暗号化された WiFi ネットワーク接続を設定する方法	68
A 比較マップ: ifconfig コマンドと ipadm コマンド	71
B 比較マップ: ndd コマンドと ipadm コマンド	75
索引	79

はじめに

『Oracle Solaris 11.1 での固定ネットワーク構成を使用したシステムの接続』へようこそ。このドキュメントは、Oracle Solaris 11.1 ネットワークの確立に関するシリーズの一部で、Oracle Solaris ネットワークを構成するための基本的なトピックおよび手順について説明しています。このドキュメントの記述は、Oracle Solaris がインストール済みであることが前提です。さらに、ネットワークを構成できる状態であり、そのネットワークに必要なネットワークソフトウェアを構成できる状態である必要があります。

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャを使用するシステムをサポートしています。サポートされるシステムは、Oracle Solaris OS: Hardware Compatibility Lists に記載されています。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

サポートされるシステムについては、[Oracle Solaris OS: Hardware Compatibility Lists](#) を参照してください。

対象読者

このドキュメントは、Oracle Solaris が動作しており、ネットワークに構成されているシステムの管理を行うユーザーを対象としています。このマニュアルを利用するにあたっては、UNIX のシステム管理について少なくとも 2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上的のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	プレースホルダ: 実際に使用する特定の名称または値で置き換えます。	ファイルを削除するには、rm filename と入力します。
<i>AaBbCc123</i>	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第 6 章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#
C シェル	machine_name%

表 P-2 シェルプロンプト (続き)

シェル	プロンプト
Cシェルのスーパーユーザー	machine_name#

固定ネットワーク構成の概要

Oracle Solaris 11 を実行するシステムでは、システム上でアクティブになっているネットワーク構成プロファイル (NCP) に応じて、ネットワーク構成がリアクティブと固定のどちらかになります。リアクティブおよび固定ネットワーク構成の概要については、『[Oracle Solaris 11 ネットワーキングの紹介](#)』を参照してください。NCP の作成および構成方法の詳細は、『[Oracle Solaris 11.1 でのリアクティブネットワーク構成を使用したシステムの接続](#)』を参照してください。

この章では、固定ネットワーク構成の概要を示し、次のトピックについて説明します。

- 11 ページの「固定ネットワーク構成とは」
- 12 ページの「プロファイル管理ネットワーク構成の特徴」
- 13 ページの「ネットワーク構成ツール」

固定ネットワーク構成とは

Oracle Solaris 11 では、ネットワーク構成プロファイル (NCP) によってネットワーク構成が管理されます。システムのネットワーク構成は、その特定のシステムで動作する NCP のタイプによって決まります。NCP がリアクティブの場合、そのシステムのネットワーク構成は動的に実装されます。NCP が固定の場合、ネットワーク構成は静的に実装されます。

固定ネットワーク構成とは、システム上で特定のネットワーク設定がインスタンス化される構成モードを指します。リアクティブネットワーク構成モードとは異なり、固定構成モードでインスタンス化された構成は、システムのネットワーク環境が変化しても変更されません。インタフェースの追加など、環境に変化が発生した場合は、システムが新しい環境に適応するように、システムのネットワーク設定を手動で再構成する必要があります。

注- 固定ネットワーク構成を単なる静的な IP アドレスの構成と混同しないでください。固定ネットワーク構成では、インタフェースに DHCP アドレスを割り当てることができます。同様に、リアクティブネットワーク構成では、インタフェースを静的な IP アドレスで構成する NCP を作成できます。このように、固定ネットワーク構成は適用範囲がより広く、具体的にはシステムの環境の変化に応じてシステムのネットワーク構成が変化する機能を指しています。

次の表は、2つのネットワーク構成モードの比較を示しています。

機能	リアクティブネットワーク構成	固定ネットワーク構成
システムのネットワーク環境の変化に対する自動適応	構成可能な複数の NCP によってサポートされます	サポートされていません (必要に応じて手動で再構成する必要があります)
システムで動作する NCP のタイプ	リアクティブ (Automatic) またはユーザーが作成したその他の NCP)	固定 (DefaultFixed)
複数の NCP	サポートされています (ただし、一度にアクティブにできる NCP は 1 つだけです)	サポートされていません
ユーザーが作成した NCP	サポートされています	システムによって生成される固定 NCP (DefaultFixed) が 1 つだけ存在します。ただし、DefaultFixed の内容はユーザーがすべて決定します。

次のセクションでは、プロファイル管理ネットワーク構成とネットワーク構成に使用されるツールについて詳しく説明します。

プロファイル管理ネットワーク構成の特徴

Oracle Solaris 11 では、ネットワーク構成はプロファイルに基づいています。システムのネットワーク構成は、NCP と対応する Location プロファイルによって管理されます。プロファイル管理ネットワーク構成の概要については、『Oracle Solaris 11 ネットワーキングの紹介』の「ネットワーク構成プロファイル」を参照してください。NCP の詳細は、『Oracle Solaris 11.1 でのリアクティブネットワーク構成を使用したシステムの接続』を参照してください。

注-ネットワーク構成の主要なプロファイルのタイプは、NCP、Location プロファイル、外部ネットワーク識別子 (ENM)、および無線ローカルエリアネットワーク (WLAN) です。これらのタイプのうち、主要なプロファイルは NCP です。このドキュメント全体を通じて、特に明記しないかぎり、「プロファイル」という用語は NCP を指します。

プロファイルベースのネットワーク構成の特徴は、次のとおりです。

- あるシステムのネットワーク構成を管理するために一度にアクティブになるのは、1組の NCP と場所プロファイルだけです。システム上のほかの既存の NCP はすべて、非運用状態になります。
- アクティブな NCP は「リアクティブ」または「固定」のいずれかになります。リアクティブプロファイルでは、システムのネットワーク環境の変化に適応できるように、ネットワーク構成が監視されます。固定プロファイルでは、ネットワーク構成はインスタンス化されますが、監視されません。
- アクティブな NCP がリアクティブの場合、システムのネットワーク構成は適応性を持っています。アクティブな NCP が固定の場合、システムのネットワーク構成は不変です。
- NCP のさまざまなプロパティの値が、そのプロファイルによるネットワーク構成の管理方法を制御するポリシーを構成します。
- NCP のプロパティに対する変更は、新しいプロパティ値としてただちに実装されます。これらの新しい値は、ネットワーク構成を管理するプロファイルのポリシーの一部になります。

システムを固定ネットワーク用に構成すると、ネットワーク構成を管理するアクティブな NCP は DefaultFixed になります。このプロファイルは、OS によって生成され、システム上の唯一の固定プロファイルになります。システムは複数の固定プロファイルをサポートしません。

DefaultFixed NCP のプロパティには、DefaultFixed NCP がアクティブである間に作成または変更された永続的な構成が反映されます。

ネットワーク構成ツール

Oracle Solaris 11 では、4つのネットワークコマンドを使用してネットワークを構成できます。

- netcfg コマンド
- netadm コマンド
- dladm コマンド
- ipadm コマンド

netcfg および netadm コマンドは、システム上のリアクティブネットワーク構成を管理するため使用します。netcfg コマンドは、リアクティブネットワーク構成を実装するプロファイル (NCP、Location プロファイル、ENM、および WLAN) を作成および構成するために使用します。ただし、固定ネットワーク構成を持つシステムでは、DefaultFixed プロファイルを表示する場合にのみ netcfg コマンドを使用できません。netadm コマンドは、システム上のすべてのプロファイルを管理する (特に、システムのネットワークプロファイルを一覧表示したり、アクティブな NCP を別のものに置き換えたりする) ために使用します。

dladm および ipadm コマンドは、それぞれデータリンクおよび IP インタフェースを構成するために使用します。これらのコマンドは、永続的な構成を作成し、コマンドの使用時にシステム上でアクティブになっているプロファイルに適用されます。

たとえば、データリンク net0 の最大転送単位 (MTU) が 1200 に構成され、アクティブな NCP が Automatic である場合、Automatic NCP ではこの MTU 値が net0 に対して永続的になります。次に、myncp という名前の 2 つ目の NCP をアクティブにします。dladm コマンドを実行して MTU を別の値に設定すると、その値が myncp に適用されます。このように、net0 は異なるプロファイルで異なる MTU 値を持つことができます。したがって、dladm および ipadm コマンドを使用して間接的にプロファイルを構成することもできます。

dladm または ipadm コマンドを使用してデータリンクおよび IP インタフェースを構成するときは、次に示すこれらの適用範囲に注意してください。

- 2 つのコマンドは、アクティブなプロファイルのデータリンクと IP インタフェースのみを構成します。デフォルトルートの設定など、プロファイルのほかのプロパティを構成するには、netcfg コマンドを使用して、そのプロファイルのデフォルトルートを参照するプロパティを構成します。または、システムのルーティングテーブルにデフォルトルートを直接設定する routeadm コマンドを使用します。後者の場合、構成はシステム上でアクティブになっている方のプロファイルに適用されます。
- リアクティブプロファイルがアクティブであれば、そのプロファイルに対して dladm および ipadm コマンドを使用できます。ただし、netcfg コマンドを使用して、システムの唯一の固定プロファイルである DefaultFixed プロファイルを構成することはできません。netadm および netcfg コマンドは、DefaultFixed プロファイルのプロパティを表示する場合にのみ使用でき、それらを構成する場合には使用できません。

dladm および ipadm コマンドは、リアクティブプロファイルか固定プロファイルかに関係なく、アクティブなプロファイルに対して有効です。したがって、これらのコマンドを使用する前に、次を確認する必要があります。

- 正しいターゲットプロファイルに対して変更を行えるように、システム上でどのプロファイルがアクティブであるかを確認します。

- コマンド使用後に予期しない構成動作が発生しないように、ターゲットプロファイルがリアクティブ、固定のいずれであるかを確認します。リアクティブプロファイルは、固定プロファイルと異なる方法でネットワーク構成を管理します。したがって、変更実装時の2つのプロファイルの動作も異なります。

次のセクションでは、`dladm` および `ipadm` コマンドについて詳しく説明します。

dladm コマンド

`dladm` コマンドは、データリンクを構成するために使用します。`dladm` コマンドを使用してデータリンクプロパティをカスタマイズできるのは、`e1000g` のように、リンクのネットワークドライバが GLDv3 ドライバ構成フレームワーク用に変換されている場合です。特定のドライバがこの機能をサポートしているか確認するには、ドライバのマニュアルページを参照してください。

GLDv3 ドライバ構成フレームワークの完全な実装により、ネットワークインタフェースカード (NIC) ドライバの構成は次のように拡張されています。

- ネットワークドライバのプロパティの構成に必要なのは、単一のコマンドインタフェースである `dladm` コマンドだけです。
- プロパティにかかわらず、`dladm subcommand properties datalink` という一貫した構文が使用されます。
- `dladm` コマンドの使用は、ドライバの公開プロパティと非公開プロパティの両方に適用されます。
- 特定のドライバに対して `dladm` コマンドを使用しても、似たタイプのほかの NIC のネットワーク接続に支障が生じることはありません。したがって、データリンクのプロパティを動的に構成できます。
- データリンク構成の値は `dladm` リポジトリ内に格納され、システムのリブート後も永続します。

データリンクの構成時にこれらのメリットの恩恵を受けるには、`ndd` コマンドのような以前のリリースの従来型ツールではなく、`dladm` を構成ツールとして使用するようにしてください。

