Solaris のシステム管理 (IP サービス)

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 816–3958–11
2003 年 8 月

原典: *System Administration Guide: IP Services*

Part No: 806-4075-11

Revision A

030605@5943

目次

はじめに

『*Solaris* のシステム管理 *(IP* サービス*)*』は、Solaris™ システム管理に関する重要な情報を提供する、7 巻構成のマニュアルの中の 1 巻です。このマニュアルでは、SunOS™ 5.9 オペレーティングシステムをすでにインストールし、使用するネットワークソフトウェアの設定が済んでいるものと想定しています。SunOS 5.9 オペレーティングシステムは Solaris 製品ファミリの一部であり、Solaris 共通デスクトップ環境 (CDE) などが含まれます。また、SunOS 5.9 は、AT&T System V リリース 4 オペレーティングシステムに準拠しています。

注 – Solaris オペレーティング環境は、SPARC™ と x86 の 2 種類のハードウェア (プラットフォーム) 上で動作します。また、Solaris オペレーティング環境は、64 ビットと 32 ビットの両方のアドレス空間で動作します。このマニュアルで説明する情報は、章、節、注、箇条書き、図、表、例、またはコード例において特に明記しないかぎり、両方のプラットフォームおよびアドレス空間に該当します。

対象読者

このマニュアルは、Solaris 9 リリースを実行するシステムの管理者を対象にしています。このマニュアルを使いこなすには、UNIX® のシステム管理について 1 ～ 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。

内容の紹介

第 2 章では、TCP/IP とそのコンポーネントの概要について説明します。また、インターネットプロトコル群について紹介します。

第 3 章では、TCP/IP ネットワークの計画方法について説明します。

第 4 章では、ネットワーク上の Solaris オペレーティング環境で TCP/IP を管理する方法について説明します。

第 5 章では、ネットワークへの TCP/IP の実装に関するリファレンス情報を記載しています。

第 7 章では、DHCP とそのコンポーネントの概要について説明します。また、ネットワーク上の Solaris オペレーティング環境での DHCP の動作について説明します。

第 8 章では、ネットワークに DHCP サービスを設定する前に必要な作業について説明します。

第 9 章では、ネットワークで DHCP サービスを構成する方法について説明します。また、DHCP マネージャの使用手順について説明します。

第 10 章では、ネットワークで Solaris DHCP サービスを管理するために必要な作業について説明します。

第 11 章では、DHCP サーバーまたは DHCP クライアントを構成する際に発生する可能性がある問題点の解決に役立つ情報を提供します。

第 12 章では、ネットワーク上の DHCP サービスが使用するコマンドとファイルのリファレンス情報を記載しています。

第 14 章では、IPv6 として知られる新しいインターネットプロトコルの概要について説明します。

第 15 章では、IPv6 や IPv6 ルーターを有効にする方法、IPv6 アドレスを DNS、NIS、NIS+ 用に設定する方法、ルーター間のトンネルの作成方法、IPv6 対応したコマンドを使って診断する方法、IPv6 ネームサービス情報の表示方法について説明します。

第 16 章では、IPv6 の Solaris 実装に伴う概念について説明します。

第 17 章では、IPv4 から IPv6 への移行方法と、標準的な解決法の概要について説明します。

第 19 章では、IP データグラムを保護する新しい IP セキュリティアーキテクチャーの概要について説明します。

第 20 章では、ネットワークに IPsec (IP セキュリティ) を実装する手順について説明します。

第 21 章では、IPsec で使用する IKE (インターネットキー交換) の実装について、その概要と手順を説明します。

第 23 章では、新しいモバイル IP サービスの概要について説明します。モバイル IP サービスは、ラップトップやワイヤレス通信など、モバイルコンピュータとの情報の送受信を有効にします。

第 24 章では、モバイル IP 構成ファイルのパラメータの変更、追加、削除、および表示の方法について説明します。また、モビリティエージェント状態の表示方法についても説明します。

第 25 章では、モバイル IP の Solaris 実装に付属しているコンポーネントについて説明します。

第 27 章では、新しい IP ネットワークマルチパスサービスの概要について説明します。IP ネットワークマルチパスサービスを使うと、同じ IP リンク (たとえば Ethernet) に接続された複数のネットワークインタフェースカードがある場合に負荷分散とフェイルオーバーの両機能を実現できます。

第 28 章では、インタフェースグループを作成および使用するための手順や、検査用 IP アドレス、`hostname` ファイル、マルチパス構成ファイルを構成するための手順について説明します。

用語集では、主要な IP サービス用語の定義を提供します。

『Solaris のシステム管理』全 7 巻の内容

『*Solaris* のシステム管理』全 7 巻の残りの 6 巻で解説される各トピックを以下に示します。

『Solaris のシステム管理 (基本編)』

- 『*Solaris* のシステム管理 (基本編)』の「Solaris 管理ツール (製品概要)」
- 『*Solaris* のシステム管理 (基本編)』の「Solaris Management Console の操作 (手順)」
- 『*Solaris* のシステム管理 (基本編)』の「ユーザーアカウントとグループの管理」
- 『*Solaris* のシステム管理 (基本編)』の「サーバーとクライアントサポートの管理」
- 『*Solaris* のシステム管理 (基本編)』の「システムのシャットダウンとブート」
- 『*Solaris* のシステム管理 (基本編)』の「リムーバブルメディアの管理」

- 『*Solaris* のシステム管理 (基本編)』の「ソフトウェアの管理」
- 『*Solaris* のシステム管理 (基本編)』の「デバイスの管理」
- 『*Solaris* のシステム管理 (基本編)』の「ディスクの管理」
- 『*Solaris* のシステム管理 (基本編)』の「ファイルシステムの管理」
- 『*Solaris* のシステム管理 (基本編)』の「ファイルとファイルシステムのバックアップおよび復元」

『Solaris のシステム管理 (上級編)』

- 『*Solaris* のシステム管理 (上級編)』の「印刷サービスの管理 (トピック)」
- 『*Solaris* のシステム管理 (上級編)』の「端末とモデムの管理 (トピック)」
- 『*Solaris* のシステム管理 (上級編)』の「システム資源の管理 (トピック)」
- 『*Solaris* のシステム管理 (上級編)』の「システムパフォーマンスの管理 (トピック)」
- 『*Solaris* のシステム管理 (上級編)』の「Solaris ソフトウェアで発生する問題の解決」

『Solaris のシステム管理 (資源管理とネットワークサービス)』

- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「システム資源の管理とネットワークサービス (概要)」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「Web キャッシュサーバーの管理」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「システムの時刻関連サービス」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「Solaris 9 リソースマネージャ」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「リモートファイルシステムへのアクセス (トピック)」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「SLP (トピック)」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「メールサービス (トピック)」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「モデム関連ネットワークサービス (トピック)」
- 『*Solaris* のシステム管理 (資源管理とネットワークサービス)』の「リモートシステムの利用 (トピック)」

- 『*Solaris* のシステム管理 *(*資源管理とネットワークサービス*)*』の「ネットワークサービスの監視 (トピック)」

『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』

- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「ネームサービスとディレクトリサービス (概要)」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「ネームサービススイッチ (概要)」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「ドメインネームシステム (概要)」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「ネットワーク情報サービス (NIS) (概要)」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「LDAP ネームサービスの紹介 (概要/リファレンス)」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「NIS+ から LDAP への移行」

『Solaris のシステム管理 (セキュリティサービス)』

- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「セキュリティサービス (概要)」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「認証サービスの使用 (手順)」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「Secure Shell の使用 (手順)」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「Secure Shell の管理 (参照)」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「SEAM について」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「システムセキュリティの管理」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「役割によるアクセス制御 (概要)」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「自動セキュリティ拡張ツールの使用 (手順)」
- 『*Solaris* のシステム管理 *(*セキュリティサービス*)*』の「監査のトピック」

『Solaris のシステム管理 (ネーミングとディレクトリサービス : FNS、NIS+ 編)』

- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)* 』の「ネームサービススイッチ」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)* 』の「NIS+ の紹介」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)* 』の「フェデレーテッド・ネーミング・サービス (FNS)」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)* 』の「NIS から NIS+ への移行」
- 『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)* 』の「エラーメッセージ」

関連マニュアル

以下に、本書で参照している関連マニュアルおよび関連書籍を示します。

- 『*The Whole Internet User's Guide and Catalog*』、Krol, Ed. 著、 O' Reilly & Associates, Inc 発行.、1993 年
- 『*TCP/IP Illustrated, Volume 1, The Protocols*』、Stevens、W. Richard 著、Addison Wesley 発行、1994 年
- 『*Mobile IP Design Principles and Practices*』Perkins, Charles E. 著、Massachusetts、Addison-Wesley Publishing Company 発行、1998 年
- 『*RFC 2002*』、Internet Engineering Task Force (IETF) より。オンライン版は http://ietf.org/rfc.html
- 『*Mobile IP: The Internet Unplugged*』、Solomon、James D. 著、New Jersey、Prentice-Hall, Inc. 発行、1998 年

Sun のオンラインマニュアル

`docs.sun.com` では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、`http://docs.sun.com` です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 **P–1** 表記上の規則

字体または記号	意味	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。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前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。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% **grep '^#define ** **XV_VERSION_STRING'**

コード例は次のように表示されます。

- C シェル

  machine_name% **command y|n** [*filename*]

- C シェルのスーパーユーザー

  machine_name# **command y|n** [*filename*]

- Bourne シェルおよび Korn シェル

  $ **command y|n** [*filename*]

- Bourne シェルおよび Korn シェルのスーパーユーザー

  # **command y|n** [*filename*]

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: `Shift` キーを押します)。ただし、キーボードによっては `Enter` キーが `Return` キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、`Ctrl-D` は `Control` キーを押したまま `D` キーを押すことを意味します。

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。
- このマニュアルでは、「x86」という用語は、Intel 32 ビット系列のマイクロプロセッサチップ、および AMD が提供する互換マイクロプロセッサチップを意味します。

第 1 章

TCP/IP (トピック)

第 2 章	TCP/IP の概要
第 3 章	TCP/IP ネットワークの計画
第 4 章	TCP/IP ネットワークの設定手順と障害追跡手順
第 5 章	TCP/IP のリファレンス情報

第 2 章

TCP/IP (概要)

この章では、Solaris 実装の TCP/IP ネットワークプロトコル群を紹介します。この章の情報は、まだあまり TCP/IP に慣れていないネットワーク管理者を対象としています。 TCP/IP の経験のあるネットワーク管理者の場合は、実行する作業について説明する章だけ参照してもかまいません。

この章では、以下の内容について説明します。

インターネットプロトコル群の概要

この節では、TCP/IP を構成するプロトコルについて詳しく紹介します。ここに示す情報は概念的なものですが、各プロトコルの名前とそれぞれの働きを理解することができます。

TCP/IP は、インターネットプロトコル群を形成するネットワークプロトコルの集合を示す省略名として使用されています。多くの書籍では、「インターネット」という用語は、プロトコル群と広域ネットワークの両方を表すものとして使用されています。本書では、「TCP/IP」は特にインターネットプロトコル群を表し、「インターネット」は広域ネットワークとそれを運営する組織をあらわすものとします。

TCP/IP ネットワークと他のネットワークとを相互接続するには、一意な IP ネットワーク番号を入手する必要があります。本書を作成した時点では、IP ネットワーク番号は、InterNIC と呼ばれる組織によって割り当てられていました。

ネットワーク上のホストがインターネットドメイン名システム (DNS) に参加する場合は、一意なドメイン名を入手し登録する必要があります。InterNIC は、いくつかのトップレベルのドメイン、たとえば .com (商業)、.edu (教育)、.gov (政府) などのドメ

インの傘下にあるドメイン名の登録も行なっています。InterNIC については、第 3 章で詳しく説明します。DNS についての詳細は、『 *Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

プロトコル層と OSI モデル

ほとんどのネットワークプロトコル群は、一連の層として構築されており、これはしばしば総称的にプロトコルスタックと呼ばれます。各層はそれぞれ特定の目的のために設計されていて、送信側ホストと受信側ホストの両方に存在しています。一方のマシンの特定の層が、相手のマシンの対等プロセスが送受信するオブジェクトと同じものを送受信するように設計されています。このような動作は、問題の層の上下の層で進行していることとは独立して行われます。つまり、ホストの各層は、同じマシンの他の層から独立して、他のホストの同じ層と協調して働きます。

OSI 参照モデル

ほとんどのネットワークプロトコル群は層状に構造化されているように見えます。国際標準化機構 (ISO) は構造化された層を使用する開放型相互接続 (OSI) 参照モデルを設計しました。OSI モデルは、ネットワーク活動が 7 つの層から成る構造を持つものと規定しています。それぞれの層に 1 つまたは複数のプロトコルが関連付けられます。層は、連携するネットワーク相互間でのすべての種類のデータ転送に共通するデータ転送操作を表します。

OSI 参照モデルでは、プロトコル層を上 (第 7 層) から下 (第 1 層) へ並べて表します。次の表に OSI 参照モデルを示します。

表 **2–1** OSI (開放型システム間相互接続) 参照モデル

層番号	層の名前	説明
7	アプリケーション	誰でも使用できる標準の通信サービスとアプリケーション
6	プレゼンテーション	情報が解読可能な形で受信側マシンに渡されるようにする
5	セッション	連携コンピュータ間の接続と終了を管理する
4	トランスポート	データの転送を管理する。また、受信されたデータと送信されたデータが同じになることを保証する
3	ネットワーク	ネットワーク間でのデータのアドレス指定と配送を管理する
2	データリンク	ネットワークメディアを通過するデータの転送を取り扱う
1	物理	ネットワークハードウェアの特性を定義する

OSI モデルは、特定のネットワークプロトコル群に特有ではない概念的な動作について定義します。たとえば、OSI ネットワークプロトコル群は、OSI 参照モデルの 7 つの層をすべて実装しています。TCP/IP は、OSI モデル層のいくつかを使用し、その他の層を合併しています。その他のネットワークプロトコル、たとえば SNA では、8 番目の層が追加されています。

TCP/IP プロトコルアーキテクチャモデル

OSI モデルはプロトコルファミリを使用する理想的なネットワーク通信について規定します。TCP/IP は OSI モデルに直接対応していません。TCP/IP は、いくつかの OSI 層を合併して 1 つの層にしたり、一部の層をまったく使用していません。次の表は、Solaris 実装の TCP/IP の層を示しています。最上位層 (アプリケーション) から最下位層 (物理ネットワーク) まで並べてあります。

表 **2–2** TCP/IP プロトコルスタック

OSI 参照の層番号	対応する **OSI** 層	**TCP/IP** 層	**TCP/IP** プロトコルの例
5,6,7	アプリケーション、セッション、プレゼンテーション	アプリケーション	NFS、NIS+、DNS、`telnet`、`ftp`、`rlogin`、`rsh`、`rcp`、 RIP、RDISC、SNMP、その他
4	トランスポート	トランスポート	TCP、UDP
3	ネットワーク	インターネット	IP、ARP、ICMP
2	データリンク	データリンク	PPP、IEEE 802.2
1	物理	物理ネットワーク	Ethernet (IEEE 802.3) トークンリング、RS-232、その他

この表は、TCP/IP プロトコルの層を示しています。また、対応する OSI モデルの層、および TCP/IP プロトコルスタックの各レベルで使用できるプロトコルの例を示しています。通信トランザクションに関与する各ホストは、それぞれ固有の実装によるプロトコルスタックを実行します。

物理ネットワーク層

物理ネットワーク層は、ネットワークに使用するハードウェアの特性を規定します。たとえば、通信メディアの物理特性を規定します。TCP/IP の物理層はハードウェア規格を意味しています。たとえば、Ethernet ネットワークメディアの仕様である IEEE 802.3 や、標準ピンコネクタの仕様である RS-232 などです。

データリンク層

データリンク層は、パケットのネットワークプロトコルの種類を識別します。この場合は TCP/IP です。また、この層には、エラー制御と「フレーミング」の働きもあります。データリンク層の例としては、Ethernet IEEE 802.2 フレーミングと、ポイントツーポイントプロトコル (PPP) フレーミングがあります。

インターネット層

この層はネットワーク層とも呼ばれるもので、ネットワークに対してパケットを受け入れたり、配送したりします。この層には、強力なインターネットプロトコル (IP)、アドレス解決プロトコル (ARP)、インターネットコントロールメッセージプロトコル (ICMP) が組み込まれています。

IP プロトコル

IP プロトコルとそれに関連したルーティングプロトコルは、TCP/IP 群全体の中でたいへん重要なものです。IP は次の機能を受け持ちます。

- *IP* アドレス指定 - IP アドレス指定の規則は IP プロトコルの一部です。IPv4 アドレス指定については、第 3 章で詳しく説明します。IPv6 アドレス指定については、第 14 章で詳しく説明します。
- ホスト間通信 - IP は、受信側ホストの IP アドレスに基づいてパケットが進む経路を決定します。
- パケット形式設定 - IP は、パケットを IP データグラムと呼ばれる単位に組み立てます。データグラムについては、42 ページの「インターネット層」で詳しく説明します。
- フラグメント化 - パケットが大きすぎてネットワークメディアを介して転送できないときは、送信側ホストの IP は、パケットを小さいフラグメントに分割します。受信側ホストの IP は、これらのフラグメントを組み立てて元のパケットに戻します。

前のリリースの Solaris オペレーティング環境では、インターネットプロトコルバージョン 4 (IPv4 と記述される) が実装されています。しかし、インターネットの急速な成長によって、新しいインターネットプロトコルが作成されました。このプロトコルではアドレス空間が拡張されています。バージョン 6 として知られるこの新バージョンは IPv6 と記述されます。Solaris オペレーティング環境では、両方のバージョンを使用することができます。インターネットプロトコルについて言及するときに混乱を避けるため、以下の規則を適用します。

- 用語 IP を使用している説明は、IPv4 と IPv6 の両方に適用されます。
- 用語 IPv4 を使用している説明は、IPv4 のみに適用されます。
- 用語 IPv6 を使用している説明は、IPv6 のみに適用されます。

ARP プロトコル

アドレス解決プロトコル (ARP) は、データリンク層とインターネット層の間に概念的に存在するものです。ARP は、Ethernet アドレス (48 ビット長) を既知の IP アドレス (32 ビット長) にマッピングし、IP はこの情報に基づいてデータグラムを正しい受信側ホストに向けることができます。

ICMP プロトコル

インターネット制御メッセージプロトコル (ICMP) は、ネットワークエラー条件を検出し、報告します。ICMP は以下の事項について報告します。

- 取りこぼしたパケット - 到着が速すぎて処理が間に合わないパケット
- 接続障害 - 到達できない宛先ホスト
- リダイレクト - 送信側ホストに別のルーターを使用させる指示

76 ページの「`ping` コマンド」の節には、エラー検出に ICMP を使用するオペレーティングシステムコマンドについての詳細な説明があります。

トランスポート層

TCP/IP トランスポート層プロトコルは、パケットが正しい順序でエラーなしに到着するようにするために、データ受領の肯定応答を交換し、失われたパケットがあれば転送し直します。この種類の通信を「終端間」通信と呼びます。このレベルのトランスポート層プロトコルは、トランスミッションコントロールプロトコル (TCP) とユーザーデータグラムプロトコル (UDP) です。

TCP プロトコル

TCP は、物理的な回線で接続されているのと同じようにしてアプリケーション相互間の通信ができるようにします。TCP は、独立したパケットの形ではなく、文字単位で転送されているような形でデータを送信します。この転送では、まず開始ポイントで接続がオープンされ、次にバイト順序ですべてのデータが転送され、終了ポイントで接続がクローズされます。

TCP は、転送するデータにヘッダーを添付します。このヘッダーには、送信側マシン上のプロセスが受信側マシン上の対等プロセスに接続できるようにするための、多数のパラメータが含まれています。

TCP は、送信側ホストと受信側ホストとの間に終端間接続を確立することにより、パケットが宛先に到達したことを確認します。したがって、TCP は、「信頼性の高い接続指向型」プロトコルとみなすことができます。

UDP プロトコル

もう 1 つのトランスポート層プロトコルである UDP は、データグラム配送サービスを提供します。UDP は、受信側ホストと送信側ホストとの間の接続の検査は行いません。UDP は接続の確立と検査を省略するので、少量のデータを送信するアプリケーションにとっては、TCP よりも効率的です。

アプリケーション層

アプリケーション層は、誰でも使用できる標準的なインターネットサービスとネットワークアプリケーションを定義します。これらのサービスとトランスポート層の両方の働きにより、データの送受信が行われます。アプリケーション層のプロトコルは多数存在します。以下に、アプリケーション層プロトコルの例を示します。

- 標準 TCP/IP サービス。たとえば、`ftp`、`tftp`、`telnet` コマンドなど
- UNIX の “r” (リモート) コマンド。たとえば、 `rlogin` や `rsh` など
- ネームサービス。たとえば、NIS+ やドメインネームシステム (DNS) など
- ファイルサービス。たとえば NFS サービスなど
- SNMP (ネットワーク管理用プロトコルの一種。Simple Network Mamagement Protocol の略)
- RIP と RDISC ルーティングプロトコル

標準 *TCP/IP* サービス

- *FTP* と匿名 *FTP* - ファイル転送プロトコル (FTP) は、リモートネットワークとの間でファイルを転送します。このプロトコルには、`ftp` コマンド (ローカルマシン) と `in.ftpd` デーモン (リモートマシン) が含まれています。ユーザーは、リモートホストの名前とファイル転送コマンドのオプションを、ローカルホストのコマンド行に指定します。すると、リモートホストの `in.ftpd` デーモンが、ローカルホストからの要求を処理します。`rcp` とは違って、`ftp` は、リモートコンピュータのオペレーティングシステムが UNIX でない場合でも動作します。リモートコンピュータが匿名 FTP を認めるように設定されている場合を除いて、 `ftp` 接続を行うときにはリモートコンピュータにログインする必要があります。

  現在では、インターネットに接続されている匿名 *FTP* サーバーから大量の資料を入手できます。大学その他の研究機関がこれらのサーバーを設定して、ソフトウェア、研究報告、その他の情報をパブリックドメインに公開しています。この種のサーバーにログインするときには、ログイン名として `anonymous` を使用します。「匿名 (anonymous) FTP サーバー」という言葉はこれに由来しています。

  匿名 FTP の使用法と匿名 FTP サーバーの設定については、本書では説明しません。しかし、たとえば『*The Whole Internet User's Guide & Catalog*』など、匿名 FTP について詳しく説明している多数の書籍が市販されています。FTP を使用して標準マシンに到達するための方法については、『*Solaris* のシステム管理 *(*資源管理とネットワークサービス*)*』に説明があります。`ftp(1)` のマニュアルページには、コマンドインタプリタによって呼び出されるすべての `ftp` コマンドオプションに

ついての説明があります。 ftpd(1M) のマニュアルページには、 in.ftpd デーモンが提供するサービスについての説明があります。

- Telnet - Telnet プロトコルは、端末と端末指向プロセスが、TCP/IP を実行するネットワーク上で通信できるようにします。このプロトコルは、telnet プログラム (ローカルマシン上の) と in.telnetd デーモン (リモートマシン上の) として実装されます。Telnet は、2 つのホストが文字単位または行単位で通信できるようなユーザーインタフェースを提供します。アプリケーションにはコマンドのセットが含まれていますが、これについては、telnet(1) のマニュアルページに詳しい説明があります。
- TFTP - 簡易ファイル転送プロトコル (tftp) は ftp に似た機能を備えていますが、ftp の対話型接続を確立する機能はありません。したがって、ユーザーは、ディレクトリの内容を表示したり、ディレクトリを変更したりすることはできません。ユーザーは、コピーするファイルのフルネームを知っている必要があります。tftp のコマンドセットについては、tftp(1) のマニュアルページに説明があります。

UNIX の *"r"* (リモート) コマンド

UNIX の "r" (リモート) コマンドを使用すると、ユーザーは、リモートホストで実行するコマンドを各自のローカルマシンで発行することができます。この種のコマンドには次のものがあります。

- rcp
- rlogin
- rsh

これらのコマンドの使い方については、rcp(1)、rlogin(1)、rsh(1) の各マニュアルページに説明されています。

ネームサービス

Solaris オペレーティング環境では、次のネームサービスを提供しています。

- *DNS* - ドメインネームシステム (DNS) は TCP/IP ネットワーク用にインターネットが提供するネームサービスです。DNS は、ホスト名から IP アドレスに変換するサービスを提供します。また、メール管理用のデータベースとしての働きもします。このサービスの詳細は、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。 resolver(3RESOLV) のマニュアルページも参照してください。
- /etc ファイル - ホストベースの UNIX™ のネーミングシステムは、最初はスタンドアロンの UNIX マシン用に開発された後、ネットワークで使用されるように改良されました。UNIX オペレーティングシステムの旧バージョンの多くや UNIX マシンでは、現在でもこのシステムが使用されていますが、大規模で複雑なネットワークにはあまり適切ではありません。
- *NIS* - ネットワーク情報サービス (NIS) は DNS とは独立して開発され、目的はやや異なっています。DNS は数値 IP アドレスの代わりにマシン名を使うことによって、通信を簡略化することに焦点を当てているのに対して、NIS の場合は、多

様なネットワーク情報を集中管理することによりネットワーク管理機能を高めることに焦点を絞っています。NIS には、マシンの名前とアドレス、ユーザー、ネットワークそのもの、ネットワークサービスについての情報も格納されます。NIS 名前空間情報は NIS マップに格納されています。NIS アーキテクチャーと NIS 管理の詳細は、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

- NIS+ - NIS+ は、ホスト名から IP アドレスと Ethernet アドレスへのマッピング、パスワードの検査など、ネットワーク管理サービスに対する集中制御の機能を提供します。『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』 を参照してください。
- *FNS* - フェデレーテッド・ネーミング・サービス (FNS) を使うと、単独の Solaris オペレーティング環境で複数の異なるネームサービスを使うことができます。FNS を使用すれば、ネットワーク上のさまざまなネームサービスすべてに、1 つの簡単なネーミングシステムインタフェースで対応できます。FNS は、X/Open federated naming (XFN) 規格に適合しています。FNS は、NIS+、NIS、DNS、`/etc` ファイルの代わりとして使用することはできません。FNS はむしろこれらのサービスの一番上に位置しており、共通の名前をデスクトップ上のアプリケーションで使用できるようにします。『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照してください。

ディレクトリサービス

Solaris オペレーティング環境では、iPlanet Directory Server 5.x やその他の LDAP ディレクトリサーバーと関連して LDAP (Lightweight Directory Access Protocol) をサポートします。ネームサービスとディレクトリサービスの違いは、拡張機能の差です。ディレクトリサービスはネームサービスと同じ機能のほかに、追加機能を提供します。『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

ファイルサービス

NFS アプリケーション層プロトコルは、Solaris オペレーティングシステム用のファイルサービスを提供します。NFS サービスの詳細については、『*Solaris* のシステム管理 *(*資源管理とネットワークサービス*)*』を参照してください。

ネットワーク管理

SNMP (Simple Network Management Protocol) を使うと、ネットワークのレイアウトを表示し、主要マシンの状態を表示できます。また、GUI ベースのソフトウェアで複雑なネットワーク統計情報を参照できます。多くの企業が、SNMP を実装するネットワーク管理パッケージを提供しています。SunNet Manager™ はその一例です。

ルーティングプロトコル

TCP/IP ネットワーク用の 2 つのルーティングプロトコルとして、RIP (Routing Information Protocol) と RDISC (Router Discovery Protocol) があります。これらのプロトコルについては、109 ページの「ルーティングプロトコル」で説明します。

TCP/IP プロトコルがデータ通信を行う方法

ユーザーが TCP/IP アプリケーション層プロトコルを使用するコマンドを発行すると、一連のイベントが発生します。ユーザーのコマンドまたはメッセージはローカルマシン上の TCP/IP プロトコルスタックを通過します。次に、ネットワークメディアを通過して、受信側のプロトコルに到達します。送信側ホストの各層のプロトコルにより、オリジナルのデータに情報が付加されていきます。

送信側ホストの各層のプロトコルは、受信側ホストのそれぞれの対等プロトコルとの間で対話します。図 2–1に、この対話がどのように行われるかを示します。

データのカプセル化と TCP/IP プロトコルスタック

パケットは、ネットワーク上で転送される情報の基本単位です。パケットには、少なくとも、送信側ホストと受信側ホストのアドレスが入ったヘッダーと、転送するデータが入ったボディーが含まれます。パケットが TCP/IP プロトコルスタックを通過するとき、各層のプロトコルは、基本ヘッダーにフィールドを追加したり、そこからフィールドを削除したりします。送信側ホストのプロトコルがパケットヘッダーにデータを追加する場合、その動作をデータのカプセル化と呼びます。また、変更後のパケットを表す言葉は、図 2–1 に示すように層によって異なります。

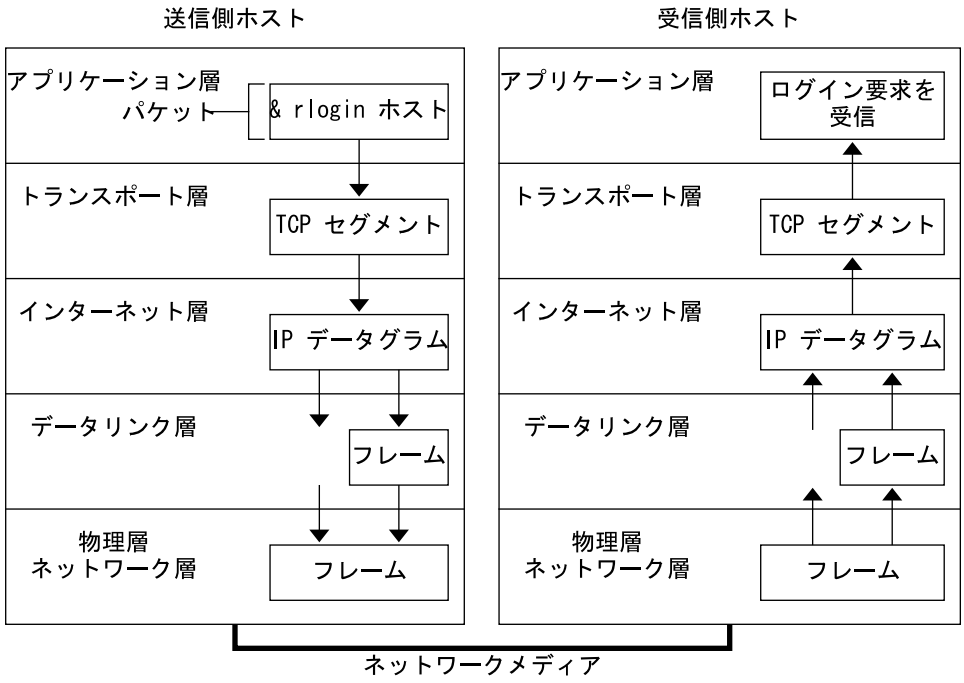


図 **2–1** パケットが TCP/IP スタックを通過する方法

この節では、パケットのライフサイクルを要約して示します。ライフサイクルは、ユーザーがコマンドを発行するか、またはメッセージを送信することによって開始します。受信側ホストの該当アプリケーションがパケットを受信するとライフサイクルは終了します。

アプリケーション層 — ユーザーが通信を開始

パケットの履歴は、あるホストのユーザーが、リモートホストへのアクセスを必要とするようなメッセージを送信するかコマンドを発行した時点から始まります。アプリケーションプロトコルは、対応する TCP か UDP のどちらかのトランスポート層プロトコルがそのパケットを取り扱えるように、パケットの形式を設定します。

図 2–1 に示したように、ユーザーが、リモートホストにログインするために `rlogin` コマンドを発行したとします。`rlogin` コマンドは TCP トランスポート層プロトコルを使用します。TCP は、コマンド内の情報を含むデータをバイトストリーム形式で受け取るものと仮定しています。したがって、`rlogin` はこのデータを TCP ストリームとして送信します。

しかし、すべてのアプリケーション層プロトコルが TCP を使用するわけではありません。あるユーザーが、リモートホストのファイルシステムをマウントしようとして、NIS+ アプリケーション層プロトコルを開始したとします。NIS+ は UDP トランスポート層プロトコルを使用します。したがって、このコマンドを含むパケットは、UDP が仮定しているような方法に形式化する必要があります。この種類のパケットをメッセージと言います。

トランスポート層 — データのカプセル化の開始

データがトランスポート層に到着すると、この層のプロトコルはデータのカプセル化を開始します。最終的な結果は、TCP と UDP のどちらが情報を処理したかによって異なります。

TCP のセグメンテーション

TCP はしばしば「接続指向型」プロトコルと呼ばれますが、これは、TCP が、受信側ホストにデータが正常に到達したかどうかを確認するからです。 図 2–1 に、TCP プロトコルが `rlogin` コマンドからのストリームをどのように受け取るかを示してあります。TCP は、アプリケーション層から受け取ったデータをセグメントに分割し、各セグメントにヘッダーを添付します。

セグメントヘッダーには、送信側と受信側のポート、セグメント順序に関する情報、検査合計と呼ばれるデータフィールドが含まれています。両方のホストの TCP プロトコルがこの検査合計データを使用して、データがエラーなしに転送されたかどうかを判別します。

TCP 接続の確立

TCP は、受信側ホストでデータ受信の準備が整っているかどうかを判別するためにも、セグメントを使用します。送信側 TCP は、接続を確立するために、受信側ホストの TCP プロトコルに SYN と呼ばれるセグメントを送信します。受信側 TCP は ACK と呼ばれるセグメントを戻して、セグメントを正しく受信したことを知らせます。送信側 TCP は新たな ACK セグメントを送信して、それからデータの送信を開始します。このような制御情報の交換を「3 相ハンドシェーク」と呼びます。

UDP パケット

UDP は「コネクションレス」プロトコルです。TCP の場合と異なり、UDP は、受信側ホストにデータが到達したかどうかを確認しません。そのかわりに、UDP は、アプリケーション層から受信したメッセージを*UDP* パケットの形式に設定します。UDP は、各パケットにヘッダーを付加します。ヘッダーには、送信側ホストと受信側ホストのポート、パケットの長さを示すフィールド、検査合計が含まれます。

送信側 UDP プロセスは、受信側ホストの対等 UDP プロセスにパケットを送ろうとします。アプリケーション層は、受信側 UDP プロセスが、パケットを受信したことを示す肯定応答を戻すかどうかを判別します。UDP は受領の通知を必要としません。UDP は 3 相ハンドシェークを使用しません。

インターネット層

図 2–1 に示したように、TCP と UDP はどちらもセグメントとパケットを下位のインターネット層に送り、セグメントとパケットはそこで IP プロトコルにより処理されます。IP は、セグメントとパケットを IP データグラムと呼ばれる単位に形式化して、配送の準備を整えます。次に、IP はデータグラムの IP アドレスを判別して、受信側ホストへの効率的な配送ができるようにします。

IP データグラム

IP は、TCP または UDP が付加した情報に付け加える形で、セグメントまたはパケットのヘッダーに「IP ヘッダー」を付加します。IP ヘッダーには、送信側ホストと受信側ホストの IP アドレス、データグラムの長さ、データグラムのシーケンス番号が含まれます。これらの情報が付加されるのは、データグラムがネットワークパケットとしての許容バイトサイズを超過してフラグメント化が必要になった場合に備えるためです。

データリンク層 — フレーミングの実施

PPP などのデータリンク層プロトコルは、IP データグラムをフレームの形に形式化します。これらのプロトコルは、第 3 のヘッダーとフッターを付加することにより、データグラムを「フレーミング」します。フレームヘッダーには、フレームがネットワークメディアを通過するときのエラーを検査するための、巡回冗長検査 (CRC) フィールドが含まれています。次に、データリンク層は物理層にフレームを渡します。

物理ネットワーク層 — フレームの転送準備

送信側ホストの物理ネットワーク層は、フレームを受け取ると、IP アドレスをネットワークメディアに合わせたハードウェアアドレスに変換します。次に、物理ネットワーク層は、フレームをネットワークメディアに送り出します。

受信側ホストでのパケットの取り扱い

受信側ホストに到着したパケットは、送信側ホストのときと逆の順序で TCP/IP プロトコルスタックを通過します。 図 2–1 にこの経路を示してあります。受信側ホストの各プロトコルは、送信側ホストの対等プロトコルがパケットに付加したヘッダー情報を取り除きます。この処理の順序を以下に示します。

1. 物理ネットワーク層はフレーム形式のパケットを受け取ります。パケットの CRC を計算し、データリンク層にフレームを送信します。
2. データリンク層はフレームの CRC が正しいかどうかを検査し、フレームヘッダーと CRC を取り除きます。最後に、データリンクプロトコルは、インターネット層にフレームを送ります。

3. インターネット層はヘッダーの情報を読み、転送の種別を識別します。そして、パケットがフラグメントであるかどうかを判別します。その転送がフラグメントである場合は、IP は、フラグメントを組み立て直して、オリジナルのデータグラムに戻します。そして、IP ヘッダーを取り除いてから、データグラムをトランスポート層プロトコルに渡します。
4. トランスポート層 (TCP と UDP) はヘッダーを読んで、どのアプリケーション層プロトコルにデータを渡すかを判断します。次に、TCP または UDP は、自分に関連するヘッダーを取り除き、メッセージまたはストリームを受信アプリケーションに送信します。
5. アプリケーション層はメッセージを受信し、送信側ホストから要求された操作を実行します。

TCP/IP 内部トレース機能

TCP/IP は、RST パケットにより接続が終了したときに、TCP 通信のログを記録することで内部トレースをサポートします。 RST パケットが送信または受信されたときに、直前に送受信された最大 10 パケットの情報が接続情報とともにログに記録されます。

TCP/IP とインターネットについてもっと詳しく知るには

TCP/IP とインターネットについては、さまざまな方法で情報を入手できます。本書で説明していない特別な情報は、以下に挙げる情報源からも入手できます。

コンピュータ関係書籍

地域の図書館やコンピュータ関係の書店に、TCP/IP とインターネットに関する多数の書籍がそろっています。その中でも次の書籍をお勧めします。

- Craig Hunt 著『*TCP/IP Network Administration*』 - 異種 TCP/IP ネットワークの管理について、ある程度の理論と、豊富な実践的情報が記載されています。
- W. Richard Stevens 著『TCP/IP Illustrated, Volume I』 - TCP/IP のプロトコルが詳細に解説されています。これは、TCP/IP に関する技術的な背景知識を必要とするネットワーク管理者およびネットワークプログラマにとって最適です。
- Ed Krol 著『*The Whole Internet User's Guide & Catalog* 』 - インターネットを介して情報を検索するためのさまざまなツールの使用に関心がある方にとって最適です。