`dladm` コマンドの詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。`dladm` コマンドで使用するサブコマンドの一覧については、次を入力します。

```
# 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 Aggr   : 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>`.

データリンクに対して `dladm` コマンドを使用するには、第3章「データリンクの操作」を参照してください。

ipadm コマンド

Oracle Solaris における進歩は、ネットワーク構成のさまざまな側面を効率的に管理するための従来のツールの機能範囲を超えています。たとえば、`ifconfig` コマンドはこれまで、ネットワークインタフェースを構成するために慣例的に使用されるツールでした。ただし、このコマンドには永続的な構成が実装されていません。`ifconfig` にはこれまで、ネットワーク管理における追加機能のための拡張機能が追加されてきました。ただし、その結果、このコマンドは複雑で、使いづらくなっています。

インタフェースの構成と管理に関する別の問題として、TCP/IP のプロパティまたはチューニング可能パラメータを管理するための簡単なツールが存在しない点があります。`ndd` コマンドは、この目的のための定番のカスタマイズツールでした。ただし、`ifconfig` コマンドと同様に、`ndd` には永続的な構成が実装されていません。以前は、ブートスクリプトの編集によって、ネットワークのシナリオで永続的な構成をシミュレートすることが可能でした。Oracle Solaris のサービス管理機能 (SMF) の導入により、特に、Oracle Solaris インストールへのアップグレードを考慮した場合の SMF 依存性の管理の複雑さのために、このような回避策の使用はリスクを伴うことがあります。

`ipadm` コマンドは、インタフェース構成のための `ifconfig` コマンドを最終的に置き換えるために導入されました。このコマンドはまた、プロトコルのプロパティを構成するための `ndd` コマンドも置き換えます。

インタフェースを構成するためのツールとして、`ipadm` コマンドには次の利点があります。

- インタフェース構成以外の目的にも使用される `ifconfig` コマンドとは異なり、IP インタフェースの管理に特化したツールであるため、IP インタフェースと IP アドレスをより効率的に管理します。
- インタフェースとアドレスの永続的な構成を実装します。

`ifconfig` のオプションおよびそれらと同等の `ipadm` サブコマンドの一覧については、付録 A 「比較マップ: `ifconfig` コマンドと `ipadm` コマンド」を参照してください。

プロトコルのプロパティを設定するためのツールとして、`ipadm` コマンドには `ndd` に優る次の利点があります。

- IP、アドレス解決プロトコル (ARP)、SCTP (Stream Control Transmission Protocol)、ICMP (Internet Control Messaging Protocol) のほか、TCP やユーザーデータグラムプロトコル (UDP) などの上位階層プロトコルの一時的または永続的なプロパティを設定できます。
- プロパティの現在値とデフォルト値、指定可能な値の範囲などの、各 TCP/IP プロパティに関する情報を提供します。そのため、デバッグ情報の取得が容易になります。
- また、一貫性のあるコマンド構文に従っているため、より容易に使用できます。

`ndd` のオプションおよびそれらと同等の `ipadm` サブコマンドの一覧については、[付録 B 「比較マップ: `ndd` コマンドと `ipadm` コマンド](#)」を参照してください。

`ipadm` コマンドの詳細は、[ipadm\(1M\)](#) のマニュアルページを参照してください。 `ipadm` で使用するサブコマンドの一覧については、次を入力します。

```
# 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>.
```


ネットワーク用のシステムの構成

この章では、固定ネットワーク構成を使用するシステム上の IP インタフェースを構成するときに従う手順を示します。次の内容について説明します。

- 19 ページの「ネットワークの構成(タスクマップ)」
- 27 ページの「その他のネットワーク構成および管理タスク」

ネットワークの構成(タスクマップ)

このセクションでは、IP インタフェースの基本的な構成手順について説明します。次の表は、各構成タスクについて説明し、これらのタスクを対応する手順にマップしています。

タスク	説明	参照先
一意の MAC アドレスをサポートするようにシステムを構成します。	インタフェースに対して一意の MAC アドレスを許可するように SPARC ベースのシステムを構成します。	20 ページの「SPARC: 各インタフェースの MAC アドレスが一意であることを確認する方法」
システム上のアクティブな NCP を判定します。	システム上のアクティブな NCP を表示し、DefaultFixed を有効にします。	21 ページの「システム上のアクティブな NCP を変更する方法」
ipadm コマンドを使用して、基本的な IP インタフェース構成を実行します。	IP インタフェースを作成し、そのインタフェースに有効な IP アドレス (静的または DHCP) を割り当てます。	23 ページの「IP インタフェースを構成する方法」
データリンクをカスタマイズします。	リンクプロパティを設定して、データリンクをさらにカスタマイズします。	32 ページの「データリンクプロパティのカスタマイズ」

タスク	説明	参照先
IP インタフェースをカスタマイズします。	インタフェースのプロパティを設定して、IP インタフェースをさらにカスタマイズします。	45 ページの「IP インタフェースのプロパティの設定」
IP アドレスをカスタマイズします。	アドレスのプロパティを設定して、IP アドレスをさらにカスタマイズします。	47 ページの「IP アドレスのプロパティの設定」
プロトコルをカスタマイズします。	プロトコルのプロパティを設定して、プロトコルをさらにカスタマイズします。	48 ページの「TCP/IP プロトコルのプロパティの設定」
ワイヤレスネットワークを構成します。	無線ネットワークを使用してノートパソコンをネットワークに接続します。	第5章「Oracle Solaris を実行するノートパソコン上での無線ネットワークの構成」

▼ SPARC: 各インタフェースの MAC アドレスが一意であることを確認する方法

すべての SPARC ベースのシステムは、システム共通 MAC アドレスを持っており、デフォルトではすべてのインタフェースがこのアドレスを使用します。ただし、アプリケーションによっては、ホスト上のすべてのインタフェースでそれぞれ一意の MAC アドレスが使用されている必要があります。特定のタイプのインタフェース構成(リンクアグリゲーションや IP マルチパス (IPMP) など)では、同様にインタフェースに固有の MAC アドレスが存在する必要があります。

EEPROM パラメータ `local-mac-address?` によって、SPARC ベースのシステム上のすべてのインタフェースがシステム共通 MAC アドレスまたは一意の MAC アドレスのどちらを使用しているかを判別します。次の手順では、`eeprom` コマンドを使用して、`local-mac-address?` の現在値をチェックし、必要に応じて変更する方法について説明します。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 システム上のすべてのインタフェースがシステム共通 MAC アドレスを現在使用しているかどうかを判断します。

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

この例では、`eeprom` コマンドの応答の `local-mac-address?=false` によって、すべてのインタフェースがシステム共通 MAC アドレスを使用していることが示されています。

す。local-mac-address?=false の値は、インタフェースを IPMP グループのメンバーにする前に、local-mac-address?=true に変更する必要があります。リンクアグリゲーションの場合も、この変更を行うべきです。

- 必要に応じて、local-mac-address? の値を次のように変更します。

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

手順 6 でシステムをリブートすると、出荷時に設定された MAC アドレスを持つインタフェースは、システム共通 MAC アドレスの代わりに、それらの出荷時の設定を使用します。出荷時に設定された MAC アドレスを持たないインタフェースは、システム共通 MAC アドレスを引き続き使用します。

- システム上のすべてのインタフェースの MAC アドレスをチェックします。

複数のインタフェースが同じ MAC アドレスを持つ場合がないかどうかを調べてください。この例では、2つのインタフェースがシステム共通 MAC アドレス 8:0:20:0:0:1 を使用しています。

```
# 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 - 同じ MAC アドレスを持つネットワークインタフェースが複数存在する場合にのみ、次の手順に進みます。それ以外の場合は、最後の手順に進みます。

- すべてのインタフェースが一意的 MAC アドレスを持つように、必要に応じて、残りのインタフェースを手動で構成します。

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

前の手順の例では、net0 と net1 をローカルで管理されている MAC アドレスで構成する必要がありました。たとえば、ローカルで管理されている MAC アドレス 06:05:04:03::02 で net0 を再構成するには、次のコマンドを入力します。

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

このコマンドの詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

- システムをリブートします。

▼ システム上のアクティブな NCP を変更する方法

システム上で有効になっている NCP のタイプによって、システムのネットワーク構成がリアクティブと固定のどちらであるかが決まります。リアクティブ構成を使用するシステムは、固定ネットワーク構成を使用するシステムとは異なる動作を行い

ます。このドキュメントのすべての手順では、アクティブな NCP に適用される永続的な構成を作成します。したがって、手順を実行する前に、構成を正しいプロファイルに適用できるように、どの NCP がアクティブであるかを知っておく必要があります。このようにして、手順が完了すると、システムのネットワーク構成が予期したとおりに動作します。

- 1 システム上のプロファイルを一覧表示します。

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

ステータスがオンラインとして一覧表示されているプロファイルがシステム上のアクティブな NCP です。

システム上の NCP の詳細を表示するには、`netadm` コマンドで `-x` オプションを使用します。

```
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
```

- 2 プロファイルのタイプを(たとえば、リアクティブプロファイルから固定プロファイルに)切り替えるには、次のコマンドを入力します。

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

ここで、*NCP-name* は NCP のタイプの名前です。

たとえば、システムのネットワーク構成がリアクティブであるとしします。このドキュメントの手順で作成した構成を DefaultFixed NCP に適用する場合は、次を入力します。

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



注意-アクティブなプロファイルを切り替えると、既存のネットワーク構成は削除され、新しい構成が作成されます。以前アクティブだった NCP に実装された永続的な構成は、新しいアクティブな NCP ではすべて除外されます。

▼ IP インタフェースを構成する方法

次の手順では、システムの IP インタフェースを構成するときに使用する基本的な手順を示します。

始める前に システム上でどの NCP がアクティブであるかを確認して、構成が正しいプロファイルに適用されていることを確認します。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 インタフェースを作成します。

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

interface-class ユーザーが作成できる次の3つのインタフェースクラスのうちの一つを示します。

- IP インタフェース。このインタフェースクラスは、ネットワーク構成を実行するときに作成するもっとも一般的なクラスです。このインタフェースクラスを作成するには、`create-ip` サブコマンドを使用します。
- STREAMS 仮想ネットワークインタフェースドライバ (VNI インタフェース)。このインタフェースクラスを作成するには、`create-vni` サブコマンドを使用します。VNI デバイスまたはインタフェースについての詳細は、[vni\(7d\)](#) のマニュアルページを参照してください。
- IPMP インタフェース。このインタフェースは、IPMP グループを構成するときに使用されます。このインタフェースクラスを作成するには、`create-ipmp` サブコマンドを使用します。IPMP グループの詳細は、『Oracle Solaris 11.1 ネットワークパフォーマンスの管理』の第5章「IPMP の概要」を参照してください。

interface インタフェースの名前を示します。この名前は、インタフェースの作成に使用しているデータリンクの名前と同じです。システム上のデータリンクを確認するには、`dladm show-link` コマンドを使用します。

3 次のいずれかのコマンドを選択して、有効な IP アドレスを持つ IP インタフェースを構成します。

- 静的アドレスを構成するには、次を入力します。

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

`-a address` インタフェース上に構成する IP アドレスを指定します。

注-トンネル構成では、通常、トンネルインタフェースにローカルアドレスとリモートアドレスの2つのアドレスが必要です。ローカルアドレスとリモートアドレス、およびトンネル構成については、『Oracle Solaris 11.1 ネットワークの構成と管理』の第6章「IP トンネルの構成」を参照してください。

数値 IP アドレスには、CIDR 表記法を使用します。CIDR 表記法を使用なかった場合は、name-service/switch サービスで netmask に対して示されているシーケンスに従って、またはクラスフルアドレスのセマンティクスを使用して、ネットマスクが計算されます。

必要に応じて、数値 IP アドレスの代わりにホスト名を指定できます。ホスト名の使用が有効なのは、そのホスト名に対応する数値 IP アドレスが /etc/hosts ファイルで定義されている場合です。このファイルで数値 IP アドレスが定義されていない場合は、name-service/switch サービスで host に対して指定されているリゾルバの順序を使用して、この数値が一意に取得されます。特定のホスト名に対して複数のエントリが存在する場合は、エラーが生成されます。

注-ブートプロセスにおいて、IP アドレスの作成は、ネームサービスがオンラインになる前に実行されます。そのため、ネットワーク構成で使用されているすべてのホスト名が /etc/hosts ファイルで定義されているようにする必要があります。

[*interface* | *addrobj*]

Oracle Solaris では、各アドレスは対応するアドレスオブジェクトによって識別され、コマンド内で *addrobj* によって表されます。そのアドレスに対するその後の構成では、実際の IP アドレスではなくアドレスオブジェクトを参照します。たとえば、`ipadm show-addr addrobj` や `ipadm delete-addr addrobj` のように入力します。アドレスオブジェクト名を自動的に作成するには、*interface* にインタフェース名を指定します。それ以外の場合は、アドレスオブジェクト名を直接指定します。

- インタフェース名を指定すると、アドレスオブジェクトに *interface/address-family* という形式の名前が自動的に付けられます。Address family は、IPv4 アドレスを示す v4 ま

たはIPv6アドレスを示すv6のいずれかです。同じインタフェース上に複数のアドレスがある場合は、net0/v4、net0/v4a、net0/v4b、net0/v6、net0/v6aのように、アドレスオブジェクト名に英字が追加されます。

- *addrobj*に手動でアドレスオブジェクトを指定する場合は、*interface/user-specified-string*という形式を使用する必要があります。*user-specified-string*は、アルファベット文字で始まり、最大長が32文字の英数字の文字列を示します。たとえば、net0/static、net0/static1、net1/privateなどのアドレスオブジェクトを指定できます。

- 静的でないアドレスを構成するには、次を入力します。

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

ここで、*address-type*はdhcpまたはaddrconfのいずれかです。addrconfは、自動的に生成されたIPv6アドレスを示します。

[*interface | addrobj*]の詳細な説明については、静的アドレスの作成に関する前の説明を参照してください。

- 4 (オプション)新しく構成されたIPインタフェースに関する情報を表示します。

調べる対象の情報に応じて、次のコマンドを使用できます。

```
# ipadm [interface]
```

*interface*を指定しなかった場合は、システム上のすべてのインタフェースの情報が表示されます。

ipadm show-*サブコマンドの出力についての詳細は、55ページの「IPインタフェースとアドレスの監視」を参照してください。

- 5 ホスト名を使用する静的IPアドレスを構成している場合は、*/etc/hosts*ファイルにそのIPアドレスのエントリを追加します。

このファイル内のエントリは、IPアドレスとそれに対応するホスト名で構成されています。

注-DHCPアドレスを構成している場合は、*/etc/hosts*ファイルを更新する必要はありません。

- 6 デフォルトルートを定義します。

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

ルーティングテーブルの内容を確認するには、netstat -r コマンドを使用します。

ルートの管理の詳細は、[route\(1M\)](#) および [routeadm\(1M\)](#) のマニュアルページを参照してください。『Oracle Solaris 11.1 ネットワークの構成と管理』の「ルーティングテーブルとルーティングの種類」も参照してください。