RFC と FYI

IAB (Internet Architecture Board) は、パブリックドメインで公開する前にすべての RFC を承認する必要があります。一般に、RFC の中の情報は開発者やその他の高度の専門知識を持つ読者を対象として設計されています。

一般に、FYI (For Your Information) 文書は RFC のサブセットとして発行されます。FYI には、インターネット規格を取り扱うような内容は含まれていません。むしろ、インターネットのもっと一般的な性格に関する情報を扱うものです。たとえば、FYI 文書には、TCP/IP の入門書や資料の目録、また、あらゆるインターネット関連のソフトウェアツールを網羅した要覧、インターネットと一般的なネットワーキングに関する用語集などが含まれています。

このマニュアルでも、また Solaris システム管理者セットに含まれる他のマニュアルでも、関連の RFC が参照されています。

第 3 章

TCP/IP ネットワークの計画 (手順)

この章では、コスト効率のよい整然とした方法でネットワークを構築するために解決しておく必要のある事柄について説明します。これらの事柄を解決後、ネットワークを設定し引き続き管理するための計画を立てることができます。

この章では、以下の内容について説明します。

ネットワークの設計

ネットワークを設計する段階では、組織のニーズを満たす最適なネットワークの種類を決定する必要があります。計画段階の決定事項には、以下のネットワークハードウェアが含まれます。

- ネットワークがサポートするホストマシンの数
- 使用するネットワークメディアの種類。たとえば、Ethernet、トークンリング、FDDI など
- ネットワークトポロジ、ネットワークハードウェアのレイアウトと接続
- ネットワークがサポートするホストの種類。スタンドアロン、データレス

これらの要因に基づいて、ローカルエリアネットワークのサイズを決定できます。

注 – ネットワークハードウェアの計画方法については、本書では説明しません。ハードウェアに付属しているマニュアルを参照してください。

ネットワーク計画の作業

ハードウェアの計画後は、次に、ソフトウェアに重点を置いたネットワーク計画に着手することができます。

この計画工程では次のような手順が必要になります。

1. ネットワーク番号を入手し、必要に応じてネットワークドメインを InterNIC に登録します。
2. IP ネットワーク番号を受け取ったら、ホストに適用する IP アドレス指定スキーマを考えます。
3. ネットワーク上のすべてのマシンの IP アドレスとホスト名を含むリストを作成します。このリストを使用してネットワークデータベースを構築します。
4. ネットワークでどのネームサービスを使用するかを決定します。使用できるのは、NIS、NIS+、DNS、または、ローカルな `/etc` ディレクトリにあるネットワークデータベースのどれかです。
5. 必要に応じて、管理作業を分担するための区分を設定します。
6. ネットワークがルーターを必要とするような規模のものかどうかを判断し、必要なら、ルーターをサポートするようなネットワークトポロジを作成します。
7. 必要に応じて、サブネットを設定します。

この章では、ネットワークの計画を立てる方法について説明します。

IP アドレススキーマの設定

サポートを予定しているマシンの数によって、ネットワークを設定する方法が影響を受けます。組織によっては、1 つの階または 1 つのビルの中にある数十台のスタンドアロンマシンから成る小さいネットワークが必要な場合もあります。また、複数のビルに散在する 1000 以上のホストを持つネットワークの設定が必要な場合もあります。このような大きい配置の場合は、ネットワークをサブネットと呼ばれる小区分に分割することが必要になる場合もあります。予定されているネットワークのサイズは、次の事項に影響を与えます。

- 適用するネットワーククラス
- 受け取るネットワーク番号
- ネットワークで使用する IP アドレス指定スキーマ

ネットワーク番号の管理

所属している組織に複数のネットワーク番号が割り当てられているか、またはサブネットを使用している場合は、組織内でネットワーク番号を割り当てる総括責任者 (人または部門) を指名してください。この責任者が、割り当てられたネットワーク番号のプールを管理する権限を保持し、ネットワーク、サブネット、ホスト番号を必要に応じて割り当てます。問題の発生を避けるために、組織内に重複したネットワーク番号や無秩序なネットワーク番号が生じないことを確認してください。IPv6 への移行を計画している場合は、第 17 章を参照してください。

IPv4 アドレス指定スキーマの設計

ネットワーク番号を受け取ったら、IPv4 アドレスのホスト部をどのように割り当てるかについて、計画を立てることができます。

表 3–1 は、IPv4 アドレス空間がどのようにネットワークアドレス空間とホストアドレス空間に分かれるかを示しています。どのクラスについても、「範囲」の欄は、ネットワーク番号の最初のバイトの 10 進数値の範囲を示しています。「ネットワークアドレス」は、IPv4 アドレスの中でネットワーク部の働きをするバイト数を示します。 *xxx* は 1 バイトを表します。「ホストアドレス」は、アドレスのホスト部を表すバイト数を示します。たとえばクラス A ネットワークアドレスの場合は、最初の 1 バイトがネットワーク番号で、残りの 3 バイトがホスト番号です。クラス C ネットワークの場合はこの関係が逆になり、最初の 3 バイトがネットワーク番号で、残りの 1 バイトがホスト番号です。

表 3–1 IPv4 アドレス空間の区分

「クラス」	範囲	ネットワークアドレス	ホストアドレス
A	0 ～ 127	*xxx*	*xxx.xxx.xxx*
B	128 ～ 191	*xxx.xxx*	*xxx.xxx*
C	192 ～ 223	*xxx.xxx.xxx*	*xxx*

IPv4 アドレスの最初のバイトの数値は、ネットワークがクラス A、B、C のどれであるかを示す値で、InterNIC が割り当てます。残りの 3 つのバイトの値の範囲は、どれも 0～255 です。番号 0 と 255 は予約されています。ネットワーク管理者は、割り当てられているネットワーク番号に応じて、各バイトに 1～254 の範囲内の番号を指定することができます。

以下の表は、IPv4 アドレスのどのバイトがインターネットから割り当てられているかを示します。また、ホストへの割り当てが可能な、各バイト内の値の範囲を示します。

表 **3–2** 使用できる番号の範囲

ネットワーククラス	バイト **1** の範囲	バイト **2** の範囲	バイト **3** の範囲	バイト **4** の範囲
A	0 ～ 127	1 ～ 254	1 ～ 254	1 ～ 254
B	128 ～ 191	インターネットにより事前割り当て	1 ～ 254	1 ～ 254
C	192 ～ 223	インターネットにより事前割り当て	インターネットにより事前割り当て	1 ～ 254

ネットワークインタフェースへの IP アドレスの適用法

ネットワークに接続するには、コンピュータは少なくとも 1 つはネットワークインタフェースを持っている必要があります。各ネットワークインタフェースは、それぞれ一意な IP アドレスを持っていなければなりません。管理者がホストに与えた IP アドレスはそのホストのネットワークインタフェースに割り当てられます。このインタフェースは、一次ネットワークインタフェースと呼ばれることがあります。あるマシンに第 2 のネットワークインタフェースを追加した場合は、そのマシンにも一意な IP アドレスが必要です。第 2 のネットワークインタフェースを追加すると、そのマシンはルーターに変わります。これについては、70 ページの「ルーターの構成」を参照してください。ホストに第 2 のネットワークインタフェースを追加し、しかもルーティング機能を無効にした場合は、そのホストはマルチホームホストとみなされます。

`/devices` ディレクトリには、各ネットワークインタフェースのデバイス名、デバイスドライバ、関連のデバイスファイルが入っています。ネットワークインタフェースのデバイス名には、たとえば `le0` または `smc0` などがあります。これらは、よく使用される 2 つの Ethernet インタフェースのデバイス名です。

注 – 本書では、Ethernet ネットワークインタフェースを持つマシンを想定して説明を進めます。別のネットワークメディアを使用する予定の場合は、そのネットワークインタフェースのマニュアルの中の構成に関する情報を参照してください。

ネットワーク上のエンティティへの名前付け

割り当てられたネットワーク番号を受け取り、ホストの IP アドレスを指定したら、次のタスクはホストへの名前付けです。ここで、ネットワーク上のネームサービスをどのように扱うかを決める必要があります。これらの名称は最初にネットワークを設定する場合や、後日ルーターや PPP を使ってネットワークを拡張する場合に使います。

TCP/IP は、ネットワーク上の特定のマシンを見つけるときに、そのマシンの IP アドレスを使用します。しかし、認識しやすい名前を付ければ、人間はマシンを識別しやすくなります。したがって、TCP/IP プロトコル (および Solaris オペレーティングシステム) では、マシンを一意なものとして識別するために、IP アドレスとホスト名の両方が必要です。

TCP/IP の視点から見れば、ネットワークは名前が付けられたエンティティの集合です。ホストは名前が付けられた 1 個のエンティティです。ルーターも名前が付けられた 1 個のエンティティです。さらに、ネットワークも名前が付けられた 1 個のエンティティです。ネットワークがインストールされているグループや部門にも、名前を付けることができます。部課、地区、会社も同様です。理論的には、ネットワークを識別するために使用できる名前の階層については、事実上まったく制限はありません。この名前でドメインが特定されます。

ホスト名の管理

多くのサイトでは、各ユーザーがそれぞれのマシンの名前を選定しています。サーバーにも少なくとも 1 つのホスト名が必要で、このホスト名は一次ネットワークインタフェースの IP アドレスに関連付けられます。

ネットワーク管理者は、自己の管轄ドメイン内のすべてのホスト名が一意なものであることを確認する必要があります。たとえば、ネットワーク内に “fred” という名前を持つマシンが 2 つあってはなりません。ただし、ネットワーク内の “fred” というマシンが複数の IP アドレスを持つことはできます。

ネットワークの計画を立てるときは、IP アドレスとそれぞれのホスト名のリストを作って、設定工程中に各マシンに簡単にアクセスできるようにしてください。このリストは、すべてのホスト名が一意かどうかを検査するために役立ちます。

ネームサービスの選択

Solaris オペレーティングシステムでは、4 種類のネームサービスのどれでも任意に選択して使用できるようになっています。4 つのネームサービスとは、ローカルファイル、NIS、NIS+、DNS です。ネームサービスは、ネットワーク上のマシンに関する重要な情報、たとえばホスト名、IP アドレス、Ethernet アドレスなどを保持しています。Solaris オペレーティング環境では、LDAP ディレクトリサービスを使うオプションも提供されています。

ネットワークデータベース

オペレーティングシステムをインストールするときに、その手順の一環として、サーバーシステム、クライアントシステム、スタンドアロンシステムのホスト名と IP アドレスを入力します。Solaris インストールプログラムは、`hosts` と `ipnodes` という 2 つのネットワークデータベースにこの情報を格納します。これらのデータベースは、ネットワーク上の TCP/IP の動作に必要な情報を格納しているネットワークデータベースセットの一部です。管理者が自己のネットワーク用として選択したネームサービスは、これらのデータベースを読み取ります。

ネットワークデータベースの設定は重要です。したがって、ネットワーク計画工程の一環として、どのネームサービスを使用するかを決定する必要があります。ネームサービスの使用の決定は、ネットワークを管理ドメインとして編成するかどうかにも影響を与えます。ネットワークデータベースのセットについては、99 ページの「ネットワークデータベースと `nsswitch.conf` ファイル」に詳しい説明があります。

ネームサービスに NIS、NIS+、DNS を使用する

NIS、NIS+、DNS ネームサービスは、ネットワーク内のいくつかのサーバー上にネットワークデータベースを維持します。これらのネームサービスについては、『*Solaris* のシステム管理*(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』と『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』で説明しています。これらのマニュアルでは、データベースの設定方法、「名前空間」と「管理ドメイン」の概念についても詳しく説明しています。

ネームサービスにローカルファイルを使用する

NIS、NIS+、DNS のどれも実装しない場合は、ネットワークはローカルファイルを使用してネームサービスの機能を提供します。「ローカルファイル」とは、ネットワークデータベースが使用するためのものとして `/etc` ディレクトリに入っている一連のファイルのことです。本書に示す手順では、特に断らない限り、ネームサービスとしてローカルファイルを使用しているものとします。

注 – ネットワーク用のネームサービスとしてローカルファイルを使用することに決めた場合、後日別のネームサービスを設定することもできます。

ドメイン名

多くのネットワークでは、ホストとルーターが管理ドメインの階層の形で編成されます。NIS、NIS+、DNS のどれかのネームサービスを使用する場合は、所属組織のドメイン名として、全世界の中で一意な名前を選択する必要があります。ドメイン名が一意であることを確認するには、そのドメイン名を InterNIC に登録する必要があります。DNS を使う予定がある場合は、必ず選択したドメイン名を登録します。

ドメイン名は階層構造になっています。一般に、新規のドメインは、既存の関連ドメインの下に配置されます。たとえば、子会社のドメイン名はその親会社のドメイン名の下に配置されます。特に他との関連性のない組織のドメイン名は、既存の最上位ドメインのいずれかの下に直接配置できます。

以下に、最上位ドメインの例を示します。

- `.com` – 民間企業 (世界規模)
- `.edu` – 教育機関 (世界規模)
- `.gov` – アメリカ政府機関
- `.fr` – フランス

組織を識別する名前は、一意なものであるという条件を満たしていれば、ネットワーク管理者が任意に選択できます。

管理作業の分化

管理作業の分化の目的は、サイズと制御に関する事項を解決することにあります。ネットワーク内のホストとサーバーの数が増えるに従って、管理作業はますます複雑になります。このような状況に対処するための方法としては、管理部門を増設することが考えられます。そのためには、特定のクラスのネットワークを増設したり、既存のネットワークをサブネットに分割したりします。ネットワーク管理の作業を分化するかどうかは、以下の要因によって判断します。

- ネットワークの規模

  数百台のホストから構成される単一のネットワークは、すべてのホストが物理的に同じ場所にありしかも同じ管理サービスを必要とする場合は、1 つの管理部門で対処できます。しかし、場合によっては、複数の管理部門を設立する必要があります。サブネットを持つ小規模ネットワークが地理的に広い範囲に散在している場合は、複数の管理部門を設立する方法が効率的です。

- ネットワーク上のユーザーのニーズが共通しているかどうか

  たとえば、1 つのビル内だけに限定され比較的少数のマシンをサポートするネットワークがあるとします。これらのマシンはいくつかのサブネットワークに分割されています。各サブネットワークは、異なるニーズを持つユーザーのグループをサ

ポートします。このような場合は、サブネットごとに管理部門を設立するとよいでしょう。

ネットワークの登録

Solaris ネットワーク上のマシンに IP アドレスを割り当てるには、その前に InterNIC からネットワーク番号を入手する必要があります。さらに、管理ドメインを使っている場合は、管理ドメインを InterNIC に登録することも必要です。

InterNIC と InterNIC Registration Services

InterNIC は、以下のインターネット情報を提供するための本部組織として、1993 年に創立されました。

- インターネットの運営方針
- インターネットへのアクセス方法。これには研修サービスも含まれる
- インターネットのユーザーが利用できる資源。たとえば、匿名 FTP サーバー、Usenet ユーザーグループなど

InterNIC には、ユーザーが TCP/IP ネットワークを登録する InterNIC Registration Services という組織も含まれています。InterNIC Registration Services は、ネットワークを入手しドメインを登録するためのテンプレートを提供しています。登録する場合は、以下の点に注意してください。

- ネットワーク番号は InterNIC が割り当てる

注 – ネットワークを他の既存の TCP/IP ネットワークに接続してなくても、勝手なネットワーク番号を割り当てることはしないでください。

  InterNIC はサブネット番号を割り当てません。サブネット番号は、割り当てられたネットワーク番号と、ネットワーク管理者が指定する番号を組み合わせたものとなります。これについては、96 ページの「サブネット化とは」で説明します。
- ドメイン名は、InterNIC ではなくネットワーク管理者が決めて、それを InterNIC に登録する

InterNIC への連絡方法

InterNIC Registration Services には、次の方法で連絡できます。

- 郵便

宛先は次のとおりです。

```
Network Solutions
Attn: InterNIC Registration Services
505 Huntmar Park Drive
Herndon, Virginia 22070
```

- 電話

  電話番号は、米国の 703-742-4777 です。電話サービスの利用可能時間は、午前 7 時から午後 7 時 (米国東部標準時) までです。米国内からかける場合、フリーダイヤルの電話番号は、800–779–1710 です。

ルーターの追加

TCP/IP から見た場合、ネットワーク上に存在するのは、2 つの種類のエンティティ、つまりホストとルーターだけです。ホストはすべてのネットワークに必要ですが、ルーターはすべてのネットワークに必要なわけではありません。ネットワークの物理的なトポロジによってルーターを使用する必要があるかどうかが決まります。この節では、ネットワークトポロジとルーティングの概念を紹介します。この概念は、既存のネットワークに別のネットワークを追加しようとするときに、重要な意味を持ちます。

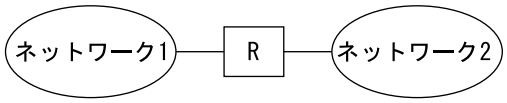
ネットワークトポロジ

ネットワークトポロジは、複数のネットワークの相互関係を示します。ルーターは、ネットワークを相互に接続するエンティティです。TCP/IP の視点から見れば、ルーターは複数のネットワークインタフェースを持つ任意のマシンです。しかし、マシンをルーターとして機能させるためには、70 ページの「ルーターの構成」の説明に従って、そのルーターを正しく構成しておく必要があります。

ルーターは、複数のネットワークを接続することで、より大きなインターネットワークを作ります。ルーターは、隣接する 2 つのネットワーク間でパケットの受け渡しをするように構成する必要があります。さらに、隣接するネットワークを越えた位置にあるネットワークに、パケットを渡す機能も備えられている必要があります。

図 3–1 に、ネットワークトポロジの基本部分を示します。最初の図は、2 つのネットワークを 1 台のルーターで接続した単純な構成です。2 番目の図は、3 つのネットワークを 2 台のルーターで相互接続した構成を示しています。最初の例では、ルーター R がネットワーク 1 とネットワーク 2 を連結して、より大きなインターネットワークを作っています。2 番目の例では、ルーター R1 がネットワーク 1 と ネットワーク 2 を接続し、ルーター R2 がネットワーク 2 とネットワーク 3 を接続して、ネットワーク 1、2、3 からなる 1 つのネットワークが作られています。

1 つのルーターによって接続されている 2 つのネットワーク



2 つのルーターによって接続されている 3 つのネットワーク

図 **3–1** 基本的なネットワークトポロジ

ルーターは、ネットワークを連結してインターネットワークを作ります。また、ルーターは、宛先ネットワークのアドレスに基づいて、ネットワーク相互間でパケットをルーティングします。インターネットワークがより複雑になるにつれて、ルーターがパケットの宛先を決定する回数は増加します。

次の図は、複雑さが増加する例を示します。ルーター R3 は、ネットワーク 1 とネットワーク 3 を直接接続します。この冗長性により信頼性が向上します。ネットワーク 2 がダウンしても、ルーター R3 はネットワーク 1 と 3 の間のルートを提供することができます。ネットワークは複数個を相互接続することができます。ただし、相互接続するネットワークは、同じネットワークプロトコルを使う必要があります。

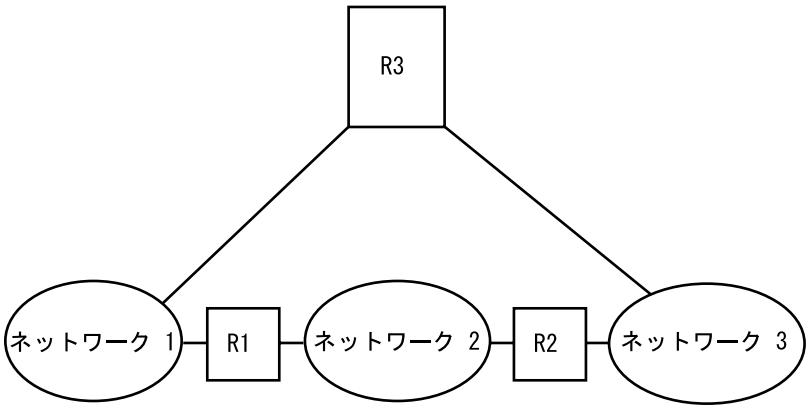


図 **3–2** ネットワーク間のパスの追加

ルーターがどのようにパケットを転送するか

パケットヘッダーに含まれている受信側 IP アドレスによってパケットをルーティングする方法が決まります。このアドレスにローカルネットワークのネットワーク番号が含まれている場合は、その IP アドレスを持つホストに直接パケットが送られます。ネットワーク番号がローカルネットワークではない場合は、パケットはローカルネットワーク上のルーターに送られます。

ルーターは、ルーティングテーブル内にルーティング情報を維持します。このテーブルには、ルーターが接続されているネットワーク上のホストとルーターの IP アドレスが含まれています。また、それらのネットワークを指すポインタも含まれています。ルーターは、パケットを受信すると、ルーティングテーブルを調べて、ヘッダー内の宛先アドレスがテーブルにリストされているかどうかを確認します。テーブルにその宛先アドレスが含まれていない場合は、ルーターは、ルーティングテーブルにリストされている他のルーターにパケットを転送します。ルーターについての詳細は、70 ページの「ルーターの構成」を参照してください。

次の図は、2 つのルーターにより接続された 3 つのネットワークのネットワークトポロジを示します。

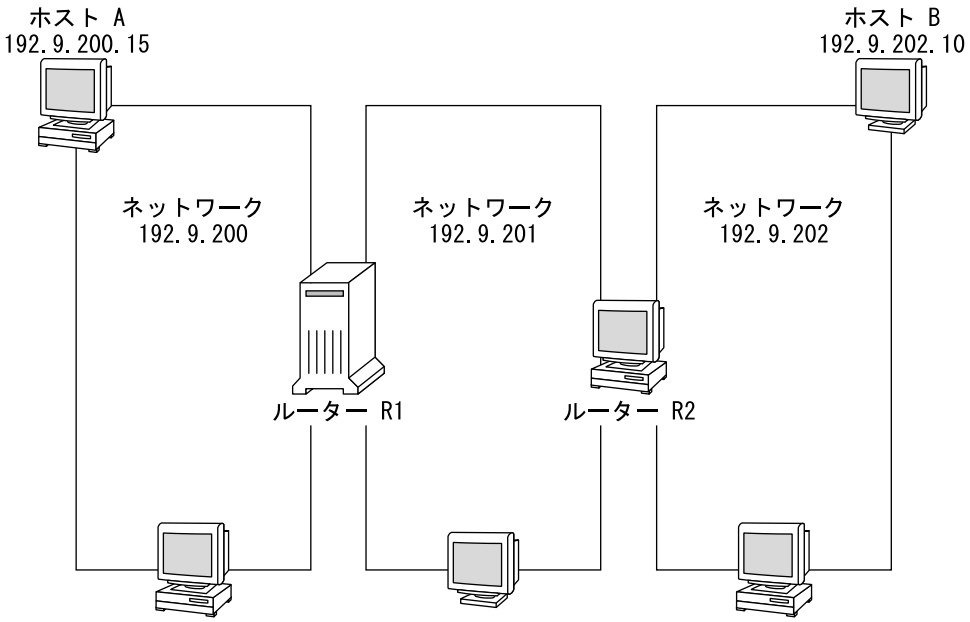


図 **3–3** 相互接続された 3 つのネットワーク

ルーター R1 は、ネットワーク 192.9.200 とネットワーク 192.9.201 と接続しています。ルーター R2 は、ネットワーク 192.9.201 とネットワーク 192.9.202 と接続しています。ネットワーク 192.9.200 のホスト A がネットワーク 192.9.202 のホスト B にメッセージを送る場合、以下のイベントが発生します。

1. ホスト A は、ネットワーク 192.9.200 にパケットを送り出します。パケットヘッダーには、受信側ホスト B の IPv4 アドレスである 192.9.202.10 が含まれています。
2. ネットワーク 192.9.200 には、192.9.202.10 の IPv4 アドレスを持つマシンはありません。したがって、ルーター R1 がパケットを受け取ります。
3. ルーター R1 は自己のルーティングテーブルを調べます。ネットワーク 192.9.201 には、アドレスが 192.9.202.10 であるマシンはありません。ただし、ルーティングテーブルにはルーター R2 がリストされています。
4. R1 は「次のホップ」ルーターとして R2 を選択し、パケットを R2 に送信します。
5. R2 はネットワーク 192.9.201 を 192.9.202 に接続しているので、ホスト B に関するルーティング情報を保持しています。そこで、ルーター R2 はパケットをネットワーク 192.9.202 に転送し、ホスト B がそのパケットを受信します。

第 4 章

TCP/IP の管理 (手順)

TCP/IP の管理には、ネットワークを設定するための手順が含まれます。まず、ハードウェアを組み立てます。次に TCP/IP を構成します。この章では、TCP/IP を構成する方法について説明します。また、TCP/IP に関する障害追跡方法について説明します。

この章では、以下の内容について説明します。

TCP/IP を構成する前に

TCP/IP を構成する前に、次の表に示す作業を完了する必要があります。

表 **4–1** TCP/IP を構成する前に (作業マップ)

説明	参照先
ネットワーク設計者の場合は、ネットワークトポロジを設計する	53 ページの「ネットワークトポロジ」
インターネットのアドレス指定機関からネットワーク番号を入手する	47 ページの「IPv4 アドレス指定スキーマの設計」
ネットワークトポロジに従ってネットワークハードウェアを組み立てる。ハードウェアが正しく動作することを確認する	ハードウェアのマニュアルと 53 ページの「ネットワークトポロジ」
ネットワークインタフェースとルーターが必要とする構成ソフトウェアがあれば、それを実行する	53 ページの「ルーターの追加」と 70 ページの「ルーターの構成」
ネットワークに対する IP アドレス指定スキーマの計画を立てる。これには、必要に応じてサブネットアドレス指定も含まれる	47 ページの「IPv4 アドレス指定スキーマの設計」と 288 ページの「IPv6 アドレス指定」
ネットワーク内のすべてのマシンに、IP 番号とホスト名を割り当てる	47 ページの「IPv4 アドレス指定スキーマの設計」と 288 ページの「IPv6 アドレス指定」
ネットワークでどのネームサービス、つまり NIS、NIS+、DNS、またはローカルファイルのどれを使用するかを決定する	『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : *DNS*、*NIS*、*LDAP* 編)』と『 *Solaris* のシステム管理 (ネーミングとディレクトリサービス : *FNS*、*NIS*+ 編) 』
必要なら、ネットワークで使用するドメイン名を選択する	『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : *DNS*、*NIS*、*LDAP* 編)』と『 *Solaris* のシステム管理 (ネーミングとディレクトリサービス : *FNS*、*NIS*+ 編) 』
ネットワーク上の少なくとも 1 台のマシンにオペレーティングシステムをインストールする	『*Solaris 9* インストールガイド』

ホスト構成モードの決定

ネットワーク管理者は、ホストとルーター (必要な場合) で実行できるように TCP/IP を構成します。これらのマシンは、ローカルマシン上のファイルまたはネットワーク上の他のマシンにあるファイルから構成情報を入手するように構成できます。必要な構成情報を以下に示します。

- マシンのホスト名
- マシンの IP アドレス
- マシンが所属するドメイン名
- デフォルトルーター
- マシンのネットワークで使用しているネットマスク (適用可能な場合)

TCP/IP 構成情報をローカルファイルから入手するマシンは、ローカルファイルモードで稼動します。TCP/IP 構成情報をリモートマシンから入手するマシンは、ネットワーククライアントモードで稼動します。

ローカルファイルモードで実行するマシン

ローカルファイルモードで実行するマシンは、TCP/IP 構成ファイルをローカルに持っている必要があります。これらのファイルについては、89 ページの「TCP/IP 構成ファイル」で説明します。このマシンが専用のディスクを持っていることが望ましいですが、不可欠というわけではありません。

ほとんどのサーバーはローカルファイルモードで実行します。このようなサーバーの一部を以下に示します。

- ネットワーク構成サーバー
- NFS サーバー
- NIS、NIS+、または DNS のサービスを提供するネームサーバー
- メールサーバー

また、ルーターはローカルファイルモードで実行する必要があります。

印刷サービス専用として機能するマシンは、ローカルファイルモードで実行する必要はありません。個々のホストをローカルファイルモードで実行する方がよいかどうかは、ネットワークの規模によって異なります。

ネットワークがきわめて小さい場合は、個々のホストのファイルを管理する作業は比較的簡単です。しかし、数百のホストから成るネットワークの場合は、そのネットワークがいくつかの管理サブドメインに分割されていたとしても、この作業は困難なものとなります。したがって、規模の大きいネットワークの場合は、ローカルファイルモードを使用しても一般に効率は上がりません。ただし、ルーターとサーバーはそれぞれ自身で構成されるものなので、ローカルファイルモードで構成する必要があります。

ネットワーク構成サーバー

ネットワーク構成サーバーは、ネットワーククライアントモードで構成されているホストに、TCP/IP 構成情報を提供するマシンです。この種のサーバーは、次の 3 つのブートプロトコルをサポートしています。

- RARP – 逆アドレス解決プロトコル (RARP) は、Ethernet アドレス (48 ビット) を IPv4 アドレス (32 ビット) にマッピングします。つまり、ARP と逆のことを行ないます。ネットワーク構成サーバーで RARP を実行すると、ネットワーククライアントモードで動作しているホストは、各自の IP アドレスと TCP/IP 構成ファイルをそのネットワーク構成サーバーから入手します。RARP サービスは、`in.rarpd` デーモンを使用して使用可能にできます。詳細については、`in.rarpd(1M)` のマニュアルページを参照してください。
- TFTP – 簡易ファイル転送プロトコル (TFTP) は、リモートマシン間でファイルを転送するアプリケーションです。`in.tftpd` デーモンが TFTP サービスを実施し、その結果、ネットワーク構成サーバーとそれぞれのネットワーククライアントとの間のファイル転送が可能になります。詳細については、`in.tftpd(1M)` のマニュアルページを参照してください。
- bootparams – bootparams プロトコルは、ネットワークブートを行うクライアントが必要とする、ブート用パラメータを供給します。このサービスを実行するのは `rpc.bootparamd` デーモンです。詳細については、`bootparamd(1M)` のマニュアルページを参照してください。

ネットワーク構成サーバーは、NFS ファイルサーバーとしても使用できます。

ホストのどれかをネットワーククライアントとして構成する場合は、ネットワーク内のマシンの少なくとも 1 つをネットワーク構成サーバーとして構成する必要があります。ネットワークをサブネット化する場合は、ネットワーククライアントを持つ各サブネットについて、ネットワーク構成サーバーが少なくとも 1 つは必要です。

ネットワーククライアントであるマシン

ネットワーク構成サーバーから自己の構成情報を入手するホストは、ネットワーククライアントモードで稼動します。ネットワーククライアントとして構成したマシンでは、TCP/IP 構成ファイルのローカルコピーは不要です。

ネットワーククライアントモードを使用すると、大規模ネットワークの管理が大幅に簡素化されます。個々のホストで行う構成作業が最小限の量で済み、ネットワーク上のすべてのマシンが同じ構成標準に従っていることが保証されます。

すべての種類のコンピュータでネットワーククライアントモードを構成できます。たとえば、スタンドアロンシステムやデータレスマシンだけでネットワーククライアントモードを構成できます。

混合構成

すべてをローカルホストモードに構成したり、すべてをネットワーククライアントモードに構成する以外の構成方法も可能です。ルーターとサーバーは常にローカルモードで構成する必要があります。ホストについては、ローカルモードとネットワーククライアントモードを任意に組み合わせて使用できます。

サンプルネットワーク

次の図は、ネットワーク番号が 192.9.200 である架空のネットワークのホストを示しています。このネットワークにはネットワーク構成サーバーが 1 つあり、それは `sahara` というマシンです。`tenere` と `nubian` の 2 つのマシンはそれぞれ独自にディスクを持っており、ローカルファイルモードで動作します。マシン `faiyum` もディスクを持っていますが、これはネットワーククライアントモードで動作します。

最後に、マシン `timbuktu` はルーターとして構成されています。このマシンには 2 つのネットワークインタフェースが組み込まれています。そのうちの 1 つは `timbuktu` という名前です。このネットワークインタフェースはネットワーク 192.9.200 に属しています。残りの 1 つは `timbuktu-201` という名前です。このネットワークインタフェースはネットワーク 192.9.201 に属しています。どちらのネットワークも、組織ドメイン `deserts.worldwide.com` に含まれています。このドメインは、ローカルファイルをネームサービスとして使用します。

この章で示されている例のほとんどは、次の図で示されるネットワークを使っています。

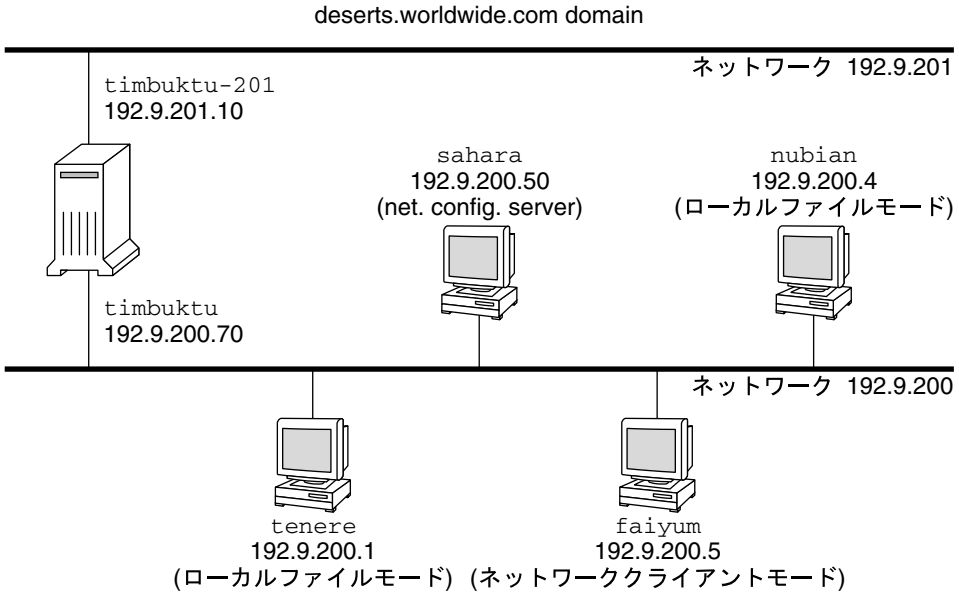


図 **4–1** サンプルネットワーク内のホスト

ネットワークにサブネットを追加する (作業マップ)

サブネットを使っていないネットワークを、サブネットを使うネットワークに変更する場合、次の表に示す作業を実行します。

表 **4–2** ネットワークにサブネットを追加する (作業マップ)

説明	参照先
1. 新しいサブネットトポロジについて決定する。これには、ルーターに関する考慮事項や、サブネット上でのホストの位置などが含まれる	53 ページの「ルーターの追加」、96 ページの「サブネット化とは」、および 111 ページの「ネットワーククラス」
2. すべてのサブネットアドレスとホストアドレスを割り当てる	46 ページの「IP アドレススキーマの設定」、および 110 ページの「IPv4 アドレスの構成部分」

表 **4–2** ネットワークにサブネットを追加する (作業マップ)　*(続き)*

説明	参照先
3. 手動で TCP/IP を構成している場合は、`/etc/inet/netmasks` ファイルを修正する。そうでない場合は Solaris インストールプログラムを使用してネットマスクを修正する	95 ページの「`netmasks` データベース」、および 96 ページの「IPv4 アドレス用のネットワークマスクの作成」
4. 新しいホストアドレスを反映させるために、すべてのホスト上で `/etc/inet/hosts` ファイルおよび `/etc/inet/ipnodes` ファイルを変更する	92 ページの「`hosts` データベース」、および 95 ページの「`ipnodes` データベース」
5. すべてのマシンをリブートする	

ネットワーク構成手順

オペレーティングシステムのソフトウェアをインストールするときに、同時にネットワークのソフトウェアもインストールされます。そのときに、いくつかの IP 構成パラメータを対応するファイルに格納して、ブート時に読み取れるようにしておく必要があります。

ここで必要な手順は、ネットワーク構成ファイルを作成または編集するということです。構成情報をマシンのカーネルに提供する方法は、状況に応じて決めます。これらのファイルがローカルに格納されているか (ローカルファイルモード) ネットワーク構成サーバーから入手するか (ネットワーククライアントモード) によって提供方法が変わります。

ネットワーク構成時に指定するパラメータを以下に示します。

- すべてのマシンの各ネットワークインタフェースの IP アドレス
- ネットワーク上の各マシンのホスト名。ホスト名は、ローカルファイルまたはネームサービスデータベースに入力できる
- マシンが設置されている、NIS、NIS+、または DNS のドメイン名 (該当する場合)
- デフォルトのルーターアドレス。この情報は、各ネットワークにルーターが 1 つしか接続していないような単純なネットワークトポロジの場合、または、ルーターが RDISC (Router Discovery Protocol) や RIP (Routing Information Protocol) などのルーティングプロトコルを実行しない場合に指定する。これらのプロトコルについての詳細は、109 ページの「ルーティングプロトコル」を参照
- サブネットマスク (サブネットを持つネットワークの場合に限り必要)

ここでは、ローカル構成ファイルを作成および編集する手順を説明しています。ネームサービスデータベースの処理については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』と『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照してください。

ネットワークを構成する (作業マップ)

表 **4–3** ネットワークを構成する (作業マップ)

タスク	説明	参照先
ホストをローカルファイルモード用に構成する	`nodename`、`hostname`、`hosts`、`defaultdomain`、`defaultrouter`、および `netmasks` ファイルを編集する	64 ページの「ローカルファイルモードの場合のホストの構成方法」
ネットワーク構成サーバーをセットアップする	`in.tftp` デーモンをオンにし、`inetd.conf`、`hosts`、`ethers`、`bootparams` ファイルを編集する	65 ページの「ネットワーク構成サーバーの設定方法」
ホストをネットワーククライアントモード用に構成する	`hostname` ファイルを作成し、`hosts` ファイルを編集する。また、`nodename` ファイルと `defaultdomain` ファイルがある場合はこれらを削除する	67 ページの「ネットワーククライアントモードの場合のホストの構成方法」
ネットワーククライアントに対してルーターを指定する	`defaultrouter` ファイルと `hosts` ファイルを編集する	67 ページの「ネットワーククライアント用のルーターの指定方法」