例 2-1 静的 IP アドレスによるネットワークインタフェースの構成

この例では、静的 IP アドレスを使用してインタフェースを構成する方法について説明します。この例では、最初にシステム上の DefaultFixed NCP を有効にして、固定ネットワーク構成に対して dladm および ipadm コマンドを使用できるようにします。

```
# 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         --          --
  lo0/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
```

/etc/hosts ファイルで campus01 がすでに定義されている場合は、次のアドレスを割り当てるときにそのホスト名を使用できます。

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

例 2-2 IP アドレスによるネットワークインタフェースの自動的な構成

この例では、アドレスを DHCP サーバーから受け取るように IP インタフェースを構成します。

```
# 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         --          --
  lo0/v4   static      ok         --          127.0.0.1/8
net3      ip          ok         --          --
  net3/v4  dhcp        ok         --          10.0.1.13/24
```

その他のネットワーク構成および管理タスク

このドキュメントでは、システムをネットワークに接続する基本的なネットワーク構成について説明しています。特に、システムのデータリンクとインタフェースの構成を中心に説明しています。ほかのネットワーク関連ドキュメントで説明されているその他のネットワーク構成および管理タスクを実行できます。システムが固定ネットワーク構成で構成されている場合は、これらのほかのタスクについて次のドキュメントを参照できます。

- システムをルーターやネットワーク構成サーバーなどとして構成するには、『[Oracle Solaris 11.1 ネットワークの構成と管理](#)』を参照してください。
- データリンクおよび IP インタフェースの高度な構成を行うには、『[Oracle Solaris 11.1 ネットワークパフォーマンスの管理](#)』を参照してください。
- 基本的な構成を基盤にして、(リンクアグリゲーションや IPMP グループの構成などで) ネットワークのパフォーマンスを向上させるには、『[Oracle Solaris 11.1 ネットワークパフォーマンスの管理](#)』を参照してください。
- ネットワークのセキュリティーを確立するには、『[Oracle Solaris 11.1 でのネットワークのセキュリティー保護](#)』を参照してください。

- ネットワークの仮想化を実装するには、『[Oracle Solaris 11.1 での仮想ネットワークの使用](#)』を参照してください。

特定のネットワーク分野 (DHCP やネームサービスなど) を専門に扱うその他のドキュメントもライブラリに用意されています。

データリンクの操作

この章では、`dladm` コマンドについて説明するとともに、データリンクに対してこのコマンドを使用して、現在の構成の表示、プロパティのデフォルト値の変更、またはシステムからのデータリンクの削除を行う方法について説明します。次の内容について説明します。

- 29 ページの「基本的な `dladm` コマンド」
- 32 ページの「データリンクプロパティのカスタマイズ」
- 38 ページの「`dladm` コマンドによるその他の構成タスク」

基本的な `dladm` コマンド

このセクションでは、システムのデータリンクに対して定期的に使用する可能性がある基本的な `dladm` コマンドについて説明します。このセクションに記載されている以外の `dladm` サブコマンドもサポートされます。ほかのサブコマンドについては、[`dladm\(1M\)` のマニュアルページ](#)を参照してください。

注-データリンクの情報を表示する `dladm` サブコマンドを除き、ほかのすべてのサブコマンドでは、最初にデータリンク上の既存のインタフェース構成を削除する必要があります。IP インタフェース構成を削除するには、[43 ページの「IP インタフェース構成の削除 \(`ipadm delete-ip`\)](#)」を参照してください。

データリンクに関する一般情報の表示 (`dladm`)

`dladm` コマンドを単独で使用すると、システムのデータリンクに関する一般情報(クラス、状態、およびベースとなる物理リンクを含む)が表示されます。

```
# 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
```

データリンクは、物理リンク以外のさまざまなクラス (リンクアグリゲーション、仮想 LAN (VLAN)、仮想 NIC (VNIC) など) である可能性があります。これらのほかのデータリンクは、`dladm` コマンドで表示されるデフォルトの情報にも含まれます。たとえば、この出力には、物理データリンク `net1` と `net4` 上に構成されたリンクアグリゲーション `aggr0` が表示されています。

リンクアグリゲーションと VLAN については、『[Oracle Solaris 11.1 ネットワークパフォーマンスの管理](#)』を参照してください。VNIC については、『[Oracle Solaris 11.1 での仮想ネットワークの使用](#)』を参照してください。

システムのデータリンクの表示 (`dladm show-link`)

システム上のデータリンクを表示するには、`dladm show-link` を使用します。システムには、取り付けられた NIC と同じ数のデータリンクがあります。このコマンドでは、オプションを使用して取得する情報をカスタマイズできます。たとえば、`-P` オプションを使用すると、データリンクに関する永続的な構成の情報が取り込まれます。このコマンドで表示された情報に基づいて、その後のネットワーク構成を進めることができます。たとえば、システム上の NIC の数を決定したり、IP インタフェースを構成するとき使用するデータリンクを選択したりできます。

このコマンドを実行すると、次のような情報が表示されます。

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

この例では、システムに、対応する物理 NIC に直接関連付けられた 3 つのデータリンクがあります。`phys` クラスに属するデータリンク上に構成された特殊なデータリンク (集積体や仮想 NIC など) は存在しません。

データリンクの物理属性の表示 (`dladm show-phys`)

システムのデータリンクに関連付けられている物理 NIC に関する情報を表示するには、`dladm show-phys` を使用します。このコマンドをオプションなしで使用すると、次のような情報が表示されます。

```
# 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
```

この出力には、ほかの詳細とともに、汎用リンク名を持つデータリンクが関連付けられている物理 NIC が表示されています。たとえば、net0 は NIC e1000g0 のデータリンク名です。データリンクに設定されたフラグに関する情報を表示するには、-P オプションを使用します。たとえば、r というフラグが設定されたデータリンクは、そのベースとなる NIC が取り外されたことを意味します。

このコマンドのもう 1 つの便利なオプションは、各データリンクの物理的な場所を表示する -L です。この場所によって、データリンクのインスタンス番号 (net0、net1 など) が決まります。

```
# 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
```

データリンクの削除 (dladm delete-phys)

システムからデータリンクを削除するには、dladm delete-phys を使用します。

データリンクの削除は、物理 NIC の削除と緩やかに結び付いています。たとえば、ある物理 NIC がシステムから削除されます。ソフトウェア層がハードウェア層にバインドされなくなるため、その NIC に関連付けられているデータリンク構成は残っています (『Oracle Solaris 11 ネットワーキングの紹介』の「Oracle Solaris でのネットワークスタック」を参照)。したがって、そのデータリンクの名前をほかの NIC に関連付けられたリンクに割り当てることにより、別のベースとなる物理 NIC でこのデータリンク構成を引き続き使用できます。

NIC を交換せずに切り離し、そのデータリンク構成が不要になった場合は、次のようにしてデータリンクを削除できます。

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

ヒント - データリンクの NIC が削除されたかどうかを確認するには、dladm show-phys -P コマンドを使用します。

データリンクの名前変更 (dladm rename-link)

データリンクの名前を変更するには、dladm rename-link を使用します。Oracle Solaris 11 システムでは、OS がすべてのデータリンクに対して自動的に汎用名を設定しま

す。汎用データリンク名については、『Oracle Solaris 11 ネットワーキングの紹介』の「デフォルトの汎用リンク名」を参照してください。

これらの汎用名には、デフォルトで `netn` (`net0`、`net1`、`net2` など) という命名形式が使用されます。OS が名前を管理しているため、通常の管理タスクの一部としてデータリンクの名前を変更することはありません。リンク名の変更が必要な手順については、38 ページの「システム上のプライマリインタフェースを切り替える方法」を参照してください。

データリンクプロパティのカスタマイズ

基本的なデータリンク構成に加えて、`dladm` コマンドを使用してデータリンクプロパティを設定し、ネットワークの要件に応じてそれらをカスタマイズすることもできます。

データリンクプロパティに対しては、3 つの `dladm` サブコマンドを使用します。

- `dladm show-linkprop [-p property] [datalink]` は、データリンクプロパティとそれらの現在値を表示します。-p *property* オプションを使用しなかった場合は、データリンクのすべてのプロパティが一覧表示されます。データリンクを指定しなかった場合は、すべてのデータリンクのすべてのプロパティが一覧表示されます。
- `dladm set-linkprop -p property=value datalink` は、データリンクのプロパティに値を割り当てます。
- `dladm reset-linkprop -p property datalink` は、特定のプロパティをそのデフォルト値にリセットします。

データリンクプロパティの概要

カスタマイズ可能なデータリンクプロパティは、ある特定の NIC ドライバがサポートするプロパティに依存します。`dladm` コマンドを使用して構成可能なデータリンクプロパティは、2 つのカテゴリのいずれかに該当します。

- Ethernet 用のリンク速度や自動ネゴシエーション、またはすべてのデータリンクドライバに適用可能な最大転送単位 (MTU) サイズなど、特定メディアタイプの任意のドライバに適用可能な「公開プロパティ」。
- 特定メディアタイプの NIC ドライバの特定のサブセットに固有の「非公開プロパティ」。これらのプロパティはそのサブセットに固有のものとなる可能性があります。これは、これらのプロパティが、ドライバに関連付けられたハードウェアに密接に関連しているか、あるいはデバッグ関連のチューニング可能パラメータのように、ドライバ実装自体の詳細に密接に関連しているからです。

リンクプロパティには通常、デフォルト値があります。ただし、特定のネットワークシナリオでは、特定のプロパティの値を変更する必要がある場合もあります。たとえば、NICは、自動ネゴシエーションを正しく実行しない古いスイッチと通信する可能性があります。また、スイッチは、ジャンボフレームをサポートするように構成されている可能性があります。または、特定のドライバで、パケットの送信やパケットの受信を制御するドライバ固有のプロパティを変更する必要がある可能性もあります。以降のセクションでは、選択したプロパティについて説明し、それらの値をネットワーク環境で機能するように変更する方法について説明します。

ジャンボフレームのサポートの有効化

MTUは、あるプロトコルによってシステムから転送できるパケットの最大サイズを定義します。大部分のNICドライバでは、MTUサイズがデフォルトで1500に定義されています。しかし、ジャンボフレームがネットワークを通過する場合は、デフォルト値では不十分です。ジャンボフレームをサポートするには、MTUサイズを少なくとも9000にする必要があります。

MTUサイズをデフォルト値から変更するには、次のコマンドを入力します。

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

MTUサイズを変更したあとは、データリンク上のIPインタフェースを再構成できます。

次の例は、ジャンボフレームのサポートを有効にする手順を示しています。この例は、データリンク上の既存のIPインタフェース構成がすでに削除されていることを前提にしています。

```
# 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         --         --
```

リンク速度パラメータの変更

ほとんどのネットワーク設定は、異なる速度機能を備えたシステムの組み合わせから構成されます。各システムは、ネットワーク内のほかのシステムに速度機能を通知して、各システムがネットワークトラフィックをどのように送受信するかを知らせます。次の対になったデータリンクプロパティは、システムによって通知される速度機能を制御します。

- `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`

それぞれのリンク速度機能は、通知速度 (`adv_*_cap`) と有効化通知速度 (`en_*_cap`) からなる一組のプロパティで参照されます。さらに、プロパティ名に含まれる `*fdx*` と `*hdx*` が示すように、全二重機能と半二重機能のどちらに対してもデータリンク速度の情報が提供されます。通知速度プロパティは、特定のデータリンク速度が通知されるかどうかを示す読み取り専用プロパティです。特定のデータリンク速度が通知されるかどうかを決定するには、対応する `en_*_cap` プロパティを設定します。

デフォルトでは、データリンクのすべての速度機能と二重機能が通知されます。しかし、新しいシステムが古いシステムと通信していて、自動ネゴシエーションが無効または未サポートになっている場合もあります。これらの2つのシステム間の通信を有効にするには、古いシステムと新しいシステム間の通知速度を低い値に変更する必要がある場合があります。場合によっては、システムのギガビット機能をオフにする必要があり、低速な速度機能のみが通知されます。この場合は、全二重機能と半二重機能の両方について、次を入力します。