▼ ローカルファイルモードの場合のホストの構成方法

ローカルファイルモードで動作するマシン上の TCP/IP を構成するための手順は、次のとおりです。

1. スーパーユーザーになり、**`/etc`** ディレクトリに移動します。
2. マシンのホスト名を **`/etc/nodename`** ファイルに入力します。

   たとえば、ホストの名前が `tenere` であるとすれば、このファイルに `tenere` と入力します。
3. 各ネットワークインタフェースについて、**`/etc/hostname.`***interface* という名前のファイルを作成します。

   一次ネットワークインタフェースについては、Solaris インストールプログラムが自動的にこのファイルを作成します。詳細については、90 ページの「`/etc/hostname.`*interface* ファイル」を参照してください。IPv6 を使用している場合は、326 ページの「IPv6 ネットワークインタフェース構成ファイル」を参照してください。
4. **`/etc/hostname.`***interface* ファイルに、インタフェース **IP** アドレスまたはインタフェース名を入力します。

   たとえば、`hostname.ie1` という名前のファイルを作成し、ホストのインタフェースの IP アドレスまたはホスト名を入力します。

5. **/etc/inet/hosts** ファイルを編集して以下の内容を追加します。

   a. ローカルマシンに増設したネットワークインタフェースに割り当てた **IP** アドレスと、各インタフェースのホスト名

      一次ネットワークインタフェースとループバックアドレスについてのエントリは、すでに Solaris インストールプログラムにより作成されています。

   b. **/usr** ファイルシステムを **NFS** マウントする場合は、ファイルサーバーの **IP** アドレス

注 – Solaris インストールプログラムは、ローカルマシン用のデフォルトの /etc/inet/host を作成します。このファイルが存在していない場合は、92 ページの「hosts データベース」の説明に従って作成してください。また、IPv6 を使用している場合は、339 ページの「/etc/inet/ipnodes ファイル」を参照してください。

---

6. ホストの完全指定ドメイン名を **/etc/defaultdomain** ファイルに入力します。

   たとえば、ホスト tenere がドメイン deserts.worldwide.com に所属しているとします。その場合は、/etc/defaultdomain に deserts.worldwide.com を入力します。詳細は、 91 ページの「/etc/defaultdomain ファイル」を参照してください。

7. ルーターの名前を **/etc/defaultrouter** に入力します。

   詳細は、92 ページの「/etc/defaultrouter ファイル」を参照してください。

8. デフォルトのルーターの名前とその **IP** アドレスを **/etc/inet/hosts** に入力します。

   上記以外にも、使用できるルーティングオプションがいくつかあります。 67 ページの「ネットワーククライアントモードの場合のホストの構成方法」の、ルーティングオプションについての説明を参照してください。これらのオプションは、ローカルファイルモード構成にも適用できます。

9. ネットワークをサブネット化する場合は、ネットワーク番号とネットマスクを **/etc/inet/netmasks** ファイルに入力します。

   NIS または NIS+ サーバーを設定してある場合は、サーバーとクライアントが同じネットワーク上にあれば、サーバー上の該当のデータベースにネットマスク情報を入力できます。

10. ネットワーク上の各マシンをリブートします。

▼ ネットワーク構成サーバーの設定方法

1. スーパーユーザーになり、予定しているネットワーク構成サーバーのルートディレクトリに移動します。

2. ディレクトリ **/tftpboot** を作成することにより、**in.tftpd** デーモンが動作するようにします。

   ```
   # mkdir /tftpboot
   ```

   このコマンドにより、マシンは、TFTP、bootparams、RARP のサーバーに構成されます。

3. 手順 **2** で作成したディレクトリに対するシンボリックリンクを作成します。

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

4. **inetd.conf** ファイルにある **tftp** の行を有効にします。

   /etc/inetd.conf のエントリが次のようになっていることを確認してください。

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

   これによって、/tftpboot に格納されたファイル以外のファイルを inettftpd() で検索できなくなります。

5. **hosts** データベースを編集して、ネットワーク上のすべてのクライアントのホスト名と **IP** アドレスを追加します。

6. **ethers** データベースを編集して、ネットワーククライアントモードで実行するネットワーク上のすべてのホストについてエントリを作成します。

7. **bootparams** データベースを編集します。

   104 ページの「bootparams データベース」を参照してください。ワイルドカードエントリを作成するか、または、ネットワーククライアントモードで実行するすべてのホストについてエントリを作成します。

8. コマンド行から次のコマンドを入力します。

   ```
   # pkill -HUP inetd
   ```

インストールサーバー、ブートサーバーを設定する方法については、『*Solaris 9* インストールガイド』を参照してください。

ネットワーククライアントの構成

ネットワーククライアントは、各自の構成情報をネットワーク構成サーバーから入手します。したがって、あるホストをネットワーククライアントとして構成するときは、このネットワーク用として、ネットワーク構成サーバーが少なくとも 1 つは設定されていることを確認してください。

▼ ネットワーククライアントモードの場合のホストの構成方法

ネットワーククライアントモードで構成する必要のある各ホストについて、次のことを行います。

1. スーパーユーザーになります。
2. ディレクトリを調べて、**/etc/nodename** ファイルがあるかどうかを確認します。ある場合は、このファイルを削除してください。

   /etc/nodename を削除すると、システムは hostconfig プログラムを使用して、ネットワーク構成サーバーから、ホスト名、ドメイン名、ルーターアドレスを入手するようになります。63 ページの「ネットワーク構成手順」を参照してください。
3. **/etc/hostname.***interface* ファイルが存在していない場合は、それを作成します。

   そのファイルが空であることを確認してください。/etc/hostname.*interface* ファイルが空であれば、システムはネットワーク構成サーバーから IP アドレスを入手します。IPv6 を使用している場合は、326 ページの「IPv6 ネットワークインタフェース構成ファイル」を参照してください。
4. **/etc/inet/hosts** ファイルに、ループバックネットワークインタフェースのホスト名と **IP** アドレス以外の内容が入っていないことを確認します。

   詳細については、93 ページの「ループバックアドレス」を参照してください。このファイルには、ローカルマシン (一次ネットワークインタフェース) の IP アドレスとホスト名が入っていてはいけません。IPv6 を使用している場合は、339 ページの「/etc/inet/ipnodes ファイル」を参照してください。
5. **/etc/defaultdomain** ファイルがあるかどうかを調べます。ある場合は、このファイルを削除してください。

   hostconfig プログラムは、自動的にドメイン名を設定します。hostconfig プログラムが設定したドメイン名を上書きしたい (無効にしたい) ときは、/etc/defaultdomain に代わりのドメイン名を入力します。
6. クライアントの **/etc/nsswitch.conf** の中の検索パスが、ネットワークのネームサービスの要件を満たしていることを確認します。

▼ ネットワーククライアント用のルーターの指定方法

1. ネットワーク上にルーターが **1** つしかなく、ネットワーク構成サーバーが自動的にそのルーターの名前を指定するようにしたい場合は、ネットワーククライアントが **/etc/defaultrouter** ファイルを持っていないことを確認します。

2. 次の手順に従って、ネットワーク構成サーバーが設定したデフォルトのルーターの名前を上書き (無効に) します。

   a. ネットワーククライアント上に **/etc/defaultrouter** を作成します。

   b. デフォルトのルーターとして指定してあるマシンのホスト名と **IP** アドレスを入力します。

   c. 指定したデフォルトのルーターのホスト名と **IP** アドレスを、ネットワーククライアントの **/etc/inet/hosts** に追加します。

3. ネットワークに複数のルーターがある場合は、ネットワーククライアント上に **/etc/defaultrouter** を作成し、空のままにしておきます。

/etc/defaultrouter を作成し、それを空のままにしておくと、2 つの動的ルーティングプロトコル、つまり、ICMP RDISC (Router Discovery Protocol) か RIP (Routing Information Protocol) のどちらか一方が実行されます。システムは、まず in.rdisc プログラムを実行します。このプログラムは、ルーター検出プロトコルを実行しているルーターを捜します。該当するルーターが見つかった場合は、in.rdisc はそのまま実行を続け、RDISC プロトコルを実行するルーターを監視します。

RDISC プロトコルに応答しているルーターがないと判断した場合は、システムは RIP を使用して in.routed デーモンを実行し、ルーターを監視します。

標準 TCP/IP サービスの構成

telnet、ftp、rlogin などのサービスは、inetd デーモンによって開始されます。このデーモンは、ブート時に自動的に実行されます。inetd(1M) と inetd.conf(4) のマニュアルページを参照してください。

/etc/inetd.conf ファイル内のサービス定義のほかに、/etc/default/inetd ファイルを使って inetd を構成できます。たとえば、すべての着信接続をログに記録するように構成できます。また、アクセス制御するための TCP ラッパー機能を使用するように構成できます。

▼ すべての着信 TCP 接続の IP アドレスを記録する方法

1. スーパーユーザーになります。

2. **/etc/default/inetd** を編集し、次の行を追加して、ログ記録をオンにします。

```
ENABLE_CONNECTION_LOGGING=YES
```

注 – この行がコメント記号付きでファイルに存在している場合は、コメント記号を削除するだけでログ記録をオンにできます。

3. **inetd** デーモンを終了します。
4. **inetd** デーモンを再起動します。

ネームサービスの詳細については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』と『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照してください。

▼ TCP ラッパーを使って TCP サービスのアクセスを制御する方法

1. スーパーユーザーになります。
2. **/etc/default/inetd** を編集し、次の行を追加して、**TCP** ラッパーをオンにします。

```
ENABLE_TCPWRAPPERS=YES
```

注 – この行がコメント記号付きでファイルに存在している場合は、コメント記号を削除するだけでログ記録をオンにできます。

3. **inetd** デーモンを終了します。
4. **inetd** デーモンを再起動します。
5. **hosts_access(3)** のマニュアルページ **(**パスは **/usr/sfw/man)** の説明に従って **TCP** ラッパーアクセス制御を構成します。

ルーターの構成

TCP/IP がルーターに求める第 1 の必要条件は、少なくとも 2 つのネットワークインタフェースが取り付けられていることです。ネットワークインタフェースのどれか 1 つが使用可能な状態にあれば、ルーターは自動的に RDISC プロトコルと RIP プロトコルで「情報交換」します。これらのプロトコルは、ネットワーク上でのルーターの状態を監視し、ネットワーク上のホストにルーターを通知します。

ルーターを物理的にネットワークにインストール後、64 ページの「ローカルファイルモードの場合のホストの構成方法」 の説明に従って、ルーターをローカルファイルモードで動作するように構成します。これで、ネットワーク構成サーバーがダウンしても、ルーターが確実にブートされるようになります。ホストと違って、ルーターには構成を要するインタフェースが 少なくとも 2 つあるということを忘れないでください。

ルーターを構成する (作業マップ)

表 **4–4** ルーターを構成する (作業マップ)

タスク	説明	参照先
マシンをルーターとして構成する	`hostname` および `hosts` ファイルを作成し、アドレスを追加する	71 ページの「マシンをルーターとして構成する方法」
ネットワーククライアントであるホスト上で静的ルーティングを選択する	`defaultrouter` ファイルにエントリを追加する	71 ページの「ネットワーククライアントであるホスト上で静的ルーティングを選択する方法」
ネットワーククライアントであるホスト上で動的ルーティングを選択する	`defaultrouter` ファイルのエントリを編集する	72 ページの「ネットワーククライアントであるホスト上で動的ルーティングを選択する方法」
マシンを強制的にルーターにする	`gateways` ファイルを作成する	72 ページの「マシンを強制的にルーターにする方法」

ルーターの両方のネットワークインタフェースの構成

ルーターは、複数のネットワーク間のインタフェースを提供するものなので、ルーターの各ネットワークインタフェースカードに、それぞれ一意な名前と IP アドレスを割り当てる必要があります。これで、各ルーターは、その一次ネットワークインタフェースのホスト名と IP アドレスに加えて、増設した各ネットワークインタフェースについて少なくとも 1 つずつ、一意な名前と IP アドレスを持つことになります。

▼ マシンをルーターとして構成する方法

1. ルーターとして構成するマシン上でスーパーユーザーになります。
2. マシンにインストールされている追加の各ネットワークインタフェースについて、**/etc/hostname.***interface* ファイルを作成します。

   例えば、hostname.ie0 と hostname.ie1 を作成します。詳細は、90 ページの「/etc/hostname.*interface* ファイル」を参照してください。IPv6 を使用している場合は、326 ページの「IPv6 ネットワークインタフェース構成ファイル」を参照してください。
3. 各ファイルに、そのインタフェースに対して選択したホスト名を入力します。

   たとえば、hostname.ie0 ファイルに timbuktu という名前を入力し、hostname.ie1 ファイルに timbuktu-201 という名前を入力します。どちらのインタフェースも同じマシンに置かれることになります。
4. 各インタフェースのホスト名と **IP** アドレスを **/etc/inet/hosts** に入力します。

   例 :

   ```
   192.9.200.20     timbuktu       #interface for network 192.9.200
   192.9.201.20     timbuktu-201   #interface for network 192.9.201
   192.9.200.9      gobi
   192.9.200.10     mojave
   192.9.200.110    saltlake
   192.9.200.12     chilean
   ```

   インタフェース timbuktu と timbuktu-201 は、同じマシンにあります。timbuktu-201 のネットワークアドレスが、 timbuktu とは異なる点に注意してください。これは、ネットワーク 192.9.201 のメディアが timbuktu-201 ネットワークインタフェースに接続されるのに対し、ネットワーク 192.9.200 のメディアは timbuktu インタフェースに接続されるためです。IPv6 を使用している場合は、339 ページの「/etc/inet/ipnodes ファイル」を参照してください。
5. サブネット化したネットワークにルーターを接続する場合は、**/etc/inet/netmasks** を編集して、ローカルネットワーク番号 **(**たとえば **129.9.0.0)** と、関連のネットマスク番号 **(**たとえば **255.255.255.0)** を入力します。

起動スクリプトは、マシン上でルーティングプロトコル (RIP または RDISC) を起動するか、静的ルーティングを使用するかを決定します。

▼ ネットワーククライアントであるホスト上で静的ルーティングを選択する方法

1. ホスト上でスーパーユーザーになります。
2. ネットワーク上のルーターのエントリを **/etc/defaultrouter** ファイルに追加します。

92 ページの「/etc/defaultrouter ファイル」を参照してください。唯一の静的なデフォルトルートがルーティングテーブルに組み込まれます。この条件下では、ホストは動的ルーティングプロトコル (RIP や RDISC など) を実行しません。

▼ ネットワーククライアントであるホスト上で動的ルーティングを選択する方法

1. ホスト上でスーパーユーザーになります。
2. **/etc/defaultrouter** ファイルが空であることを確認します。
   このファイルが空である場合、ネットワーククライアントは必ず動的ルーティングプロトコルを選択します。

使用される動的ルーティングのタイプは以下の判定条件に従って選択されます。

- /usr/sbin/in.rdisc プログラムが存在する場合は、起動スクリプトは in.rdisc を起動する。すると、ネットワーク上で RDISC を実行しているすべてのルーターが、ホストからのすべての RDISC 照会に応答するようになる。少なくとも 1 つのルーターが応答すれば、ホストはルーティングプロトコルとして RDISC を選択する。
- ネットワークルーターが RDISC を実行していない場合、または RDISC 照会に対する応答が失敗した場合は、ホストでの in.rdisc は終了する。ホストは in.routed を起動し、その結果 RIP が実行される。

▼ マシンを強制的にルーターにする方法

/etc/hostname.*interface* ファイルを 1 つだけ持つマシン (デフォルトではホスト) を、強制的にルーターにすることができます。

1. マシン上でスーパーユーザーになります。
2. 名前が **/etc/gateways** というファイルを作成し、空のままにしておきます。

この手順は、PPP リンクを構成することに決めた場合は特に重要です。詳細は、『*Solaris* のシステム管理 (資源管理とネットワークサービス)』を参照してください。

マルチホームホストの作成

デフォルトでは、TCP/IP は、複数のネットワークインタフェースを持つマシンをすべてルーターとみなします。しかし、ルーターをマルチホームホストに変更することもできます。マルチホームホストとは、複数のネットワークインタフェースを持っているけれども、ルーティングプロトコルの実行も IP パケットの転送もしないマシンのことです。一般に、次のような種類のマシンはマルチホームホストとして構成します。

- NFS サーバー、特に大規模なデータセンターは、複数のネットワークに接続することによって、多数のユーザー間でファイルを共有できるようになります。この種のサーバーはルーティングテーブルを備えている必要はありません。
- データベースサーバーは、NFS サーバーの場合と同じ目的で複数のネットワークインタフェースを持つことにより、多数のユーザーに資源を提供できます。
- ファイアウォールゲートウェイは、企業のネットワークとインターネットなどの公共ネットワークとの間の接続を提供するマシンです。管理者は、セキュリティの手段としてファイアウォールを設定します。ファイアウォールとして構成されたホストは、自己に接続されているネットワーク相互間でのパケットの受け渡しを行いません。しかし、許可されたユーザーに対しては、通常どおり `ftp` や `rlogin` などの標準 TCP/IP サービスを提供します。

TCP/IP は、複数のネットワークインタフェースを持つマシンのすべてをルーターとみなすので、そのマシンをマルチホームホストに変えるには、いくつかの操作が必要になります。

▼ マルチホームホストの作成方法

1. マルチホームホストにしたいマシン上でスーパーユーザーになります。
2. マシンにインストールされている追加の各ネットワークインタフェースについて、**`/etc/hostname.`***interface* ファイルを **1** つずつ作成します。
3. 次のように入力します。

   % **`touch /etc/notrouter`**

   このコマンドで、`/etc/notrouter` という名前の、空のファイルが作成されます。
4. マシンをリブートします。

マシンをリブートすると、起動スクリプトは /etc/notrouter ファイルの有無を確認します。このファイルが存在する場合は、起動スクリプトは、in.routed -s も in.rdisc -r も実行しません。 また、ifconfig により “up” として構成されているすべてのインタフェースで IP 転送がオフに設定されます。この処理は、/etc/gateways ファイルが存在しているかどうかに関係なく行われます。これで、マシンはマルチホームホストになります。

省スペースモードをオンにする

省スペースモードでは、デフォルトのルートだけを含むテーブルがホストに提供されます。デフォルトでは、省スペースモードをオフにした状態で、ホストで in.routed が実行されます。

フルルーティングテーブル (これは、構成に誤りのあるルーターを排除するための保護を強化します) をホストが持つ必要がない場合は、省スペースモードをオンにします。

▼ 省スペースモードをオンにする方法

1. ホスト上でスーパーユーザーになります。
2. **/etc/rc2.d/S69inet** 起動スクリプトを編集し、**/usr/sbin/in.routed -q** の行に、**-S** オプションを追加します。

   ```
   /usr/sbin/in.routed -q
   ```

   上記の行を次のように変更します。

   ```
   /usr/sbin/in.routed -q -S
   ```

ICMP ルーター検索をオフにする

ルーターの信頼性などの理由で、ホストに RDISC (ルーター検索プロトコル) を使用させたくない場合があります。ホストにおいて、RDISC ではなく RIP の自動選択が確実に動作する場合は、ネットワーク内のルーター (特に RDISC を実行するもの) も確実に動作しなければなりません。