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

このコマンドは、全二重機能と半二重機能について、システムのギガビット機能の通知をオフにします。

これらのプロパティの新しい値を表示するには、`dladm show-linkprop` コマンドを使用します。

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

通常、特定の有効化速度プロパティと対応する通知プロパティの値は同一です。ただし、NICがPower Managementなどの一部の高度な機能をサポートしている場合は、それらの機能が、ホストとそのリンクパートナーとの間で実際に通知されるビット数の制限を設定する可能性があります。たとえば、Power Managementを使用する場合、`adv_*_cap` プロパティの設定が `en_*_cap` プロパティの設定のサブセットにすぎなくなる可能性があります。

データリンク上の STREAMS モジュールの設定

データリンクが開かれたときにストリーム上にプッシュされる STREAMS モジュールを、最大8個まで設定できます。これらのモジュールは通常、仮想プライベート

ネットワーク (VPN) やファイアウォールなど、他社製のネットワークソフトウェアによって使用されます。そのようなネットワークソフトウェアに関するドキュメントは、ソフトウェアベンダーから提供されています。

特定のデータリンク上にプッシュするモジュールのリストは、`autopush` プロパティによって制御されます。また、`autopush` プロパティの値は、`dladm set-linkprop` サブコマンドを使用することによって設定されます。

また、別個の `autopush` コマンドを使用して、データリンクのストリーム上にモジュールをドライバ単位でプッシュすることもできます。このコマンドは、ドライバごとに設定された構成ファイルを使用して、プッシュするモジュールの情報を取得します。ただし、ドライバは常に NIC にバインドされます。データリンクのベースとなる NIC が取り外されると、そのリンクの `autopush` プロパティの情報も失われます。

したがって、このためには `autopush` コマンドではなく `dladm` コマンドを使用すべきです。特定のデータリンクでドライバ単位とリンク単位の両方のタイプの `autopush` 構成が存在している場合は、`dladm set-linkprop` で設定されたリンク単位の情報が使用され、ドライバ単位の情報は無視されます。

データリンクが開かれたときに STREAMS にモジュールをプッシュするには、同じ `dladm set-linkprop` コマンドを使用して、`autopush` プロパティのモジュールを指定します。たとえば、`vpnmod` および `bufmod` モジュールをリンク `net0` の上にプッシュするには、次を入力します。

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

ダイレクトメモリーアクセスバインディングを使用するための **e1000g** ドライバの設定

このセクションと次のセクションでは、非公開プロパティの構成方法を示します。どちらのセクションも、**e1000g** ドライバ固有のプロパティに適用されます。ただし、これらのセクションの一般情報は、ほかの NIC ドライバの非公開プロパティを構成するときに適用されます。

ファイル転送などの一括トラフィックでは通常、ネットワーク経由で大きなパケットのネゴシエーションが必要となります。そのような場合に、**e1000g** ドライバからより高いパフォーマンスを引き出すには、ダイレクトメモリーアクセス (DMA) バインディングを自動的に使用するようにドライバを構成し、その中でパケットフラグメントサイズのしきい値を定義します。フラグメントサイズがこのしきい値を超えた場合は、パケットの伝送に DMA バインディングが使用されます。フラグメントサイズがこのしきい値以下である場合は、`bcopy` モードが使用され、事前に割り当てられた伝送バッファにフラグメントデータがコピーされます。

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

このプロパティでは、しきい値の有効な値は 60 から 2048 までの範囲です。

注-すべてのデータリンクは、汎用名を使用して自動的に命名されます。ベースとなる NIC が `e1000g` であるデータリンクに対してこの非公開プロパティが構成されていることを確認してください。プロパティを設定する前に、`dladm show-phys` を使用して確認してください。

公開プロパティの構成と同じく、非公開プロパティの値を変更する前に IP インタフェースも削除する必要があります。

たとえば、次のような手順を実行します。

```
# 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
```

割り込みレートの手動設定

`e1000g` ドライバによって割り込みが発行されるレートを制御するプロパティは、ネットワークやシステムのパフォーマンスにも影響を与えます。通常、パケットごとに割り込みを生成することによって、ネットワークパケットがスタックの上位層に配信されます。また、割り込みレートはデフォルトで、カーネル内の GLD 層によって自動的に調整されます。ただし、ネットワークトラフィックの状態によっては、このモードが望ましくない場合があります。この問題の詳細については、1996 年の USENIX 技術会議で発表されたこのドキュメント (<http://www.stanford.edu/class/cs240/readings/mogul.pdf>) を参照してください。したがって、特定の環境でより良いパフォーマンスを得るには、割り込みレートの手動設定が必要になります。

割り込みレートを定義するには、次のプロパティを設定します。

- `_intr_throttling_rate` は、ネットワークトラフィックの状態にかかわらず、割り込み表明間の遅延を決定します。
- `_intr_adaptive` は、割り込みスロットリングレートの自動チューニングを有効にするかどうかを決定します。デフォルトでは、このプロパティは有効になっています。

最初に、割り込みスロットリングレートの自動チューニングをオフにします。次に、割り込みスロットリングレートプロパティを手動で設定します。

割り込みスロットリングレートを変更する必要がある `e1000g` NIC を含む x86 ベースのシステムがあるとします。また、`e1000g0` のデータリンク名が `net1` であるとします。次のコマンドを入力します。

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

データリンクプロパティに関するステータス情報の取得

データリンクプロパティに関する情報を取得するには、次のいずれかのコマンドを使用できます。

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

データリンクプロパティの表示 (`dladm show-linkprop`)

この方法については、[32 ページ](#)の「データリンクプロパティのカスタマイズ」で説明しています。データリンクプロパティの完全な一覧を表示するには、プロパティを指定せずにコマンドを入力します。例:

```
# 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
```

Ethernet プロパティ値の表示 (`dladm show-ether`)

`dladm show-ether` コマンドでオプションを使用しなかった場合は、データリンクの現在の Ethernet プロパティ値のみが表示されます。デフォルトで提供されるものより詳細な情報を取得するには、`-x` オプションを使用します。次は、このコマンドの使用方法を示す例です。

```
# 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

```

-x オプションを使用すると、このコマンドは指定されたリンクの組み込み機能や、ホストとリンクパートナーとの間で現在通知されている機能も表示します。次は、直前の例で表示された情報についての説明です。

- Ethernet デバイスの現在の状態については、リンクは稼働しており、毎秒 1 ギガビットの全二重で機能しています。その自動ネゴシエーション機能が有効化されており、ホストとリンクパートナーの両方が一時停止フレームを送受信できる双方向フロー制御が備わっています。この情報は、出力の 1 行目に表示されています。
- 後続の行には、データリンク速度機能に関する情報、通知される実際のデータリンク速度、およびピアシステムからの情報が次のように表示されています。
 - Ethernet デバイスの機能が一覧表示されています。ネゴシエーションタイプは自動的に設定できます。さらに、このデバイスは毎秒 1 ギガビット、毎秒 100 メガビット、および毎秒 10 メガビットの速度を、全二重、半二重の両方でサポートできます。同様に、一時停止フレームはホストとリンクパートナーとの間でどちらの方向にも送受信できます。
 - net1 の機能が次のように通知されています: 自動ネゴシエーション、速度 - 二重、一時停止フレームのフロー制御。
 - 同様に、net1 のリンクパートナーつまりピアパートナーが、次の機能を通知しています: 自動ネゴシエーション、速度 - 二重、一時停止フレームのフロー制御。

dladm コマンドによるその他の構成タスク

このセクションでは、プライマリインタフェースの切り替えや動的再構成 (DR) の実行など、dladm コマンドを使用することで単純化されたその他の構成手順について説明します。

▼ システム上のプライマリインタフェースを切り替える方法

システムのプライマリインタフェースの変更は、データリンクの名前を変更するときに行います。次の手順は、次のシステム構成に基づいています。

- システムに net0 と net1 という 2 つのデータリンクがあります。
- ベースとなる NIC は、それぞれ e1000g0 と nge0 です。
- net0 上に IP インタフェースが構成されています。この IP インタフェースには、常にベースとなるデータリンクの名前が付けられます。

システムのプライマリインタフェースは、そのインスタンス番号である `0` に基づいて `net0` です。プライマリインタフェースは `e1000g0` 上に構成されています。次の手順では、`nge0` 上のデータリンク構成をプライマリインタフェースの構成にする方法を示します。

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 システムのデータリンクの物理属性を表示します。

```
# dladm show-phys
```

- 3 プライマリ IP インタフェースを削除します。

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

注 - `ipadm` コマンドの詳細は、第4章「IP インタフェースの操作」および `ipadm(1M)` のマニュアルページを参照してください。

- 4 プライマリリンクの名前を、システム上のほかのデータリンクで使用されていない名前に置き換えます。

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

- 5 プライマリリンク名を、プライマリデバイスになるように指定されたデータリンクに割り当てます。

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

例 3-1 プライマリインタフェースの切り替え

次の例は、システム上のプライマリインタフェースを変更する操作に含まれるすべての手順を組み合わせたものです。この例の最後では、`e1000g0` 上に構成されたプライマリインタフェースが `nge0` 上に構成されたインタフェースに置き換わっています。プライマリリンクを別の NIC に切り替えたあとは、新しい NIC のデータリンク上にインタフェースを構成できます。

```
# 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
```

▼ 動的再構成を使用してネットワークインタフェースカードを交換する方法

この手順は、動的再構成 (DR) をサポートするシステムにのみ適用されます。特に、DR の完了後の構成手順を示します。Oracle Solaris 11 では、DR の完了後にネットワークリンクを構成し直す必要はなくなりました。代わりに、取り外した NIC のリンク構成を交換した NIC に移動するだけです。

この手順では、DR 自体の詳しい実行手順は示しません。これらの手順については、システムのドキュメントを参照してください。

DR の概要については、『Oracle Solaris 11.1 の管理: デバイスとファイルシステム』の第 4 章「デバイスの動的構成 (タスク)」を参照してください。

始める前に DR の実行手順はシステムのタイプによって異なります。まず、次を必ず完了してください。

- システムが DR をサポートしていることを確認します。
- システムでの DR を説明した適切なマニュアルを参照します。

Oracle 製 Sun サーバーの DR に関する最新ドキュメントを見つけるには、<http://www.oracle.com/technetwork/indexes/documentation/index.html> で「動的再構成 (dynamic reconfiguration)」を検索します。

Oracle Solaris Cluster 環境での DR の実行については、『Oracle Solaris Cluster システム管理』を参照してください。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 (オプション) データリンクの物理属性や各データリンクのシステム上の場所に関する情報を表示します。

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

dladm show-phys -L によって表示される情報のタイプの詳細については、[dladm\(1M\)](#) のマニュアルページを参照してください。

3 システムのドキュメントの説明に従って DR を実行します。

この手順を実行するには、システムの DR ドキュメントを参照してください。

交換用 NIC を取り付けたら、次の手順に進みます。

- 4 該当する状況に応じて、次のいずれかの手順を実行します。
 - 古い NIC と同じスロットに交換用 NIC を挿入した場合は、手順 5 に進みます。
古い NIC が以前に占有していたのと同じ場所を新しい NIC が使用しているため、新しい NIC が古い NIC のリンク名と構成を継承します。
 - 交換用 NIC を別のスロットに挿入し、新しい NIC が取り外された NIC のデータリンク構成を継承する必要がある場合は、次を入力します。

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

new-datalink 古い NIC が取り外された場所とは異なるスロットに挿入された交換用 NIC のデータリンクを示します。

old-datalink 取り外された古い NIC に関連付けられているデータリンク名を示します。

注- このシナリオでは、古い NIC が取り外されたスロットは空けておく必要があります。

たとえば、スロット 1 の NIC が取り外され、スロット 2 に新しい NIC が挿入されています。スロット 1 には NIC が挿入されていません。スロット 1 上のデータリンクが `net0` で、スロット 2 上のデータリンクが `net1` であるとします。新しい NIC のデータリンクが古い NIC のデータリンク構成を継承するためには、次を入力します。

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

- 5 新しい NIC リソースが Oracle Solaris によって使用可能になるようにして、DR プロセスを完了します。
たとえば、`cfgadm` コマンドを使用して NIC を構成できます。詳細は、`cfgadm(1M)` のマニュアルページを参照してください。
- 6 (オプション) リンクの情報を表示します。
`dladm show-phys` または `dladm show-link` のいずれかを使用して、データリンクに関する情報を表示できます。

例 3-2 新しいネットワークカードを取り付けることによる動的再構成の実行

この例は、リンク名 `net0` の `bge` カードを `e1000g` カードに交換する方法を示しています。`e1000g` がシステムに接続されると、`net0` のリンク構成が `bge` から `e1000g` に転送されます。

```
# 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
```

管理者は、`cfgadm` を使用して `bge` を取り外し、その代わりに `e1000g` を取り付けるなど、DR 固有の手順を実行します。カードを取り付けると、`e1000g0` のデータリンクは自動的に名前 `net0` を引き継ぎ、そのリンク構成を継承します。

```
# 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    ---
```

IP インタフェースの操作

この章では、`ipadm` コマンドと、IP インタフェースに対するこのコマンドの使用方法について説明します。`ipadm` コマンドとその利点の概要については、16 ページの「`ipadm` コマンド」を参照してください。

次の内容について説明します。

- 43 ページの「基本的な `ipadm` コマンド」
- 45 ページの「IP インタフェースのプロパティの設定」
- 47 ページの「IP アドレスのプロパティの設定」
- 48 ページの「TCP/IP プロトコルのプロパティの設定」
- 55 ページの「IP インタフェースとアドレスの監視」

基本的な `ipadm` コマンド

23 ページの「IP インタフェースを構成する方法」では、3つの主要な `ipadm` サブコマンドを紹介しました。

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

このセクションでは、IP インタフェースに対する `ipadm` コマンドのその他の選択された用途について説明します。この一覧は完全なものではありません。`ipadm` コマンドと使用可能なすべてのサブコマンドおよびオプションの完全な説明については、`ipadm(1M)` のマニュアルページを参照してください。

IP インタフェース構成の削除 (`ipadm delete-ip`)

データリンク上に構成された IP インタフェースを削除するには、このコマンドを使用します。このコマンドは、特定のデータリンク構成を行うときに特に重要で

す。たとえば、データリンク上に IP インタフェースが構成されていると、そのデータリンクの名前変更が失敗します。データリンクの名前を変更する前に、まず `ipadm delete-ip` を実行する必要があります。

通常、このコマンドはほかの `ipadm` および `dladm` サブコマンドと一緒に使います (システムのプライマリインタフェースの変更など)。このタスクでは、インタフェースを削除し、リンクの名前を変更してから、名前が変更されたデータリンク上にインタフェースを再構成する必要があります。順序は次のとおりです。

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

31 ページの「データリンクの名前変更 (`dladm rename-link`)」のプライマリインタフェースを変更する例を参照してください。データリンク名の変更後に IP インタフェースを再構成するには、23 ページの「IP インタフェースを構成する方法」を参照してください。

IP インタフェース構成の無効化 (`ipadm disable-ip`)

デフォルトでは、`ipadm create-ip` で IP インタフェースを作成したときに、そのインタフェースに `UP` フラグが設定され、そのインタフェースがアクティブな構成の一部になります。アクティブな構成を破棄せずに構成からインタフェースを削除するには、`ipadm disable-ip` サブコマンドを使用します。このコマンドは、特定のインタフェースに `DOWN` フラグを設定します。

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

IP インタフェースを使用可能にし、そのフラグを `UP` にするには、次を入力します。

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

ヒント-インタフェースのステータスを表示するには、`ipadm` を使用します。56 ページの「IP インタフェースに関する情報の取得」を参照してください。

インタフェースのアドレスの削除 (`ipadm delete-addr`)

このコマンドは、IP インタフェースの特定のアドレス構成を削除します。このコマンドは、特定のインタフェースの IP アドレスを変更するときに便利です。新しいアドレス構成を割り当てる前に、元のアドレス構成を削除する必要があります。次の一般的な手順を実行します。

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

インタフェースの IP アドレスの作成例については、23 ページの「[IP インタフェースを構成する方法](#)」を参照してください。

注-1つのインタフェースに複数のアドレスを構成できます。各アドレスは、アドレスオブジェクトによって識別されます。確実に正しいアドレスを削除するには、アドレスオブジェクトを知っている必要があります。システム上のインタフェースアドレスを表示するには、`ipadm show-addr` サブコマンドを使用します。アドレスオブジェクトの説明については、23 ページの「[IP インタフェースを構成する方法](#)」を参照してください。アドレスの表示の詳細は、59 ページの「[IP アドレスに関する情報の取得](#)」を参照してください。

IP インタフェースのプロパティの設定

このセクションでは、`ipadm` コマンドを使用して選択した IP インタフェースのプロパティを設定する方法について説明します。

IP インタフェースには、データリンクと同様に、特定のネットワーク環境に合わせてカスタマイズできるプロパティがあります。各インタフェースには2つのプロパティセットが存在します。1つは IPv4 プロトコル用のセットで、もう1つは IPv6 プロトコル用のセットです。MTU などの一部のプロパティは、データリンクと IP インタフェースの両方に共通です。そのため、データリンクに対してある MTU 値を構成し、そのリンク上に構成されているインタフェースに対して別の MTU 値を構成できます。さらに、その IP インタフェースをたどる IPv4 パケットと IPv6 パケットに適用される異なる MTU 値を構成できます。

IP インタフェースのプロパティを設定するには、3つの `ipadm` サブコマンドを使用します。

- `ipadm show-ifprop -p property interface` サブコマンドは、IP インタフェースのプロパティとそれらの現在値を表示します。`-p property` オプションを使用しなかった場合は、IP インタフェースのすべてのプロパティが一覧表示されます。IP インタフェースを指定しなかった場合は、すべての IP インタフェースのすべてのプロパティが一覧表示されます。
- `ipadm set-ifprop -p property=value interface` サブコマンドは、IP インタフェースのプロパティに値を割り当てます。
- `ipadm reset-ifprop -p property interface` サブコマンドは、特定のプロパティをそのデフォルト値にリセットします。

パケット転送の有効化

ネットワーク内で、あるホストは、別のホストシステムに宛てられたデータパケットを受信できます。受信側のローカルシステムでパケット転送を有効にすることによって、そのシステムは、データパケットを宛先ホストに転送できます。デフォルトでは、IP 転送は無効になっています。

パケット転送は、IP インタフェースと TCP/IP プロトコルのどちらにも設定できるプロパティによって管理されます。パケットの転送方法を選択できるようにする場合は、IP インタフェース上でパケット転送を有効にします。たとえば、複数の NIC を備えたシステムがあるとしたら、一部の NIC が外部ネットワークに接続されているのに対して、その他の NIC はプライベートネットワークに接続されています。そのため、すべてのインタフェースではなく、一部のインタフェース上でのみパケット転送を有効にします。

TCP/IP プロトコルのプロパティを設定することで、システム上でグローバルにパケット転送を有効化することもできます。[49 ページの「グローバルなパケット転送の有効化」](#)を参照してください。

注 - IP インタフェースとプロトコルの forwarding プロパティは、どちらも排他的ではありません。インタフェースとプロトコルのプロパティを同時に設定できます。たとえば、プロトコル上でグローバルにパケット転送を有効にしてから、システム上の IP インタフェースごとにパケット転送をカスタマイズできます。したがって、そのシステムではパケット転送はグローバルに有効ですが、選択的でもあります。

IP インタフェース上でパケット転送を有効にするには、次のコマンドを使用します。

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

ここで、*protocol-version* は IPv4 または IPv6 のどちらかです。このコマンドは、IPv4 パケットと IPv6 パケットで別個に実行する必要があります。

次は、システム上で IPv4 パケット転送のみを有効にする方法の例です。

```
# 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
...
```

IPアドレスのプロパティの設定

ipadm コマンドを使用すると、IPアドレスがインタフェースに割り当てられたあとで、それらのアドレス固有のプロパティを設定できます。これらのプロパティを設定することによって、次のことを決定できます。

- ネットマスクの長さ
- IPアドレスをアウトバウンドパケットの発信元アドレスとして使用できるかどうか
- アドレスが大域ゾーンまたは非大域ゾーンのどちらに属するか
- アドレスがプライベートアドレスであるかどうか

IPアドレスのプロパティを操作するときは、次の ipadm サブコマンドを使用します。

- ipadm show-addrprop [-p *property*] [*addrobj*] サブコマンドは、使用するオプションに応じてアドレスのプロパティを表示します。
すべてのIPアドレスのプロパティを一覧表示するには、プロパティおよびアドレスオブジェクトを指定しません。1つのプロパティの値をすべてのIPアドレスについて一覧表示するには、そのプロパティのみを指定します。特定のアドレスオブジェクトのすべてのプロパティを一覧表示するには、アドレスオブジェクトのみを指定します。
- ipadm set-addrprop -p *property=value addrobj* サブコマンドは、アドレスのプロパティに値を割り当てます。一度に設定できるアドレスのプロパティは1つだけです。
- ipadm reset-addrprop -p *property addrobj* サブコマンドは、アドレスのプロパティをデフォルト値に戻します。

注-特定のインタフェースのIPアドレスを変更する場合は、set-addressprop サブコマンドを使用しないでください。代わりに、そのアドレスオブジェクトを削除し、新しいIPアドレスを使用して新しいオブジェクトを作成します。[44 ページ](#)の「[インタフェースのアドレスの削除 \(ipadm delete-addr\)](#)」を参照してください。

たとえば、あるIPアドレスのネットマスクを変更するとします。このIPアドレスは、IPインタフェース net3 上に構成され、アドレスオブジェクト名 net3/v4 で識別されます。次のコマンドは、ネットマスクの改訂方法を示しています。

```
# 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
```

TCP/IP プロトコルのプロパティの設定

プロトコルのプロパティ(チューニング可能パラメータとも呼ばれる)を構成するには、`ipadm` コマンドを使用します。`ipadm` は、チューニング可能パラメータを設定するために以前のリリースで一般的に使用されていた `ndd` コマンドを置き換えます。

TCP/IP プロパティは、インタフェースベースまたはグローバルのどちらにもできます。プロパティを特定のインタフェースに適用することも、ゾーン内のすべてのインタフェースにグローバルに適用することもできます。グローバルなプロパティは、非大域ゾーンごとに異なる値にすることができます。サポートされるプロトコルのプロパティの一覧については、[ipadm\(1M\)](#) のマニュアルページを参照してください。

ネットワークが機能するには、通常、TCP/IP インターネットプロトコルのデフォルト値で十分です。ただし、デフォルト値が特定のネットワークポートに対して十分でない場合は、必要に応じてこれらのプロパティをカスタマイズできます。

TCP/IP インタフェースのプロパティを設定するには、3つの `ipadm` サブコマンドを使用します。

- `ipadm show-prop -p property protocol` コマンドは、プロトコルのプロパティとそれらの現在値を表示します。`-p property` オプションを使用しなかった場合は、プロトコルのすべてのプロパティが一覧表示されます。プロトコルを指定しなかった場合は、すべてのプロトコルのすべてのプロパティが一覧表示されます。
- `ipadm set-prop -p property=value protocol` サブコマンドは、IP インタフェースのプロパティに値を割り当てます。
- `ipadm reset-prop -p property protocol` サブコマンドは、特定のプロトコルのプロパティをそのデフォルト値にリセットします。

注- プロパティが複数の値を受け取ることができる場合は、次のように += 修飾子を使用してプロパティに複数の値を割り当てます。

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

プロパティの値のセットから1つの値を削除するには、次のように -= 修飾子を使用します。

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

グローバルなパケット転送の有効化

46 ページの「パケット転送の有効化」では、インタフェース上でパケット転送を有効にする方法を示します。IP インタフェースのプロパティでパケット転送を設定すると、この機能を選択的に実装できます。このプロパティは、システム上の特定のインタフェース上でのみ有効にできます。

IP インタフェースの数に関係なく、システム全体でパケット転送を有効にするには、プロトコルのプロパティを使用します。プロトコルでのプロパティ名は、IP インタフェースと同じ `forwarding` です。このコマンドは、IPv4 プロトコルと IPv6 プロトコルでパケット転送を有効にするために別個に実行する必要があります。

次の例は、システム上のすべての IPv4 および IPv6 トラフィックに対してパケット転送を有効にする方法を示しています。

```
# 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 インタフェースとプロトコルの `forwarding` プロパティは、どちらも排他的ではありません。インタフェースとプロトコルのプロパティを同時に設定できます。たとえば、プロトコル上でグローバルにパケット転送を有効にしてから、システム上の IP インタフェースごとにパケット転送をカスタマイズできます。したがって、そのシステムではパケット転送はグローバルに有効ですが、選択的でもあります。

特権ポートの設定

TCP、UDP、SCTP などのトランスポートプロトコルでは、ポート 1-1023 はデフォルトの特権ポートであり、`root` アクセス権で実行されるプロセスのみがこれらのポートにバインドできます。`ipadm` コマンドを使用することにより、この指定されたデフォルト範囲を超えて、あるポートを特権ポートになるように予約することができます。したがって、`root` プロセスのみがそのポートにバインドできます。特権ポートを設定するには、トランスポートプロトコルの次のプロパティをカスタマイズします。

- `smallest_nonpriv_port` - 通常のユーザーがバインドできるポート番号の範囲を示す値を持つプロパティ。指定されたポートがこの範囲内にある場合は、そのポートを特権ポートとして設定できます。プロパティの値を表示するには、`ipadm show-prop` コマンドを使用します。
- `extra_priv_ports` - どのポートに特権があるかを示すプロパティ。制限するポートを指定するには、`ipadm set-prop` サブコマンドを使用します。このプロパティには複数の値を割り当てることができます。

たとえば、TCP ポート 3001 および 3050 を、`root` ユーザーのみにアクセスを制限する特権ポートとして設定するとします。`smallest_nonpriv_port` プロパティは、1024 が非特権ポートの最小ポート番号であることを示しています。したがって、指定されたポート 3001 および 3050 は特権ポートに変更できます。次のようなコマンドを実行して処理を進めます。