RDISC を実行するルーターが他にないときに、Solaris ルーターを 1 つインストールすると、デフォルトの状態で、そのルーターに接続されるすべてのホストがそのルーターだけに依存することになります。そのネットワーク上のホストが他のルーターも使用できるようにするには、新しいルーターで RDISC をオフにします。

ICMP ルーター検索をオフにする (作業マップ)

表 **4–5** ICMP ルーター検索をオフにする (作業マップ)

タスク	説明	参照先
ホスト上で ICMP ルーター検索をオフにする	ホストの `in.rdisc` ファイルの名前を変更する	95 ページの「`netmasks` データベース」
ルーター上で ICMP ルーター検索をオフにする	ルーターの `in.rdisc` ファイルの名前を変更する	96 ページの「サブネット化とは」

▼ ホスト上で ICMP ルーター検索をオフにする方法

1. ホスト上でスーパーユーザーになります。
2. ホストの **`/usr/sbin/in.rdisc`** ファイルの名前を **`/usr/sbin/in.rdisc.saved`** などに変更します。
3. ホストをリブートします。

▼ ルーター上で ICMP ルーター検索をオフにする方法

1. ルーター上でスーパーユーザーになります。
2. ルーターの **`/usr/bin/in.rdisc`** ファイルの名前を他の名前に変更します。
3. ルーターをリブートします。

一般的な障害追跡方法

ネットワーク上での問題を示す最初の兆候は、1 つまたはいくつかのホストでの通信の消滅です。あるホストを初めてネットワークに追加したときに、そのホストがまったく動作しない場合は、構成ファイルのどれかに問題があることが考えられます。また、ネットワークインタフェースカードに問題がある可能性もあります。1 つのホストに突然問題が生じた場合は、ネットワークインタフェースに原因があると考えられます。ネットワーク上のホスト相互間の通信はできるが、他のネットワークとの通信ができないという場合は、ルーターに問題があるか、または他のネットワークに問題があることが考えられます。

ifconfig プログラムを使用すればネットワークインタフェースに関する情報を入手でき、netstat を使用すればルーティングテーブルとプロトコル統計を表示できます。サードパーティのネットワーク診断プログラムから、さまざまな障害追跡ユーティリティが提供されています。詳細は、サードパーティのマニュアルを参照してください。

比較的明らかになりにくいのは、ネットワーク上での性能低下の原因です。たとえば、ping のようなツールを使用することで、ホストでのパケットの消失など、問題の原因を突き止めることができます。

ソフトウェア検査の実行

ネットワークに障害が生じた場合は、以下のような処置のいずれかによって、ソフトウェア関連の問題を診断し修正することができます。

- netstat コマンドを使用してネットワーク情報を表示します。
- hosts データベース (IPv6 を使用している場合は ipnodes データベースも) を検査して、個々のエントリが適正で最新であるかどうかを確認します。
- RARP を実行している場合は、 ethers データベース内の Ethernet アドレスを検査して、個々のエントリが適正で最新であるかどうかを確認します。
- telnet によりローカルホストに接続してみます。
- ネットワークデーモン inetd が実行中であることを確認します。スーパーユーザーとしてログインし、次のように入力します。

  # **ps -ef | grep inetd**

inetd デーモンが実行中であれば、次の例に示すような出力が表示されます。

```
root 57 1 0 Apr 04 ? 3:19 /usr/sbin/inetd -s
root 4218 4198 0 17:57:23 pts/3 0:00 grep inetd
```

ping コマンド

ping コマンドは、特定のホストとの IP 接続が存在しているかどうかを確認するために使用します。基本構文は次のとおりです。

/usr/sbin/ping *host [timeout]*

この構文で、*host* は問題のマシンのホスト名を示します。オプションの *timeout* 引数は、ping がそのマシンに到達しようと試みる秒数を示し、デフォルトは 20 秒です。詳しい構文とオプションについては、ping(1M) のマニュアルページを参照してください。

ping を実行すると、ICMP プロトコルは、指定されたホストにデータグラムを送って、応答を求めます。ICMP は、TCP/IP ネットワーク上のエラー処理を担当するプロトコルです。詳細については、35 ページの「ICMP プロトコル」を参照してください。

ping コマンド (作業マップ)

表 **4–6** ping コマンド (作業マップ)

タスク	説明	参照先
ホストが動作しているか確認する	ホスト名に対して ping を実行する	99 ページの「ネットワークデータベースと nsswitch.conf ファイル」
ホストでパケットが失われていないか確認する	ping コマンドの -s オプションを使用する	99 ページの「ネットワークデータベースへのネームサービスの影響」

▼ ホストが動作しているか確認する方法

- コマンド行で次のコマンドを入力します。

  % **ping** *hostname*

  ホスト *hostname* が動作していれば、次のメッセージが表示されます。

  hostname is alive

  このメッセージは、*hostname* が ICMP の要求に応答したことを示します。*hostname* がダウン状態にあるかまたは ICMP パケットを受け取れなかった場合は、ping から次の応答が返されます。

  no answer from *hostname*

▼ ホストでパケットが失われていないか確認する方法

マシンが動作状態にあるのにパケットが失われている疑いがある場合は、ping に -s オプションを指定することにより、問題を追求できます。

- コマンド行で次のコマンドを入力します。

  % **ping -s** *hostname*

ping は、ユーザーが割り込み文字を送るかタイムアウトが発生するまで、*hostname* にパケットを送り続けます。画面上には次のように出力されます。

```
PING elvis: 56 data bytes
64 bytes from 129.144.50.21: icmp_seq=0. time=80. ms
64 bytes from 129.144.50.21: icmp_seq=1. time=0. ms
```

```
64 bytes from 129.144.50.21: icmp_seq=2. time=0. ms
64 bytes from 129.144.50.21: icmp_seq=3. time=0. ms
.
.
.
----elvis PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms) min/avg/max = 0/20/80
```

パケットロスの統計値は、ホストがパケットを失ったかどうかを示します。

ping が失敗した場合は、ifconfig と netstat が報告するネットワーク状態を調べます。これについては、78 ページの「ifconfig コマンド」と 79 ページの「netstat コマンド」を参照してくだい。

ifconfig コマンド

ifconfig コマンドは、指定したインタフェースの構成に関する情報を表示します。詳細は ifconfig(1M) のマニュアルページを参照してください。ifconfig の構文は次のとおりです。

ifconfig *interface-name [protocol_family]*

ifconfig コマンド (作業マップ)

表 **4–7** ifconfig コマンド (作業マップ)

タスク	説明	参照先
特定のインタフェースに関する情報を入手する	ifconfig コマンドを使用する	78 ページの「特定のインタフェースに関する情報を入手する方法」
ネットワーク上のすべてのインタフェースに関する情報を入手する	ifconfig コマンドの -a オプションを使用する	101 ページの「nsswitch.conf ファイル — 使用するネームサービスの指定」

▼ 特定のインタフェースに関する情報を入手する方法

1. スーパーユーザーになります。
2. コマンド行で次のコマンドを入力します。

   # **ifconfig** *interface*

le0 インタフェースの場合、出力は次のようになります。

```
le0: flags=863<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
     inet 129.144.44.140 netmask ffffff00 broadcast 129.144.44.255
ether 8:0:20:8:el:fd
```

上記の flags セクションは、インタフェースが “up” として構成されていて、ブロードキャストの能力があり、“trailer” リンクレベルのカプセル化を使用していないことを示しています。mtu フィールドは、このインタフェースの最大転送サイズが 1500 オクテットであることを示しています。2 行目には、使用しているホストの IP アドレス、現在使用されているネットマスク、インタフェースの IP ブロードキャストアドレスの情報が含まれています。3 行目は、ホストのマシンアドレス (この場合は Ethernet) です。

▼ ネットワーク上のすべてのインタフェースに関する情報を入手する方法

ifconfig の便利なオプションの 1 つに -a オプションがあります。これを使用すると、ネットワーク上のすべてのインタフェースに関する情報が提供されます。

1. スーパーユーザーになります。
2. コマンド行で次のコマンドを入力します。

   # **ifconfig -a** *interface*

このコマンドにより、たとえば次のようなメッセージが表示されます。

```
le0:  flags=49<UP,LOOPBACK,RUNNING> mtu 8232
     inet 127.144.44.140 netmask ff000000
le0:flags=863<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
     inet 129.144.44.140 netmask ffffff00 broadcast 129.144.44.255
ether 8:0:20:8:el:fd
```

動作していないインタフェースがあることが出力に示されている場合は、そのインタフェースに問題があると考えられます。その場合は、ifconfig(1M) のマニュアルページを参照してください。

netstat コマンド

netstat コマンドは、ネットワーク状態とプロトコル統計を表示します。TCP と UDP のエンドポイントの状態 (テーブル形式)、ルーティングテーブルの情報、インタフェースの情報を表示できます。

netstat は、選択したコマンド行オプションに応じて、さまざまな種類のネットワークデータを表示します。この表示は、特にシステム管理に役立ちます。このコマンドの構文は次のとおりです。

netstat [-m] [-n] [-s] [-i | -r] [-f *address_family*]

ネットワーク状態の判別のために最もよく使用されるオプションは、s、r、i です。オプションの説明については、netstat(1M) のマニュアルページを参照してください。

netstat コマンド (作業マップ)

表 **4–8** netstat コマンド (作業マップ)

タスク	説明	参照先
プロトコル別に統計情報を表示する	netstat コマンドの -s オプションを使用する	80 ページの「プロトコル別の統計情報の表示方法」
ネットワークインタフェースの状態を表示する	netstat コマンドの -i オプションを使用する	81 ページの「ネットワークインタフェースの状態の表示方法」
ルーティングテーブルの状態を表示する	netstat コマンドの -r オプションを使用する	82 ページの「ルーティングテーブルの状態の表示方法」

▼ プロトコル別の統計情報の表示方法

netstat の -s オプションは、UDP、TCP、ICMP、および IP のプロトコルについて、プロトコル別の統計情報を表示します。

- コマンド行で次のコマンドを入力します。

```
% netstat -s
```

結果は、次に示す出力例のように表示されます (出力の一部は省略してあります)。この情報には、プロトコルに問題のある箇所が示されることがあります。たとえば ICMP からの統計情報は、このプロトコルがどこにエラーを検出したかを示します。

```
UDP

      udpInDatagrams      =  39228     udpOutDatagrams     =  2455
      udpInErrors         =      0

TCP

      tcpRtoAlgorithm     =      4     tcpMaxConn          =    -1
      tcpRtoMax           =  60000     tcpPassiveOpens     =     2
      tcpActiveOpens      =      4     tcpEstabResets      =     1
      tcpAttemptFails     =      3     tcpOutSegs          =   315
```

```
            .
            .
IP

      ipForwarding        =     2      ipDefaultTTL        =   255
      ipInReceives        =  4518      ipInHdrErrors       =     0
            .
            .
ICMP

      icmpInMsgs          =     0      icmpInErrors        =     0
      icmpInCksumErrs     =     0      icmpInUnknowns      =     0
            .
            .

IGMP:

0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
```

▼ ネットワークインタフェースの状態の表示方法

netstat の -i オプションは、このコマンドを実行したマシンで構成されているネットワークインタフェースの状態を表示します。

- コマンド行で次のコマンドを入力します。

```
% netstat -i
```

次に示すのは、netstat -i による出力結果の例です。

```
Name Mtu  Net/Dest     Address   Ipkts    Ierrs Opkts    Oerrs  Collis  Queue
le0  1500 b5-spd-2f-cm tatra     14093893 8492  10174659 1119   2314178   0
lo0  8232 loopback     localhost 92997622 5442  12451748 0      775125    0
```

この表示から、各ネットワークでマシンが送受信するパケットの数がわかります。たとえば、サーバーについて表示される入力パケットカウント (Ipkts) はクライアントがブートを試みるたびに増加しているのに、出力パケットカウント (Opkts) が変化しないことがあります。この結果は、サーバーがクライアントからのブート要求パケットを見ているが、それを応答すべきものとして認識していないことを示しています。この混乱は、hosts データベース、ipnodes データベース、または ethers データベース内に誤ったアドレスがあることが原因であると考えられます。

しかし、入力パケットカウントが長時間にわたり変化しない場合は、マシンがパケットをまったく見ていません。この場合は、上記と違って、ハードウェアの問題の可能性が高くなります。

▼ ルーティングテーブルの状態の表示方法

netstat の -r オプションは、IP ルーティングテーブルを表示します。

- コマンド行で次のコマンドを入力します。

```
% netstat -r
```

次に示すのは、マシン tenere で実行した netstat -r の出力結果の例です。

```
Routing tables
Destination   Gateway Flags Refcnt Use   Interface
temp8milptp   elvis   UGH   0      0
irmcpeb1-ptp0 elvis   UGH   0      0
route93-ptp0  speed   UGH   0      0
mtvb9-ptp0    speed   UGH   0      0
                .
mtnside       speed   UG    1      567
ray-net       speed   UG    0      0
mtnside-eng   speed   UG    0      36
mtnside-eng   speed   UG    0      558
mtnside-eng   tenere  U     33     190248  le0
```

最初の列は宛先ネットワーク、2 番目の列はパケットを転送するルーターを示しています。U フラグはルートが up 状態であること、G フラグはルートがゲートウェイへのものであることを示します。H フラグは、宛先がネットワークではなく、完全指定のホストアドレスであることを示します。

Refcnt 列は 1 ルート当たりの有効ユーザーの数、Use 列は 1 ルート当たりの送信パケット数を示します。 最後の Interface 列は、ルートで使用されているネットワークインタフェースを示します。

ネットワークの問題の記録

ルーティングデーモンについて誤動作の疑いがある場合は、routed デーモンを起動するときのすべてのパケット転送も含む、ルーティングデーモンの動作をログに記録することができます。

▼ ネットワークの問題を記録する方法

1. スーパーユーザーになります。

2. コマンド行で次のコマンドを入力することにより、ルーティングデーモンの動作を記録するログファイルを作成します。

   # **/usr/sbin/in.routed /var/***logfilename*

注意 – ビジー状態のネットワークでは、このコマンドによりほとんど絶え間なく出力が生じることがあります。

パケットの内容表示

snoop を使用すると、ネットワークパケットを取得して内容を表示できます。取得したパケットについては、そのまま表示することも、ファイルに保存することも可能です。snoop が中間ファイルに書き込む場合、トレースのビジー状態でパケットロスはほとんど発生しません。その後、snoop 自体はファイルの解釈に使用されます。snoop の使用方法については、snoop(1M) のマニュアルページを参照してください。

snoop コマンドは必ずスーパーユーザー になって実行してください。プロミスキュアス (promiscuous) モードでデフォルトのインタフェースとやりとりするパケットを取得できます。最上位のプロトコルに関連するデータのみが一覧形式で表示されます。たとえば NFS パケットでは、NFS 情報のみが表示されます。RPC、UDP、IP、および Ethernet のフレーム情報は抑止されますが、verbose (詳細表示) オプションのいずれかを選択してあれば表示できます。

snoop が取得するファイルの形式は、RFC 1761 で説明しています。

snoop server client rpc rstatd は、クライアント/サーバー間のすべての RPC トラフィックを収集し、rstatd に対するフィルタをかけます。

パケットの内容を表示する (作業マップ)

表 **4–9** パケットの内容を表示する (作業マップ)

タスク	説明	参照先
システムからすべてのパケットをチェックする	netstat コマンドと snoop コマンドを使用し、その結果を解析する	84 ページの「システムから全パケットを確認する方法」
snoop の結果をファイルに取り込む	snoop コマンドの -o オプションを使用する	84 ページの「snoop の結果をファイルに取り込む方法」

表 **4–9** パケットの内容を表示する (作業マップ)　(続き)

タスク	説明	参照先
サーバーとクライアントの間のパケットをチェックする	snoop コマンドの結果をファイルに保存し、その結果を解析する	85 ページの「サーバー/クライアント間のパケットを確認する方法」

▼ システムから全パケットを確認する方法

1. スーパーユーザーになります。
2. コマンド行で次のコマンドを入力し、システムに接続されているインタフェースを見つけます。

   # **netstat -i**

   通常、snoop では最初の非ループバックデバイス (le0) が使用されます。
3. **snoop** と入力します。

   Ctrl-C キーを押してプロセスを停止します。

```
# snoop
Using device /dev/le (promiscuous mode)
     maupiti -> atlantic-82  NFS C GETATTR FH=0343
 atlantic-82 -> maupiti      NFS R GETATTR OK
     maupiti -> atlantic-82  NFS C GETATTR FH=D360
 atlantic-82 -> maupiti      NFS R GETATTR OK
     maupiti -> atlantic-82  NFS C GETATTR FH=1A18
 atlantic-82 -> maupiti      NFS R GETATTR OK
     maupiti -> (broadcast)  ARP C Who is 120.146.82.36, npmpk17a-82 ?
```

4. 結果を解釈します。

   上記の例では、クライアント maupiti からサーバー atlantic-82 への転送には NFS ファイルハンドル 0343 が使用され、atlantic-82 は OK と応答しています。「who is 120.146.82.36?」と問い合わせる ARP 要求が maupiti から伝送されるまで、会話は継続します。

   この例は、snoop の形式を説明しています。次の手順では、snoop にフィルタをかけてファイルにパケットを取り込みます。

   取り込んだファイルを解釈するには、RFC 1761 に記述された説明を参照してください。

▼ snoop の結果をファイルに取り込む方法

1. スーパーユーザーになります。
2. コマンド行で次のコマンドを入力します。

   # **snoop -o** *filename*

   例 :

```
# snoop -o /tmp/cap
Using device /dev/le (promiscuous mode)
30 snoop: 30 packets captured
```

これによって、ファイル /tmp/cap に 30 個のパケットが取り込まれました。ディスク容量が十分にあれば、ファイルはどこにでも格納できます。取り込んだパケットの数はコマンド行に表示され、Ctrl-C を押せばいつでも終了できます。

snoop 自体によってホストマシン上にネットワーク負荷がかかるので、結果に誤差が生じる場合があります。正確な結果を確認するには、第 3 のシステム (クライアントまたはサーバーに接続されているハブのいずれかを外したシステム) から snoop を実行してください (次の節を参照)。

3. コマンド行で次のコマンドを入力し、ファイルを検査します。

```
# snoop -i filename
```

例 :

```
# snoop -i /tmp/cap

1  0.00000 frmpk17b-082 -> 224.0.0.2    IP  D=224.0.0.2 S=129.146.82.1 LEN=32, ID=0
2  0.56104        scout -> (broadcast)  ARP C Who is 129.146.82.63, grail ?
3  0.16742  atlantic-82 -> (broadcast)  ARP C Who is 129.146.82.76, honeybea ?
4  0.77247        scout -> (broadcast)  ARP C Who is 129.146.82.63, grail ?
5  0.80532 frmpk17b-082 -> (broadcast)  ARP C Who is 129.146.82.92, holmes ?
6  0.13462        scout -> (broadcast)  ARP C Who is 129.146.82.63, grail ?
7  0.94003        scout -> (broadcast)  ARP C Who is 129.146.82.63, grail ?
8  0.93992        scout -> (broadcast)  ARP C Who is 129.146.82.63, grail ?
9  0.60887        towel -> (broadcast)  ARP C Who is 129.146.82.35, udmpk17b-82 ?
10 0.86691  nimpk17a-82 -> 129.146.82.255 RIP R (1 destinations)
```