```
# 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  2049,4045  1-65535
                                     3001,3050
```

特権ポートからいずれかのポート(たとえば、4045)を削除するには、次のコマンドを入力します。

```
# 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    3001,3050    2049,4045    1-65535
                                     3050
```

マルチホームホストに対する対称ルーティングの実装

デフォルトでは、複数のインタフェースを備えたシステム(マルチホームホストとも呼ばれる)は、ルーティングテーブル内のトラフィックの宛先への最長一致ルートに基づいてネットワークトラフィックをルーティングします。宛先への長さが等しいルートが複数存在する場合、Oracle Solaris はこれらのルートにトラフィックを分散させるために等コストマルチパス (ECMP) アルゴリズムを適用します。

この方法でのトラフィックの分散が、場合によっては最適でないことがあります。ある IP パケットが、そのパケット内の IP 発信元アドレスと同じサブネット上に存在しないマルチホームホスト上のインタフェース経由で送信される可能性があります。さらに、送信パケットが特定の受信要求への応答 (ICMP エコー要求など) である場合は、要求と応答が同じインタフェースをたどらない可能性があります。トラフィックのこのようなルーティング構成を非対称ルーティングと呼びます。インターネットサービスプロバイダが RFC 3704 (<http://rfc-editor.org/rfc/bcp/bcp84.txt>) に記載されているインGRESS フィルタリングを実装している場合は、非対称ルーティング構成のために、送信パケットがプロバイダによって破棄されることがあります。

RFC 3704 は、インターネット全体にわたるサービス拒否攻撃の制限を目的としています。この目的に従うには、ネットワークを対称ルーティング用に構成する必要があります。Oracle Solaris では、IP の `hostmodel` プロパティを使用して、この要件を満たすことができます。このプロパティは、マルチホームホスト経由で受信または送信されている IP パケットの動作を制御します。

`hostmodel` プロパティは、3つの値のいずれかに設定できます。

<code>strong</code>	RFC 1122 で定義されている強い終端システム (ES) モデルに対応します。この値によって、対称ルーティングが実装されます。
<code>weak</code>	RFC 1122 で定義されている弱い ES モデルに対応します。この値では、マルチホームホストは非対称ルーティングを使用します。
<code>src-priority</code>	優先ルートを使用してパケットルーティングを構成します。ルーティングテーブル内に複数の宛先ルートが存在する場合、優先ルートは、送信パケットの IP 発信元アドレスが構成され

ているインタフェースを使用するルートです。このようなルートが存在しない場合、送信パケットは、そのパケットの IP 宛先への一致する最長のルートを使用します。

次の例は、マルチホームホストに対して IP パケットの対称ルーティングを実装する方法を示しています。

```
# 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
ipv4 hostmodel rw strong -- weak strong,
src-priority,
weak
```

トラフィックの輻輳制御の実装

ネットワークの輻輳は通常、ネットワークが対応できる量を上回るパケットをノードが送信したときに、ルーターバッファオーバーフローの形で発生します。トラフィックの輻輳は、さまざまなアルゴリズムによって送信側システムに対する制御を設定することで回避されます。これらのアルゴリズムは、Oracle Solaris でサポートされており、オペレーティングシステムに容易に追加したり、直接プラグインしたりできます。

次の表では、サポートされているアルゴリズムを一覧表示し、説明します。

アルゴリズム	Oracle Solaris 名	説明
NewReno	newreno	Oracle Solaris のデフォルトアルゴリズム。制御メカニズムには、送信側の輻輳ウィンドウ、スロースタート、および輻輳回避があります。
HighSpeed	highspeed	高速ネットワーク用のもっとも有名かつシンプルな NewReno の修正版の 1 つ。
CUBIC	cubic	Linux 2.6 の現在のデフォルトアルゴリズム。輻輳回避フェーズを線形的なウィンドウ増加から cubic 関数に変更します。
Vegas	vegas	実際のパケットロスを発生させずに輻輳を予測しようとする典型的な遅延ベースのアルゴリズム。

輻輳制御を有効にするには、次に示す制御関連の TCP プロパティを設定します。これらのプロパティは TCP について表示されますが、これらのプロパティによって有効になる制御メカニズムは、SCTP トラフィックにも適用されます。

- `cong_enabled` - システムで現在動作中のアルゴリズムをコマンドで区切ったリストが格納されます。アルゴリズムを追加または削除して、使用するアルゴリズムのみを有効にすることができます。このプロパティには複数の値を設定できます。このため、行う変更に応じて `+=` 修飾子または `-=` 修飾子を使用する必要があります。
- `cong_default` - アプリケーションがソケットオプション内でアルゴリズムを明示的に指定しない場合にデフォルトで使用されるアルゴリズム。現在、`cong_default` プロパティの値は大域ゾーンと非大域ゾーンの両方に適用されます。

プロトコルに輻輳制御のアルゴリズムを追加するには、次のコマンドを実行します。

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

アルゴリズムを削除するには、次のコマンドを実行します。

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

デフォルトのアルゴリズムを置き換えるには、次のコマンドを実行します。

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

注 - アルゴリズムを追加または削除するときに従う順序の規則はありません。ほかのアルゴリズムをプロパティに追加する前に、アルゴリズムを削除できます。ただし、`cong_default` プロパティには定義済みのアルゴリズムが常に設定されている必要があります。

次の例は、輻輳制御を実装するために実行する手順を示しています。この例では、TCP プロトコルのデフォルトのアルゴリズムが `newreno` から `cubic` に変更されています。次に、`vegas` アルゴリズムが有効なアルゴリズムのリストから削除されています。

```
# 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,
                                     highspped,  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 highspeed,vegas
```

TCP 受信バッファサイズの変更

TCP 受信バッファのサイズは、TCP プロパティ `recv_buf` (デフォルトでは 128K バイト) を使用して設定します。しかし、アプリケーションは利用可能な帯域幅を均等に使用しません。このため、接続待ち時間に応じてデフォルトのサイズを変更する必要があります。たとえば、Oracle Solaris の Secure Shell 機能を使用すると、データストリームに対して追加のチェックサム処理や暗号化処理が実行されるため、帯域幅の使用量にオーバーヘッドが発生します。したがって、バッファサイズを増やす必要がある可能性があります。同様に、一括転送を行うアプリケーションが帯域幅を効率的に使用するために、同じバッファサイズ調整が必要です。

次のように帯域幅遅延積 (BDP) を推定することで、使用する正しい受信バッファサイズを計算できます。

$$BDP = \text{available_bandwidth} * \text{connection-latency}$$

接続待ち時間の値を取得するには、`ping -s host` を使用します。帯域幅の使用量を推定するには、`iperf` および `iperf3` ツールを使用します。

適切な受信バッファサイズは、BDP の値に近似します。ただし、帯域幅の使用量はさまざまな条件によって左右されます。インフラストラクチャーの共有や、帯域幅の使用で競合するアプリケーションとユーザーの数によって、その推定値は変化します。

バッファサイズの値を変更するには、次の構文を使用します。

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

次の例は、バッファサイズを 164K バイトに増やす方法を示しています。

```
# 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
```

推奨されるバッファサイズは状況によって異なるため、推奨設定値はありません。固有の条件を持つネットワークごとにBDPの値が異なることを示す、次のケースを考慮してください。

バッファサイズのデフォルト値が128Kバイトである標準的な1Gビット/秒LAN:

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

待ち時間が100ミリ秒である架空の1Gビット/秒WAN:

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

欧州米国間リンク (iperf で帯域幅を測定)

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

BDPを計算できない場合は、次の所見をガイドラインとして使用します。

- LAN経由の一括転送では、バッファサイズの値は128Kバイトで十分です。
- ほとんどのWAN配備では、受信バッファサイズは2Mバイトの範囲内にすべきです。



注意-TCP受信バッファサイズを増やすと、多くのネットワークアプリケーションのメモリーフットプリントも増加します。

IP インタフェースとアドレスの監視

IP インタフェースとそれらのプロパティに関する情報を監視および取得するには、`ipadm` コマンドを使用します。このコマンドを単独で使用すると、システム上のIP インタフェースに関する一般情報が表示されます。ただし、次の構文を使用することにより、サブコマンドを使用して表示する情報を制限することもできます。

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

- インタフェースの情報のみを取得するには、`ipadm show-if` を使用します。
- アドレスの情報のみを取得するには、`ipadm show-addr` を使用します。
- インタフェースのプロパティに関する情報を取得するには、`ipadm show-ifprop` を使用します。
- アドレスのプロパティに関する情報を取得するには、`ipadm show-addrprop` を使用します。

このセクションでは、`ipadm` サブコマンドを使用してインタフェースの情報を取得する方法の例をいくつか示します。`ipadm show-*` コマンドで表示されるすべてのフィールドの説明については、[ipadm\(1M\)](#)のマニュアルページを参照してください。

IP インタフェースに関する一般情報の取得

サブコマンドを付けずに `ipadm` コマンドを使用すると、システムのすべての IP インタフェースに関するデフォルトの情報が提供されます。例:

```
# 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
```

このサンプル出力では、次の情報が提供されています。

- IP インタフェース。
- 各インタフェースのクラス。
- 各インタフェースの状態。
- インタフェースのステータス(「スタンドアロン」の IP インタフェースであるか、別のタイプのインタフェース構成のベースとなるインタフェースであるか)。この例では、`net1` と `net2` は、`UNDER` 列に示されているように `ipmp0` のベースとなるインタフェースです。
- インタフェースに関連付けられたアドレスオブジェクト。アドレスオブジェクトは特定の IP アドレスを識別します。これらのアドレスオブジェクトは、「`NAME`」見出しの下に一覧表示され、インタフェース名と区別するためにインデントされます。
- IP アドレスのタイプ。「`CLASS/TYPE`」見出しの下にインデントされ、`static` や `dhcp` などがあります。
- 「`ADDRESS`」列の下に一覧表示される実際のアドレス。

このように、`ipadm` コマンドはシステムのインタフェースの全体像を提供します。

IP インタフェースに関する情報の取得

IP インタフェースに関する情報を取得するには、`ipadm show-if [interface]` サブコマンドを使用します。インタフェースを指定しない場合は、この情報にシステム上のすべてのインタフェースが含まれます。

コマンド出力の各フィールドは、次の内容を示します。

IFNAME	情報が表示されているインタフェースを示します。
CLASS	インタフェースのクラスを示します。次の4つのいずれかです。 <ul style="list-style-type: none"> ▪ ip はIP インタフェースを示します。 ▪ ipmp はIPMP インタフェースを示します。 ▪ vni は仮想インタフェースを示します。 ▪ loopback は、自動的に作成されるループバックインタフェースを示します。ループバックインタフェースを除き、残りの3つのインタフェースクラスは手動で作成できます。
STATE	インタフェースのステータスを示します。ok、offline、failed、down、disabled のいずれかです。 <p>failed のステータスはIPMP グループに適用され、ダウンしているためにトラフィックをホストできないデータリンクまたはIP インタフェースを示す場合があります。IP インタフェースがIPMP グループに属している場合、IPMP インタフェースは、グループ内のほかのアクティブなIP インタフェースを使用してトラフィックを引き続き送受信できます。</p> <p>down のステータスは、管理者によってオフラインに切り替えられたIP インタフェースを示します。</p> <p>disable のステータスは、ipadm disable-if コマンドを使用して unplumb されているIP インタフェースを示します。</p>
ACTIVE	インタフェースがトラフィックをホストするために使用されているかどうかを示し、yes または no のどちらかに設定されます。
OVER	インタフェースのIPMP クラスにのみ適用され、IPMP インタフェースまたはグループを構成するベースとなるインタフェースを示します。

次は、このコマンドによって提供される情報の例です。

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

IP インタフェースのプロパティに関する情報の取得

IP インタフェースのプロパティに関する情報については、`ipadm show-ifprop [interface]` コマンドを使用します。プロパティまたはインタフェースを指定しなかった場合は、システム上のすべての IP インタフェースのすべてのプロパティに関する情報が提供されます。

コマンド出力の各フィールドは、次の内容を示します。

IFNAME	情報が表示されている IP インタフェースを示します。
PROPERTY	インタフェースのプロパティを示します。インタフェースには、複数のプロパティが含まれる場合があります。
PROTO	プロパティが適用されるプロトコル (IPv4 または IPv6 のどちらか) を示します。
PERM	指定されたプロパティの許可されるアクセス権を示します。読み取り専用、書き込みのみ、またはその両方のいずれかです。
CURRENT	アクティブな構成内のプロパティの現在値を示します。
PERSISTENT	システムがリブートされたときに再適用されるプロパティの値を示します。
DEFAULT	指定されたプロパティのデフォルト値を示します。
POSSIBLE	指定されたプロパティに割り当てることができる値の一覧を示します。数値の場合は、受け入れ可能な値の範囲が表示されます。

注 - いずれかのフィールド値が不明である場合 (たとえば、情報を要求されているプロパティがインタフェースでサポートされていない場合)、その値は疑問符 (?) として表示されます。

次は、`ipadm show-ifprop` サブコマンドによって提供される情報の例です。

```
# 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
```

IP アドレスに関する情報の取得

IP アドレスに関する情報を取得するには、`ipadm show-addr [interface]` サブコマンドを使用します。インタフェースを指定しなかった場合は、システム上のすべての IP アドレスに関する情報が提供されます。

コマンド出力の各フィールドは、次の内容を示します。

ADDROBJ	IP アドレスが表示されるアドレスオブジェクトを指定します。
TYPE	IP アドレスが <code>static</code> 、 <code>dhcp</code> 、 <code>addrconf</code> のいずれであるかを示します。 <code>addrconf</code> の値は、アドレスがステートレスまたはステートフルアドレス構成を使用して取得されたことを示します。
STATE	アクティブな構成内のアドレスオブジェクトのステータスを示します。これらの値の完全な一覧については、 ipadm(1M) のマニュアルページを参照してください。
ADDR	インタフェース上に構成されている IP アドレスを指定します。このアドレスは IPv4 または IPv6 のどちらかです。トンネルインタフェースでは、ローカルアドレスとリモートアドレスの両方が表示されます。

トンネルの詳細は、『[Oracle Solaris 11.1 ネットワークの構成と管理](#)』の第 6 章「[IP トンネルの構成](#)」を参照してください。

次は、`ipadm show-addr` サブコマンドによって提供される情報の例です。

```
# 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
```

このコマンドにインタフェースを指定し、そのインタフェースに複数のアドレスがある場合は、次のような情報が表示されます。

```
# 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
```

`interface/?` として表示されているアドレスオブジェクトは、そのアドレスが `libipadm` API を使用しなかったアプリケーションによってインタフェース上に構成されたことを示しています。このようなアプリケーションは、アドレスオブジェクト名で `interface/ user-defined-string` という形式が使用されることを要求する `ipadm` コマンドの制御下にはありません。IP アドレスの割り当ての例については、[23 ページ](#)の「[IP インタフェースを構成する方法](#)」を参照してください。

IP アドレスのプロパティに関する情報の取得

IP アドレスのプロパティに関する情報を取得するには、`ipadm show-addrprop [addrobj]` サブコマンドを使用します。すべてのプロパティを一覧表示するには、`addrobj` オプションを省略します。1つのプロパティをすべてのIPアドレスについて一覧表示するには、そのプロパティのみを指定します。特定のアドレスのすべてのプロパティを一覧表示するには、`addrobj` オプションのみを指定します。

コマンド出力の各フィールドは、次の内容を示します。

ADDROBJ	プロパティが表示されているアドレスオブジェクトを示します。
PROPERTY	アドレスオブジェクトのプロパティを示します。アドレスオブジェクトには、複数のプロパティが含まれる場合があります。
PERM	指定されたプロパティの許可されるアクセス権を示します。読み取り専用、書き込みのみ、またはその両方のいずれかです。
CURRENT	現在の構成内のプロパティの実際の値を示します。
PERSISTENT	システムがリブートされたときに再適用されるプロパティの値を示します。
DEFAULT	指定されたプロパティのデフォルト値を示します。
POSSIBLE	指定されたプロパティに割り当てることができる値の一覧を示します。数値の場合は、受け入れ可能な値の範囲が表示されます。

次は、`ipadm show-addrprop` サブコマンドによって提供される情報の例です。

```
# 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           --
```

Oracle Solaris を実行するノートパソコン 上での無線ネットワークの構成

IEEE 802.11 仕様は、ローカルエリアネットワーク用の無線通信を定義しています。これらの仕様と、そこに記載されているネットワークはまとめて「WiFi」と呼ばれます。この用語は、Wi-Fi Alliance 業界グループによって商標登録されています。WiFi ネットワークは、プロバイダと想定クライアントの両方が比較的容易に構成できます。そのため、WiFi ネットワークはますます普及し、世界中で一般的に使用されています。WiFi ネットワークでは、携帯電話、テレビ、およびラジオと同じ電波テクノロジーが使用されます。

注 - Oracle Solaris には、WiFi サーバーまたはアクセスポイントを構成するための機能は含まれていません。

この章の内容は次のとおりです。

- 61 ページの「WiFi 通信のタスクマップ」
- 67 ページの「セキュアな WiFi 通信」

WiFi 通信のタスクマップ

タスク	説明	参照先
WiFi ネットワークに接続します。	ローカルの WiFi ネットワークとの通信を設定して確立します。	62 ページの「WiFi ネットワークに接続する方法」
WiFi リンク上の通信を監視します。	標準の Oracle Solaris ネットワークツールを使用して WiFi リンクの状態を確認します。	66 ページの「WiFi リンクを監視する方法」

タスク	説明	参照先
セキュアな WiFi 通信を確立します。	WEP (Wired Equivalent Privacy) キーを作成し、それを使用してセキュアな WiFi ネットワークとの接続を確立します	68 ページの「暗号化された WiFi ネットワーク接続を設定する方法」

▼ WiFi ネットワークに接続する方法

始める前に ノートパソコンを WiFi ネットワークに接続するには、次の手順を実行します。

1 管理者になります。

詳細は、『Oracle Solaris 11.1の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 データリンクの物理属性を表示します。

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

この例の出力は、2つのリンクが使用可能なことを示しています。デバイス `ath0` 上の `net0` リンクは WiFi 通信をサポートします。 `e1000g0` リンクはシステムを有線ネットワークに接続する役目を果たします。

3 WiFi インタフェースを構成します。

次の手順を使用してインタフェースを構成します。

a. WiFiをサポートするインタフェースを作成します。

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

b. リンクが `plumb` されていることを確認します。

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

4 使用可能なネットワークを確認します。

```
# 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` コマンドのこの出力例では、現在の場所で使用可能な WiFi ネットワークに関する情報が表示されています。この出力には、次の情報が含まれています。

LINK	WiFi 接続で使用されるリンク名を示します。
ESSID	拡張サービスセット識別子を示します。ESSID は WiFi ネットワークの名前であり、特定の無線ネットワークの管理者が任意に設定できます。
BSSID/IBSSID	特定の ESSID の一意の識別子である基本サービスセット識別子を示します。BSSID は、ネットワークに特定の ESSID を提供する、隣接するアクセスポイントの 48 ビットの MAC アドレスです。
SEC	ネットワークにアクセスするために必要なセキュリティのタイプを示します。この値は none または WEP です。WEP については、 67 ページの「セキュアな WiFi 通信」 を参照してください。
STRENGTH	現在の場所で使用可能な WiFi ネットワークからの無線信号の強度を示します。
MODE	ネットワークによって実行される 802.11 プロトコルのバージョンを示します。このモードは a、b、g か、またはこれらのモードの組み合わせです。
SPEED	特定のネットワークの速度 (メガビット/秒) を示します。

5 WiFi ネットワークに接続します。

次のいずれかを行います。

- 信号強度のもっとも高い、セキュリティで保護されていない WiFi ネットワークに接続します。

```
# dladm connect-wifi
```

- ESSID を指定することによって、セキュリティで保護されていないネットワークに接続します。

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

`dladm` の `connect-wifi` サブコマンドには、WiFi ネットワークに接続するためのその他のオプションがいくつかあります。詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

6 インタフェースの IP アドレスを構成します。

次のいずれかを行います。

- DHCP サーバーから IP アドレスを取得します。

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

WiFi ネットワークで DHCP がサポートされていない場合は、次のメッセージが表示されます。

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

- 静的 IP アドレスを構成します。
このオプションは、システムに専用の IP アドレスがある場合に使用します。

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

7 システムの接続先の WiFi ネットワークのステータスを確認します。

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

この例の出力は、システムが現在 ofc ネットワークに接続されていることを示しています。前の手順 4 の scan-wifi の出力は、使用可能なネットワークの中で ofc の信号強度がもっとも高いことを示していました。別のネットワークを直接指定しないかぎり、dladm connect-wifi コマンドは、信号強度のもっとも高い WiFi ネットワークを自動的に選択します。

8 WiFi ネットワーク経由でインターネットにアクセスします。

システムの接続先のネットワークに応じて、次のいずれかを行います。

- アクセスポイントが無料のサービスを提供している場合は、今すぐ目的のブラウザまたはアプリケーションを実行できます。
- アクセスポイントが料金を必要とする商用 WiFi ネットワーク内にある場合は、現在の場所で提供されている手順に従います。通常は、ブラウザを実行し、キーを指定して、ネットワークプロバイダにクレジットカード情報を提供します。

9 セッションを完了します。

次のいずれかを行います。

- WiFi セッションを終了しますが、システムは実行中のままにします。

```
# dladm disconnect-wifi
```

- 現在、複数のセッションが実行されている場合は、特定の WiFi セッションを終了します。

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

ここで、link はセッションに使用されているインタフェースを表します。

- WiFi セッションの実行中に、システムを正常にシャットダウンします。

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

shutdown コマンドを使用してシステムの電源を落とす前に WiFi セッションを明示的に切り離す必要はありません。

例 5-1 特定のWiFi ネットワークへの接続

次の例は、Oracle Solaris ノートパソコンを無線ネットワークに接続するときに行う複数の異なる手順を組み合わせたものです。この例は、OSで任意の無線ネットワークを選択する代わりに、希望する特定の無線ネットワークにシステムを強制的に接続する方法も示しています。この例は、ノートパソコン上に静的IPアドレス 10.192.16.3/24 が構成されていることを前提にしています。この例では、最初にWiFi リンクの可用性を判定しています。

```
# 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
```

ブラウザやその他のアプリケーションを実行して、WiFi ネットワーク経由で作業を開始します。

```
# firefox
```

Firefox ブラウザのホームページが表示されます。

セッションを終了しますが、ノートパソコンは実行中のままにします。

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

show-wifi の出力によって、net0 リンクが WiFi ネットワークから切り離されたことを確認できます。

▼ WiFi リンクを監視する方法

この手順では、標準のネットワークツールを使用して WiFi リンクのステータスを監視し、linkprop サブコマンドを使用して選択したリンクプロパティを変更する方法について説明します。

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 62 ページの「WiFi ネットワークに接続する方法」の説明に従って、WiFi ネットワークに接続します。

- 3 リンクのプロパティを表示します。

構文は次のとおりです。

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

たとえば、net0 無線リンク上で確立された接続のステータスを表示するには、次の構文を使用します。

```
# 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 リンクの固定された速度を設定します。



注意 - Oracle Solaris では、WiFi 接続の最適な速度が自動的に選択されます。初期のリンク速度を変更すると、パフォーマンスが低下したり、特定の WiFi 接続の確立が妨げられたりすることがあります。

リンク速度を、show-linkprop の出力に表示される、速度に指定可能な値のいずれかに変更できます。

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

- 5 リンク上のパケットフローを確認します。

```
# 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
```

例 5-2 リンクの速度の設定

この例は、WiFi ネットワークに接続したあとにリンクの速度を設定する方法を示しています。