ARP、IP、RIP その他の詳細な分析と推奨されるパラメータについては、特定のプロトコルのマニュアルを参照してください。RFC は Web で参照できます。

▼ サーバー/クライアント間のパケットを確認する方法

1. **snoop** を実行するシステムから、クライアントまたはサーバーのいずれかに接続されたハブを外します。

   この第 3 のシステム (snoop システム) はすべてのトラフィックを監視するので、snoop のトレースには実際のネットワーク上の状態が反映されます。

2. スーパーユーザーになります。

3. コマンド行で **snoop** にオプションを指定して実行し、結果をファイルに保存します。

4. 結果の検査と解釈を行います。

   snoop 取り込みファイルの詳細については、RFC 1761 を参照してください。

頻繁かつ定期的に snoop を使用して、システムが正常に動作している場合の状態を把握してください。最近の白書や RFC を参照したり、NFS や YP といった特定分野の専門家からアドバイスを受けたりするのも、パケットの分析に役立ちます。snoop とそのオプションの使用法については、snoop(1M) のマニュアルページを参照してください。

ルーティング情報の表示

traceroute ユーティリティは、IP パケットが特定のインターネットホストに至るまでのルートを追跡する際に使用します。traceroute ユーティリティは、IP プロトコルの ttl (time to live) フィールドを利用して、経路に沿った各ゲートウェイからの ICMP TIME_EXCEEDED 応答の受信を試みます。また、宛先ホストからの PORT_UNREACHABLE (または、ECHO_REPLY) の受信も試みます。traceroute ユーティリティは、ttl を 1 にして探査の送信を開始し、目的のホストが見つかるか、最大数の中間ホストを通過するまで、ttl を 1 ずつ増加します。

traceroute ユーティリティは、ルーティングの誤設定やルーティング経路の障害を判定する場合に特に役立ちます。特定のホストが到達不可能な場合には、traceroute ユーティリティ を使用して、パケットがどの経路をたどって目的のホストに到達し、どこで障害が起きる可能性があるかを調べることができます。

また、traceroute ユーティリティは、経路に沿った各ゲートウェイの宛先ホストとの間の往復時間も表示します。この情報は、2 つのホスト間のどこでトラフィックが遅くなっているかを分析する際に利用することができます。

▼ traceroute ユーティリティの実行方法

- コマンド行で次のコマンドを入力します。

  ```
  % traceroute destination-hostname
  ```

traceroute ユーティリティの詳細については、traceroute(1M) のマニュアルページを参照してください。

例: traceroute ユーティリティ

以下の traceroute コマンドの例では、パケットがホスト istanbul から ホスト sanfrancisco までにたどる 7 つの経路と、パケットが各経路を通過する時間が表示されています。

```
istanbul% traceroute sanfrancisco
    traceroute: Warning: Multiple interfaces found; using 172.31.86.247 @ le0
    traceroute to sanfrancisco (172.29.64.39), 30 hops max, 40 byte packets
```

```
1  frbldg7c-86 (172.31.86.1)  1.516 ms  1.283 ms  1.362 ms
2  bldg1a-001 (172.31.1.211)  2.277 ms  1.773 ms  2.186 ms
3  bldg4-bldg1 (172.30.4.42)  1.978 ms  1.986 ms  13.996 ms
4  bldg6-bldg4 (172.30.4.49)  2.655 ms  3.042 ms  2.344 ms
5  ferbldg11a-001 (172.29.1.236)  2.636 ms  3.432 ms  3.830 ms
6  frbldg12b-153 (172.29.153.72)  3.452 ms  3.146 ms  2.962 ms
7  sanfrancisco (172.29.64.39)  3.430 ms  3.312 ms  3.451 ms
```

第 5 章

TCP/IP (リファレンス)

この章では、TCP/IP 構成ファイルの種類、目的、ファイルエントリのフォーマットなどについて説明する、TCP/IP ネットワークの参照情報を提供します。また、既存のネットワークデータベースについても詳しく説明します。

さらにこの章では、定義されているネットワーククラスとサブネット番号に基づいて、IPv4 アドレスが構成される仕組みについても説明します。

TCP の詳細については、`tcp(7P)` のマニュアルページを参照してください。

この章では、以下の内容について説明します。

TCP/IP 構成ファイル

ネットワーク上の各マシンは、以下に示す TCP/IP 構成ファイルとネットワークデータベースから自己の TCP/IP 構成情報を入手します。

- `/etc/hostname.`*interface* ファイル
- `/etc/nodename` ファイル
- `/etc/defaultdomain` ファイル
- `/etc/defaultrouter` ファイル (オプション)
- `hosts` データベース
- `ipnodes` データベース

- netmasks データベース (オプション)

Solaris インストールプログラムは、インストール処理の一環として上記のファイルを作成します。これらのファイルは、この「TCP/IP 構成ファイル」の節の説明に従って手作業で編集することもできます。hosts データベースと netmasks データベースは、Solaris ネットワークで使用できるネームサービスが読み取るネットワークデータベースのうちの 2 つです。ネットワークデータベースの概念については、99 ページの「ネットワークデータベースと nsswitch.conf ファイル」で詳しく説明します。ipnodes ファイルについての詳細は、339 ページの「/etc/inet/ipnodes ファイル」を参照してください。

/etc/hostname.*interface* ファイル

このファイルは、IPv4 を使用するローカルホスト上のネットワークインタフェースを定義します。ローカルマシンには、/etc/hostname.*interface* ファイルが少なくとも 1 つ必要です。このファイルは、Solaris インストールプログラムが作成します。ファイル名中の *interface* には、一次ネットワークインタフェースのデバイス名が入ります。

注 – Solaris ソフトウェアの初期インストール後に、システムに新しいネットワークインタフェースを追加する場合は、そのインタフェースについて /etc/hostname.*interface* ファイルを作成し、インタフェースの IP アドレスを /etc/inet/hosts ファイルに追加し、-r オプションでシステムをリブートする必要があります。64 ページの「ローカルファイルモードの場合のホストの構成方法」で説明している手順を参照してください。また、Solaris ソフトウェアが新しいネットワークインタフェースを認識し、使用できるようにするには、インタフェースのデバイスドライバが適切なディレクトリに読み込まれるようにする必要があります。新しいネットワークインタフェースに付属しているマニュアルを参照し、正しいインタフェース名とデバイスドライバの使用方法を確認してください。

このファイルにはエントリが 1 つだけ入っています。それは、ネットワークインタフェースに結び付いているホスト名または IPv4 アドレスのどちらかです。たとえば、tenere というマシンの一次ネットワークインタフェースが smc0 であるとします。この場合、/etc/hostname.*interface* ファイルの名前は /etc/hostname.smc0 となります。このファイルには tenere というエントリが入っています。

複数のネットワークインタフェースのためのファイル

マシンが複数のネットワークインタフェースを持っている場合は、2 番目以降のネットワークインタフェース用の /etc/hostname.*interface* ファイルを、ネットワーク管理者が追加作成する必要があります。これらのファイルはテキストエディタを使用して作成します。Solaris インストールプログラムは、追加のファイルは作成しません。

たとえば、図 4–1 に示したマシン timbuktu について考えてみましょう。このマシンには、2 つのネットワークインタフェースがあり、ルーターとして機能します。一次ネットワークインタフェース le0 は、ネットワーク 192.9.200 に接続されています。その IP アドレスは 192.9.200.70 で、ホスト名は timbuktu です。Solaris インストールプログラムによって、一次ネットワークインタフェースにファイル /etc/hostname.le0 が作成され、ホスト名 timbuktu がファイルに入力されます。

第 2 のネットワークインタフェースは le1 です。このネットワークインタフェースはネットワーク 192.9.201 に接続されています。このインタフェースは物理的にはマシン timbuktu にインストールされていますが、別の IPv4 アドレスを持つ必要があります。そのため、このインタフェースに対して /etc/hostname.le1 ファイルを手動で作成する必要があります。このファイルに入れるエントリは、ルーター名の timbuktu-201 です。

/etc/hostname6.*interface* ファイル

IPv6 は初期設定で /etc/hostname6.*interface* ファイルを使用し、IPv4 における /etc/hostname.*interface* と同様の方法で、ネットワークインタフェースを自動的に定義します。/etc/hostname. ファイル または /etc/hostname6. ファイルの少なくともどちらか一方が、ローカルマシン上に存在する必要があります。これらのファイルは、Solaris インストールプログラムで生成されます。ファイル名については、「*interface*」を主ネットワークインタフェースのデバイス名で置き換えます。/etc/hostname6.*interface* ファイルについての詳細は、326 ページの「IPv6 ネットワークインタフェース構成ファイル」を参照してください。

/etc/nodename ファイル

このファイルにはエントリが 1 つ入っています。それは、ローカルマシンのホスト名です。たとえば、マシン timbuktu では、/etc/nodename ファイルには timbuktu というエントリが入ります。

/etc/defaultdomain ファイル

このファイルにはエントリが 1 つ入っています。それは、ローカルホストのネットワークが属している管理ドメインの完全指定のドメイン名です。ネットワーク管理者は、この名前を Solaris インストールプログラムに指示したり、また後日にこのファイルを編集することができます。

図 4–1 では、ネットワークはドメイン deserts.worldwide に属しており、このドメインは .com ドメインとして分類されています。したがって、/etc/defaultdomain には deserts.worldwide.com というエントリが入ります。ネットワークドメインについての詳細は、『 *Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

/etc/defaultrouter ファイル

このファイルには、直接ネットワークに接続されている各ルーターについてのエントリが入っています。このエントリは、ネットワーク間のルーターとして機能するネットワークインタフェースの名前です。

図 4–1 で、ネットワークインタフェース `le1` は、マシン `timbuktu` をネットワーク 192.9.201 に接続しています。このインタフェースには、`timbuktu-201` という一意な名前が付いています。したがって、ネットワーク 192.9.201 にあってローカルファイルモードで構成されているマシンについては、`/etc/defaultrouter` に `timbuktu-201` という名前がエントリとして入ります。

hosts データベース

`hosts` データベースには、ネットワーク上のマシンの IPv4 アドレスとホスト名が入っています。NIS、NIS+、DNS (またはネームサービスとしての LDAP) のどれかのネームサービスを使用している場合は、`hosts` データベースは、ホスト情報用として指定されているデータベースに格納されます。たとえば、NIS+ を実行するネットワークでは、`hosts` データベースはホストテーブルに格納されます。

ネームサービスとしてローカルファイルを使用している場合は、`hosts` データベースは `/etc/inet/hosts` ファイルに格納されます。このファイルには、一次ネットワークインタフェースのホスト名と IPv4 アドレス、マシンに備わっている他のネットワークインタフェース、このマシンが検査する必要がある他のネットワークアドレスが入っています。

注 – BSD ベースのオペレーティングシステムとの互換性を確保するために、`/etc/hosts` ファイルは `/etc/inet/hosts` へのシンボリックリンクになっています。

/etc/inet/hosts ファイルの形式

`/etc/inet/hosts` ファイルには、次のような基本構文を使用します。構文についての詳細は、`hosts(4)` のマニュアルページを参照してください。

IPv4-address hostname [nicknames] [#comment]

IPv4-address には、ローカルホストが認識する必要のある各インタフェースの IPv4 アドレスが入ります。

hostname には、設定時にマシンに割り当てたホスト名と、ローカルホストが認識しなければならない増設ネットワークインタフェースに割り当てたホスト名が入ります。

[nickname] は、ホストのニックネームが入ります (省略可能)。

[# comment] は、コメントを入れます (省略可能)。

初期 /etc/inet/hosts ファイル

Solaris インストールプログラムを実行すると、プログラムは初期 /etc/inet/hosts ファイルを設定します。このファイルには、ローカルホストにとって必要最小限のエントリが入っています。エントリには、ループバックアドレス、ホストの IPv4 アドレス、ホスト名が入っています。

たとえば、図 4–1 に示したマシン tenere については、Solaris インストールプログラムは次のような /etc/inet/hosts ファイルを作成します。

例 **5–1** マシン tenere 用の /etc/inet/hosts ファイル

```
127.0.0.1     localhost         loghost    #loopback address
192.9.200.3   tenere                       #host name
```

ループバックアドレス

例 5–1 では、IPv4 アドレス 127.0.0.1 はループバックアドレスです。ループバックアドレスは、ローカルマシンがプロセス間通信するために使用する予約済みネットワークインタフェースです。これを使用して、ホストは自分自身にパケットを送信できます。78 ページの「ifconfig コマンド」で説明するように、ループバックアドレスは、構成とテストのために ifconfig コマンドにより使用されます。TCP/IP ネットワーク上のすべてのマシンは、IP アドレス 127.0.0.1 をローカルホスト用に使用する必要があります。

ホスト名

IPv4 アドレス 192.9.200.1 と名前 tenere は、ローカルマシンのアドレスとホスト名です。これらは、マシンの一次ネットワークインタフェースに割り当てられます。

複数のネットワークインタフェース

マシンには複数のネットワークインタフェースを持つものがあり、これらはルーターまたはマルチホームホストとなります。マシンに接続される増設ネットワークインタフェースごとに、専用の IPv4 アドレスとそれに割り当てる名前が必要です。ルーターまたはマルチホームホストを構成するときは、この情報を手作業でルーターの /etc/inet/hosts ファイルに追加する必要があります。ルーターとマルチホームホストの設定についての詳細は、70 ページの「ルーターの構成」を参照してください。

例 5–2 は、図 4–1 に示したマシン timbuktu 用の /etc/inet/hosts ファイルです。

例 **5–2** マシン timbuktu 用の /etc/inet/hosts ファイル

```
127.0.0.1      localhost     loghost
192.9.200.70   timbuktu      #This is the local host name
192.9.201.10   timbuktu-201  #Interface to network 192.9.201
```

timbuktu は、この 2 つのインタフェースを使用してネットワーク 192.9.200 と 192.9.201 をルーターとして接続します。

ネームサービスの hosts データベースに対する影響

NIS、NIS+、DNS (またはネームサービスとしての LDAP) の各ネームサービスは、ホスト名とアドレスを 1 つまたは複数のサーバーで維持します。これらのサーバーは、各サーバーのネットワーク上のすべてのホストとルーター (もしあれば) に関する情報を含む hosts データベースを保持しています。これらのサービスの詳細については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』と『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照してください。

ローカルファイルがネームサービスを提供する場合

ローカルファイルをネームサービスとして使用するネットワークでは、ローカルファイルモードで実行されているマシンは、各自の /etc/inet/hosts ファイルを調べて、ネットワーク上の他のマシンの IPv4 アドレスとホスト名を入手します。したがって、このマシンの /etc/inet/hosts ファイルには以下の事項が含まれている必要があります。

- ループバックアドレス
- ローカルマシン (一次ネットワークインタフェース) の IPv4 アドレスとホスト名
- このマシンに接続している増設ネットワークインタフェース (もしあれば) の IPv4 アドレスとホスト名
- ローカルネットワーク上のすべてのホストの IPv4 アドレスとホスト名
- このマシンが認識する必要のあるルーター (もしあれば) の IPv4 アドレスとホスト名
- このマシンでホスト名を使用して参照したいマシンの IPv4 アドレス

次のコード例は、マシン tenere の /etc/inet/hosts ファイルを示しています。このマシンはローカルファイルモードで実行されます。このファイルには、192.9.200 ネットワーク上のすべてのマシンの IPv4 アドレスとホスト名が含まれているという点に注意してください。また、このファイルにはインタフェース名 timbuktu-201 とその IPv4 アドレスが含まれています。このインタフェースは 192.9.200 ネットワークを 192.9.201 ネットワークに接続しています。

ネットワーククライアントとして構成されているマシンは、ローカル /etc/inet/hosts ファイルから、自己のループバックアドレスと IPv4 アドレスを入手します。

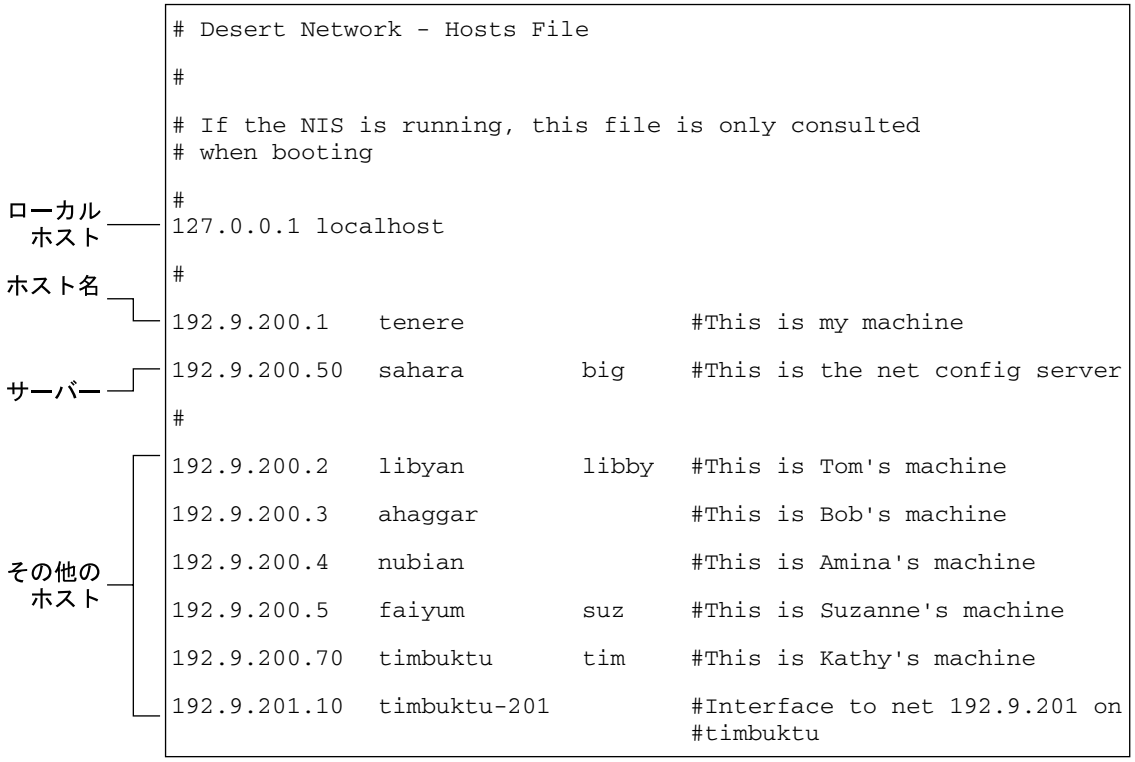


図 **5–1** ローカルファイルモードで実行されるマシン用の /etc/inet/hosts ファイル

ipnodes データベース

ipnodes データベースには、ネットワーク上の各マシンの IPv6 アドレスとホスト名が格納されています。NIS、NIS+、DNS (またはネームサービスとしての LDAP) のどれかのネームサービスを使用している場合は、ipnodes データベースは、ホスト情報用として指定されているデータベースに格納されます。たとえば、NIS+ を実行しているネットワークでは、ipnodes データベースはホストテーブル内に保持されます。ipnodes データベースについての詳細は、339 ページの「/etc/inet/ipnodes ファイル」を参照してください。

netmasks データベース

ネットワーク構成の一環として netmasks データベースを編集する必要があるのは、ネットワークをサブネット化してある場合だけです。netmasks データベースは、各ネットワークとそれに対応するサブネットマスクのリストで構成されています。