```
# 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
```

セキュアなWiFi通信

電波テクノロジーによって WiFi ネットワークが簡単に使用できるように、さらにたいは、多くの場所にいるユーザーから自由にアクセスできるようになっています。その結果、WiFi ネットワークへの接続が安全でなくなる場合があります。ただし、特定のタイプの WiFi 接続は、よりセキュリティーが強化されています。

- アクセスが制限されたプライベート WiFi ネットワークへの接続

企業や大学によって確立された内部ネットワークなどのプライベートネットワークは、各ネットワークへのアクセスを、正しいセキュリティーチャレンジを指定できるユーザーに制限しています。潜在的なユーザーは、接続シーケンス中にキーを指定するか、またはセキュリティー保護された VPN を経由してネットワークにログインする必要があります。
- WiFi ネットワークへの接続の暗号化

セキュリティー保護されたキーを使用して、システムと WiFi ネットワークの間の通信を暗号化できます。WiFi ネットワークへのアクセスポイントは、セキュリティー保護されたキーを生成する機能を備えた、自宅またはオフィスの

ルーターである必要があります。システムとルーターは、セキュリティー保護された接続を作成する前にキーを確立してから共有します。

`dladm` コマンドでは、アクセスポイントを経由した接続を暗号化するための WEP (Wired Equivalent Privacy) キーを使用できます。WEP プロトコルは、無線接続のための IEEE 802.11 仕様で定義されています。`dladm` コマンドの WEP 関連のオプションについての詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

▼ 暗号化された WiFi ネットワーク接続を設定する方法

次の手順では、システムと自宅のルーターの間でセキュアな通信を設定する方法について説明します。家庭向けの無線および有線ルーターの多くには、セキュリティー保護されたキーを生成できる暗号化機能があります。

始める前に 自宅の無線ネットワークに接続する場合は、ルーターが構成され、WEP キーが生成されていることを確認してください。ルーターの製造元のドキュメントに従ってキー構成を生成し、保存します。

1 管理者になります。

詳細は、『[Oracle Solaris 11.1 の管理: セキュリティーサービス](#)』の「[割り当てられている管理権限を使用する方法](#)」を参照してください。

2 WEP キーを含むセキュアオブジェクトを作成します。

システム上に端末ウィンドウを開き、次を入力します。

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

ここで、*keyname* はキーに付ける名前を表します。

3 セキュアオブジェクトに WEP キーの値を指定します。

`create-secobj` サブコマンドは次に、キーの値を要求するスクリプトを実行します。

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

この値は、ルーターによって生成されたキーです。スクリプトは、キーの値として 5 バイトまたは 13 バイトの文字列 (ASCII または 16 進数) のどちらかを受け入れます。

4 今作成したキーの内容を表示します。

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

ここで、*keyname* は、このセキュアオブジェクトの名前です。

- 5 WiFi ネットワークへの暗号化された接続を作成します。

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

- 6 接続がセキュリティーで保護されていることを確認します。

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

SEC の見出しの下の wep の値は、この接続に WEP 暗号化が適用されていることを示します。

例 5-3 暗号化された WiFi 通信の設定

この例では、次のことがすでに実行されていることを前提にしています。

- システムを、WEP キーを作成できるホームルーターに接続します。
- ルーターの製造元のドキュメントに従って WEP キーを作成します。
- キーを保存することにより、それを使用してシステム上にセキュアオブジェクトを作成できるようにします。

セキュアオブジェクトを作成します。

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

ルーターによって生成された WEP キーを指定するときは、入力した値がアスタリスクでマスクされます。

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

直前のコマンドは、セキュアオブジェクト mykey を使用して、WiFi ネットワーク citinet への暗号化された接続を確立します。

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

この出力により、WEP 暗号化を使用して citinet に接続されていることが確認されます。

比較マップ: ifconfig コマンドと ipadm コマンド

ネットワークインタフェースを構成するための ifconfig コマンドは、ipadm コマンドに置き換えられました。Oracle Solaris 11 では、ifconfig コマンドも引き続き機能しますが、ネットワーク構成用のツールとしては ipadm コマンドが推奨されます。ただし、ifconfig の一部のオプションは、ipadm サブコマンドに同等のものはありません。次の表は、ifconfig コマンドの選択されたコマンドオプションと、ipadm コマンドに含まれる同等のものを示しています。

注- この表は、ipadm のオプションの完全な一覧ではありません。完全な一覧については、[ipadm\(1M\)](#) のマニュアルページを参照してください。

表 A-1 ifconfig コマンドと ipadm コマンドの構文の対応

ifconfig コマンド	ipadm コマンド
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

表 A-1 ifconfig コマンドと ipadm コマンドの構文の対応 (続き)

ifconfig コマンド	ipadm コマンド
[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
[deprecated -deprecated] [preferred -preferred] [private -private] [zone zonenumber -zones -all-zones][xmit -xmit]	ipadm set-addprop ipadm reset-addprop ipadm show-addprop
up	ipadm up-addr
down	ipadm down-addr
[metric n] [mtu n] [nud -nud] [arp -arp] [usesrc [name none] [router -router]	ipadm set-ifprop ipadm show-ifprop ipadm reset-ifprop
[ipmp] [group [name ""]] standby -standby] [failover -failover]	ipadm create-ipmp ipadm delete-ipmp ipadm add-ipmp ipadm remove-ipmp ipadm set-ifprop -p [standby] [group]
[interface] [-a]	ipadm ipadm show-if ipadm show-addr
[tdst tunnel-dest-addr] [tsrc tunnel-srcs-addr] [encaplimit n -encaplimit] [thoplimit n]	dladm *-iptun コマンドセット。詳細は、 dladm(1M) のマニュアルページおよび『Oracle Solaris 11.1 ネットワークの構成と管理』の「dladm コマンドによるトンネルの構成と管理」を参照してください。
[auth_algs authentication algorithm] [encr_algs encryption algorithm] [encr_auth_algs encryption authentication algorithm]	ipsecconf 詳細は、 ipsecconf(1M) のマニュアルページおよび『Oracle Solaris 11.1でのネットワークのセキュリティ保護』の第7章「IPsecの構成(タスク)」を参照してください。

表 A-1 ifconfig コマンドと ipadm コマンドの構文の対応 (続き)

ifconfig コマンド	ipadm コマンド
[auth_revarp] [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 オプション - inform、ping、release、status、drop	同等のサブコマンドは現在使用不可。
[modlist] [modinsert <i>mod_name@ pos</i>] [modremove <i>mod_name@pos</i>]	同等のサブコマンドは現在使用不可。

比較マップ: ndd コマンドと ipadm コマンド

パラメータまたはチューニング可能パラメータをカスタマイズするための ndd コマンドは、ipadm コマンドに置き換えられました。Oracle Solaris 11 では、nnd コマンドも引き続き機能しますが、ネットワークパラメータをカスタマイズするためのツールとしては ipadm コマンドが推奨されます。ただし、nnd の一部のオプションには、同等の ipadm サブコマンドがありません。次の表は、nnd コマンドの選択されたコマンドオプションと、ipadm コマンドに含まれる同等のものを示しています。

注- この表は、ipadm のオプションの完全な一覧ではありません。完全な一覧については、[ipadm\(1M\)](#) のマニュアルページを参照してください。

表 B-1 ndd コマンドと ipadm コマンドの構文の対応: プロパティの取得

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

表 B-2 ndd コマンドと ipadm コマンドの構文の対応: プロパティの設定

nnd コマンド	ipadm コマンド
<pre>bash-3.2# ndd -set /dev/ip \ ip_def_ttl 64 bash-3.2# ndd -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 プロパティ, 35

B

BSSID, 63

C

cfgadm コマンド, 41

CIDR 表記法, 23

D

DefaultFixed NCP, 13

DHCP, 25

dladm コマンド, 13-17, 29-32

connect-wifi, 63

delete-phys, 31

help, 15-16

rename-link, 31-32

reset-linkprop, 32-38

scan-wifi, 62

set-linkprop, 32-38

show-ether, 37-38

show-link, 30

show-linkprop, 37-38, 66

show-phys, 30-31

show-wifi, 64

E

ECMP, 51-52

ESSID, 63

/etc/hosts ファイル, 24

Ethernet パラメータ, 37-38

G

GLDv3, 15-16

I

ICMP, 17

ifconfig コマンド, 16

と ipadm コマンド, 71-73

ipadm コマンド, 13-17, 43-60

create-addr, 23

create-ip, 23-27

delete-addr, 44-45

delete-ip, 43-44

disable-ip, 44

help, 16, 17

ifconfig コマンドとの比較, 71-73

ndd コマンドとの比較, 75-77

set-addrprop, 47

set-ifprop, 45

set-prop, 48-55

show-addr, 59

show-addrprop, 47, 60

show-if, 56-57

ipadm コマンド (続き)

show-ifprop, 45, 58

show-prop, 48-55

IP アドレス

DHCP, 25

IPv4 および IPv6, 24

監視, 55-60

削除, 44-45

静的, 23

パケット転送, 46, 49-50

プロパティ, 47, 60

ローカルおよびリモート, 23

IP インタフェース

IPMP インタフェース, 23-27

IP アドレス, 59, 60

IP アドレスの削除, 44-45

IP アドレスの変更, 44-45

IP アドレスの割り当て, 23

MAC アドレスが一意であることの確認, 20-21

TCP/IP プロトコルのプロパティ, 48-55

VNI インタフェース, 23-27

WiFi, 62

アドレスのプロパティ, 47

インタフェース構成の削除, 43-44

インタフェースのプロパティ, 58

インタフェースのプロパティの設定, 45

インタフェースのプロパティの表示, 45

監視, 55-60

構成, 26

作成と plumb, 23-27

特権ポート, 50

パケット転送の有効化, 46, 49-50

表示

IP アドレス, 59

アドレスのプロパティ, 60

一般情報, 25, 56

インタフェース, 56-57

インタフェースのプロパティ, 58

プロトコルのプロパティ, 48

プライマリインタフェースの変更, 43-44

無効化と有効化, 44

IP トンネル, 23

ローカルアドレスとリモートアドレス, 23

IP マルチパスインタフェース (IPMP), 23-27

M

MAC アドレス, 一意であることの確認, 20-21

MTU, 33

N

name-service/switch サービス, 24

NCP, 「ネットワーク構成プロファイル」を参照

nnd コマンド, 17

と ipadm コマンド, 75-77

netadm コマンド, 13-17, 21-23

netcfg コマンド, 13-17

netstat コマンド, WiFi リンク上のパケットフ

ローの確認, 67

NIC ドライバ, 32-33

P

Power Management, 34

R

route コマンド, 26

S

SCTP, 17

STREAMS モジュール, とデータリンク, 34-35

T

TCP 受信バッファサイズ, 54-55

U

UDP, 17

USENIX, 36-37

W

WiFi

- IEEE 802.11 仕様, 61
- WiFi ネットワークへの接続, 62, 63, 64
- WiFi の構成例, 65
- 暗号化された通信の例, 69
- 拡張サービスセット識別子 (ESSID), 63
- 基本サービスセット識別子 (BSSID), 63
- セキュリティー保護された WiFi リンク, 67
- 接続の暗号化, 68
- 定義, 61
- パケットフローの確認, 67
- リンクの監視, 66
- 例、リンク速度の設定, 67

WLAN, 13

あ

- アドレスオブジェクト, 24
- アドレス解決プロトコル (ARP), 17

か

- 外部ネットワーク修飾子 (ENM), 13
- 仮想ネットワークインタフェース (VNI), 23-27
- 仮想ネットワークカード (VNIC), 29-30
- 仮想プライベートネットワーク (VPN), 34-35
- 仮想ローカルエリアネットワーク (VLAN), 29-30

こ

- 固定ネットワーク構成, 11-17
 - 静的 IP アドレス, 12

さ

- サービス管理機能 (SMF), 16

し

- 自動ネゴシエーション, 33
- ジャンボフレーム, サポートの有効化, 33

せ

- セキュリティーの注意点, WiFi, 67
- 全二重, 34

た

- 帯域幅遅延積 (BDP), 54-55
- 対称ルーティング, 51-52
- ダイレクトメモリアクセス (DMA), 35-36

て

- データリンク
 - autopush プロパティ, 35
 - DMA バインディング, 35-36
 - Ethernet パラメータ値, 37-38
 - MTU サイズの変更, 33
 - STREAMS モジュール, 34-35
 - VLAN, 29-30
 - VNIC, 29-30
 - 公開プロパティと非公開プロパティ, 32-33
 - 削除, 31
 - 自動ネゴシエーション, 33
 - 通知速度と有効化速度, 33-34
 - 名前変更, 31-32
 - 汎用名, 31-32
 - 表示
 - 一般情報, 29-30
 - システム上の物理的な位置, 31
 - ネットワークドライバのプロパティ, 37-38
 - 物理属性, 30-31
 - リンク, 30
 - リンクプロパティ, 37-38
 - 物理リンク, 30
 - プロパティの設定, 32-38

データリンク (続き)

- リンクアグリゲーション, 29-30
- リンク上の IP インタフェースの構成, 23-27
- リンク速度, 33-34
- 割り込みレート, 36-37

と

- 動的再構成 (DR), NIC の交換, 40
- 特権ポート, 50

ね

- ネットワークインタフェースカード (NIC), 交換,
DR を使用, 40
- ネットワーク構成ツール, 13-17
 - dladm コマンド, 15-16
 - ipadm コマンド, 16
 - netadm コマンド, 14
 - netcfg コマンド, 14
- ネットワーク構成プロファイル (NCP), 11-17
 - DefaultFixed, 12, 13
 - NCP の一覧表示, 21-23
 - アクティブな NCP, 11-17
 - アクティブな NCP の切り替え, 14, 21-23
 - 固定, 12-13
 - リアクティブ, 12-13

は

- パケット転送
 - インタフェース上の, 46
 - プロトコル上の, 49-50
- 場所プロファイル, 13
- バッファ, 35-36
- 半二重, 34
- 汎用 LAN ドライバ (GLD), 36-37

ふ

- 輻輳制御, 52-54

- プライマリインタフェース, 切り替え, 31-32,
38-42, 43-44
- プロトコル, のプロパティ, 48-55
- プロファイル管理ネットワーク構成, 11-12

ま

- マルチホームホスト, 51-52

む

- 無線インタフェース, 61

り

- リアクティブネットワーク構成, 11-17
 - ENM, 13
 - WLAN, 13
 - 場所プロファイル, 13
- リモートアドレス, 23
- リンクアグリゲーション, 29-30
- リンク速度, 33-34
- リンク名, 31-32

ろ

- ローカルアドレス, 23

わ

- 割り込みレート, 36-37