注 – サブネットを作成するときは、新規の各ネットワークはそれぞれ独立した物理ネットワークであることが必要です。単一の物理ネットワークにサブネット化を適用することはできません。

サブネット化とは

サブネット化は、限られた 32 ビット IPv4 アドレス指定空間を最大限に活用し、大規模ネットワークでのルーティングテーブルの大きさを減らすための方法の 1 つです。どのようなアドレスクラスの場合も、サブネット化によってホストアドレス空間の一部をネットワークアドレスに割り当て、ネットワーク数を増やすことができます。新規のネットワークアドレスに割り当てられるホストアドレス空間の部分を、サブネット番号と言います。

IPv4 アドレス空間を有効活用できることの他に、サブネット化には管理上の利点もいくつかあります。ネットワークの数が増えるに伴って、ルーティングはきわめて複雑になってきます。たとえば、小規模の組織なら、個々のローカルネットワークにクラス C の番号を割り当てることができます。しかし、組織が成長するにつれて、多数の異なるネットワーク番号を管理することは、非常に複雑な作業になってきます。このような場合の改善策の 1 つとして、組織内の主要部門に対してそれぞれクラス B のネットワーク番号を割り当てる方法が考えられます。たとえば、エンジニアリング部門に対して 1 つ、オペレーション部門に対して 1 つというように番号を割り当てます。その上で、サブネット化によって得られたネットワーク番号を使用して、個々のクラス B ネットワークをさらに多くのネットワークに分割できます。これによって、ルーター間でやりとりしなければならないルーティング情報の量も減少します。

IPv4 アドレス用のネットワークマスクの作成

サブネット化工程の一環として、ネットワーク全体のネットマスクを選択する必要があります。ネットマスクは、ホストアドレス空間の中で、どの位置の何個のビットがサブネット番号を表し、どの位置の何個のビットがホスト番号を表すかを決定します。完全な IPv4 アドレスは 32 ビットで構成されることを思い出してください。ホストアドレス空間を表すために使用できるビット数は、アドレスクラスによって異なりますが、最大 24 ビット、最小 8 ビットです。ネットマスクは `netmasks` データベース内に指定します。

サブネットの使用を予定している場合は、TCP/IP を構成する前にネットマスクを決定する必要があります。ネットワーク構成の一環としてオペレーティングシステムをインストールすることを予定している場合は、Solaris インストールプログラムは、ネットワークのネットマスクを指定するよう求めます。

47 ページの「ネットワーク番号の管理」で説明したように、32 ビットの IP アドレスは、ネットワーク部とホスト部で構成されています。32 ビットは 4 個のバイトに分かれます。各バイトは、ネットワーククラスに応じて、ネットワーク番号かホスト番号のどちらかに割り当てられます。

たとえば、クラス B の IPv4 アドレスでは、左側の 2 バイトがネットワーク番号に割り当てられ、右側の 2 バイトがホスト番号に割り当てられます。クラス B の IPv4 アドレス 129.144.41.10 の場合、右側の 2 バイトをホストに割り当てることができます。

サブネット化を行う場合は、ホスト番号に割り当てるバイトの中の一部のビットを、サブネットアドレスとして使用する必要があります。たとえば、ホストアドレス空間が 16 ビットであれば、65,534 個のホストのアドレス指定が可能です。3 番目のバイトをサブネットアドレス用に使用して、4 番目のバイトをホストアドレス用に使用するとすれば、最大 254 のネットワークのアドレスと、それぞれについて最大 254 ずつのホストのアドレスを指定できます。

ホストアドレスのバイトのどのビットがサブネットアドレスに使用され、どのビットがホストアドレスに使用されるかは、サブネットマスクによって決まります。サブネットマスクは、バイトの中のどのビットをサブネットアドレス用とするかを選択するために使用します。ネットマスクのビットは連続していなければなりませんが、バイトの境界に整列している必要はありません。

ネットマスクは、ビット単位の論理積演算子を使用して IPv4 アドレスに適用できます。この演算によって、アドレスのネットワーク番号とサブネット番号の位置が選択されます。

ネットマスクは、2 進数表現の視点で説明します。2 進数と 10 進数は計算機を使用して換算できます。以下の例では、ネットマスクの 10 進数形式と 2 進数形式の両方を示してあります。

ネットマスク 255.255.255.0 を IPv4 アドレス 129.144.41.101 に適用した場合、結果の IPv4 アドレスは 129.144.41.0 になります。

129.144.41.101 & 255.255.255.0 = 129.144.41.0

2 進数形式では、この演算は次のようになります。

10000001.10010000.00101001.01100101 (IPv4 アドレス)

11111111.11111111.11111111.00000000 (IPv4 ネットマスク)

これで、システムは、ネットワーク番号 129.144 の代わりにネットワーク番号 129.144.41 を捜すようになります。129.144.41 の番号を持つネットワークがあれば、システムはそれを見つけ出します。IPv4 アドレス空間の 3 番目のバイトには最大 254 個の値を割り当てることができるので、サブネット化によって、254 個のネットワーク用のアドレス空間を作ることができます。サブネット化を使用しなければ、ネットワークは 1 つだけです。

ネットワークを 2 つだけ追加するためのアドレス空間を確保する場合は、次のようなサブネットマスクを使用します。

255.255.192.0

このネットマスクの結果は次のようになります。

11111111.11111111.1100000.00000000

ホストアドレス用に使用できるビットが、まだ 14 ビット残っています。全桁 0 と全桁 1 は予約済みなので、少なくとも 2 ビットをホスト番号用として確保する必要があります。

/etc/inet/netmasks ファイル

ネットワークで NIS、NIS+、または LDAP を実行する場合は、これらのネームサービスを提供するサーバーは netmasks データベースを保持しています。ローカルファイルをネームサービスとして使用するネットワークの場合は、この情報は /etc/inet/netmasks ファイル内に格納されます。

注 – BSD ベースのオペレーティングシステムとの互換性を確保するために、/etc/netmasks ファイルは /etc/inet/netmasks へのシンボリックリンクになっています。

次のコード例に示すのは、クラス B ネットワーク用のサンプルの /etc/inet/netmasks ファイルです。

例 **5–3** クラス B ネットワーク用の /etc/inet/netmasks ファイル

```
## The netmasks file associates Internet Protocol (IPv4) address
# masks with IPv4 network numbers.
#
#     network-number    netmask
#
# Both the network-number and the netmasks are specified in
# "decimal dot" notation, e.g:
#
#        128.32.0.0   255.255.255.0
129.144.0.0  255.255.255.0
```

このファイルが存在しない場合は、次の構文を使用して作成してください。

network-number netmask-number

詳細は、netmasks(4) のマニュアルページを参照してください。

ネットマスク番号を作成するときは、InterNIC から割り当てられたネットワーク番号 (サブネット番号ではない) とネットマスク番号を、/etc/inet/netmasks ファイルに入力します。各サブネットマスクはそれぞれ単独の行に入れてください。

たとえば、

```
128.78.0.0        255.255.248.0
```

/etc/inet/hosts ファイルに、ネットワーク番号の記号名を入力することもできます。そうすれば、ネットワーク番号の代わりにこれらのネットワーク名を、コマンドへのパラメータとして使用できます。

ネットワークデータベースと nsswitch.conf ファイル

ネットワークデータベースは、ネットワークを構成するために必要な情報を提供するファイルです。ネットワークデータベースには次のものがあります。

- hosts
- ipnodes
- netmasks
- ethers
- bootparams
- protocols
- services
- networks

構成工程の一環として、ネットワークをサブネット化する場合は、hosts データベースと netmasks データベースを編集します。マシンをネットワーククライアントとして構成するには、bootparams と ethers の 2 つのネットワークデータベースを使用します。残りのデータベースはオペレーティングシステムが使用するもので、編集が必要になることはほとんどありません。

nsswitch.conf ファイルは、ネットワークデータベースではありませんが、関連するネットワークデータベースとともに構成する必要があります。nsswitch.conf は、特定のマシンに、NIS、NIS+、DNS、ローカルファイル、または LDAP のどのネームサービスを使用するかを指定します。

ネットワークデータベースへのネームサービスの影響

ネットワークデータベースの形式は、ネットワーク用として選択するネームサービスの種類によって異なります。たとえば、hosts データベースには、少なくとも、ローカルマシンとそのマシンに直接接続されているネットワークインタフェースのホスト名と IPv4 アドレスだけは入っています。しかし、ネットワークで使用するネームサービスの種類によっては、その他の IPv4 アドレスとホスト名も hosts データベースに入ることがあります。

ネットワークデータベースは次のように使用されます。

- ローカルファイルをネームサービスとして使用するネットワークは、/etc/inet ディレクトリと /etc ディレクトリの中のファイルを使用する
- NIS+ は NIS+ テーブルと呼ばれるデータベースを使用する
- NIS は NIS マップと呼ばれるデータベースを使用する
- DNS はホスト情報が入ったレコードを使用する

注 – DNS のブートファイルとデータファイルは、直接的にはネットワークデータベースに対応していません。

図 5–2 に、これらのネームサービスにより使用される hosts データベースの形式を示します。

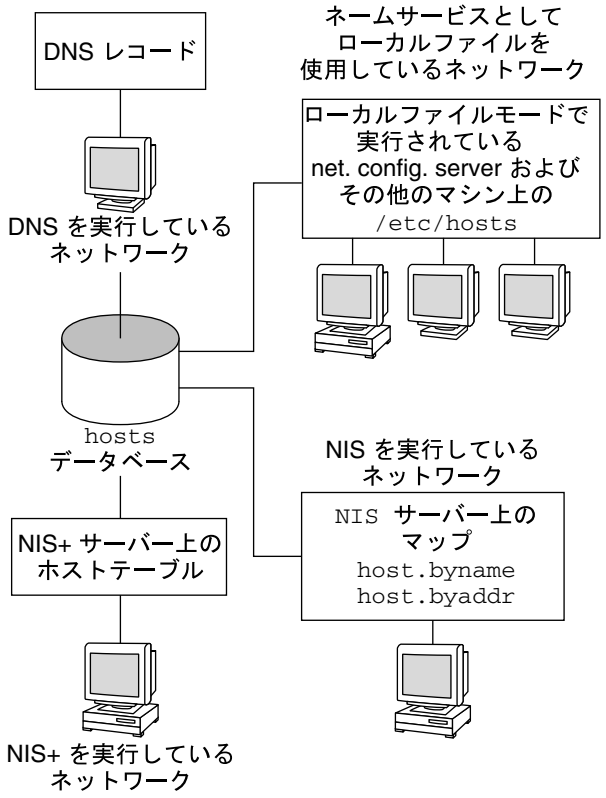


図 **5–2** ネームサービスが使用する hosts データベースの形式

表 5–1 に、ネットワークデータベースと、各ネットワークデータベースに対応するローカルファイル、NIS+ および NIS のネームサービスファイルを示します。

表 **5–1** ネットワークデータベースと対応するネームサービスファイル

ネットワークデータベース	ローカルファイル	**NIS+** のテーブル	**NIS** のマップ
hosts	/etc/inet/hosts	hosts.org_dir	hosts.byaddr hosts.byname
ipnodes	/etc/inet/ipnodes	ipnodes.org_dir	ipnodes.byaddr ipnodes.byname
netmasks	/etc/inet/netmasks	netmasks.org_dir	netmasks.byaddr
ethers	/etc/ethers	ethers.org_dir	ethers.byname ethers.byaddr
bootparams	/etc/bootparams	bootparams.org_dir	bootparams
protocols	/etc/inet/protocols	protocols.org_dir	protocols.byname protocols.bynumber
services	/etc/inet/services	services.org_dir	services.byname
networks	/etc/inet/networks	networks.org_dir	networks.byaddr networks.byname

本書では、ローカルファイルをネームサービスとして使用するネットワークで使用されるものとして、ネットワークデータベースの説明を進めます。hosts データベースについては、92 ページの「hosts データベース」を参照してください。ipnodes データベースについては、339 ページの「/etc/inet/ipnodes ファイル」を参照してください。netmasks データベースについては、95 ページの「netmasks データベース」を参照してください。NIS、NIS+、DNS、LDAP でのネットワークデータベースの対応付けについては、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』と『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照してください。

nsswitch.conf ファイル — 使用するネームサービスの指定

/etc/nsswitch.conf ファイルは、ネットワークデータベースの検索順序を定義します。Solaris インストールプログラムは、インストール中にネットワーク管理者が指定するネームサービスに基づいて、ローカルマシン用のデフォルトの /etc/nsswitch.conf ファイルを作成します。"None" オプションを指定して、ローカルファイルをネームサービスとして使用することを指示した場合は、nsswitch.conf ファイルは例 5–4 のようになります。

例 **5–4** ネームサービスにファイルを使用するネットワーク用の nsswitch.conf

```
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf;
```

例 **5–4** ネームサービスにファイルを使用するネットワーク用の nsswitch.conf　(続き)

```
# it does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file contains "switch.so" as a
# nametoaddr library for "inet" transports.

passwd:          files
group:           files
hosts:           files
networks:        files
protocols:       files
rpc:             files
ethers:          files
netmasks:        files
bootparams:      files
publickey:       files
# At present there isn't a 'files' backend for netgroup; the
# system will figure it out pretty quickly,
# and won't use netgroups at all.
netgroup:        files
automount:       files
aliases:         files
services:        files
sendmailvars:    files
```

このファイルについての詳細は、nsswitch.conf(4) のマニュアルページに説明されています。このファイルの基本構文は次のとおりです。

database name-service-to-search

database フィールドには、オペレーティングシステムが検索するさまざまの種類のデータベースを指定できます。たとえば、passwd や aliases などのようにユーザーに影響を与えるデータベースでも、またネットワークデータベースでも指定できます。ネットワークデータベースの場合、 *name-service-to-search* パラメータの値は、files、nis、 nis+ のどれかです。hosts データベースの場合は、検索するネームサービスとして dns も値に指定できます。nis+ と files というように、複数のネームサービスを指定することもできます。

例 5–4 に検索オプションとして示されているのは、files だけです。したがって、ローカルマシンは、/etc ディレクトリと /etc/inet ディレクトリに入っているファイルから、ネットワークデータベース情報のほか、セキュリティと自動マウントに関する情報を入手します。

nsswitch.conf の変更

/etc ディレクトリには、Solaris インストールプログラムが作成した nsswitch.conf ファイルが入っています。そのほかに、次のネームサービス用のテンプレートファイルも入っています。

- nsswitch.files
- nsswitch.nis
- nsswitch.nis+

あるネームサービスから別のネームサービスに変更したい場合は、対応するテンプレートを nsswitch.conf にコピーすることができます。また、nsswitch.conf ファイルを選択的に編集して、個々のデータベースを見つけるために検索するデフォルトのネームサービスを変更することができます。

たとえば、NIS を実行するネットワークでは、ネットワーククライアントについての nsswitch.conf ファイルの変更が必要な場合があります。bootparams データベースと ethers データベースの検索順序では、最初のオプションとして files、次に nis が指定されている必要があります。次のコード例に、正しい検索順序を示します。

例 **5–5** NIS を実行するネットワーク上のクライアントのための nsswitch.conf

```
## /etc/nsswitch.conf:#
.
.
passwd:         files nis
group:          file nis

# consult /etc "files" only if nis is down.
hosts:          nis     [NOTFOUND=return] files
networks:       nis     [NOTFOUND=return] files
protocols:      nis     [NOTFOUND=return] files
rpc:            nis     [NOTFOUND=return] files
ethers:         files   [NOTFOUND=return] nis
netmasks:       nis     [NOTFOUND=return] files
bootparams:     files   [NOTFOUND=return] nis
publickey:      nis
netgroup:       nis

automount:      files nis
aliases:        files nis

# for efficient getservbyname() avoid nis
services:       files nis
sendmailvars:   files
```

ネームサービススイッチの詳細については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』と『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照してください。

bootparams データベース

bootparams データベースには、ネットワーククライアントモードでブートするように構成されているマシンが使用する情報が入っています。ネットワーククライアントを持つネットワークの場合は、このデータベースの編集が必要になります。手順については、66 ページの「ネットワーククライアントの構成」を参照してください。このデータベースは /etc/bootparams ファイルに入力した情報をもとにして構築されます。

このデータベースの構文についての詳細は、bootparams(4) のマニュアルページで説明されています。基本構文は次のとおりです。

machine-name file-key-server-name:pathname

個々のディスクレスまたはネットワーククライアントマシンについて、エントリが 1 つずつあります。各エントリに入っている情報は、クライアント名、キーのリスト、サーバー名、パス名です。

各エントリの最初の項目は、クライアントマシンの名前です。その次は、キー、サーバー名、パス名をタブ文字で区切ったリストです。最初の項目以外は、すべてオプションです。次に例を示します。

例 **5–6** bootparams データベース

```
myclient   root=myserver : /nfsroot/myclient  \
swap=myserver : /nfsswap//myclient \
dump=myserver : /nfsdump/myclient
```

この例の dump=: は、ダンプファイルを捜さないようにクライアントホストに指示します。

bootparams のワイルドカードエントリ

クライアントをサポートするように bootparams データベースを編集するときには、ほとんどの場合、ワイルドカードエントリを使用する方が便利です。次のようにしてワイルドカードエントリを使用します。

```
*  root=server:/path dump=:
```

アスタリスク (*) ワイルドカードは、このエントリが、bootparams データベース内で明示的に指定されていないすべてのクライアントに適用されることを示します。

ethers データベース

ethers データベースは、/etc/ethers ファイルに入力した情報をもとにして構築されます。このデータベースは、ホスト名を Ethernet アドレスに関連付けます。ethers データベースの作成が必要になるのは、RARP デーモンを実行する場合だけです。つまり、ネットワーククライアントを構成する場合だけです。

RARP は、このファイルを使用して、Ethernet アドレスを IP アドレスにマップします。RARP デーモン `in.rarpd` を実行するときは、`ethers` ファイルを設定し、このデーモンを実行するすべてのホストでこのファイルを維持して、ネットワークに対する変更が反映されるようにする必要があります。

このデータベースの構文についての詳細は、`ethers(4)` のマニュアルページに説明されています。基本構文は次のとおりです。

Ethernet-address hostname #comment

Ethernet-address は、ホストの Ethernet アドレスです。

hostname は、ホストの公式名です。

#comment は、ファイル内のエントリに付加したい任意の注意書きです。

Ethernet アドレスは装置の製造元から提供されます。マシンの電源を入れたときに Ethernet アドレスが表示されない場合は、ハードウェアのマニュアルを調べてください。

`ethers` データベースにエントリを追加するときは、ホスト名が、ニックネームではなく、`hosts` データベースと `ipnodes` データベース内の一次名に一致していることを確かめてください (次のコード例)。

例 **5–7** `ethers` データベース内のエントリ

```
8:0:20:1:40:16  fayoum
8:0:20:1:40:15  nubian
8:0:20:1:40:7   sahara    # This is a comment
8:0:20:1:40:14  tenere
```

その他のネットワークデータベース

残りのネットワークデータベースについては、編集が必要になることはほとんどありません。

`networks` データベース

`networks` データベースは、ネットワーク名をネットワーク番号に関連付けて、一部のアプリケーションが番号の代わりに名前を使用し表示できるようにします。`networks` データベースは、`/etc/inet/networks` ファイルの中の情報をもとにして作られます。このデータベースには、このネットワークがルーターを介して接続されるすべてのネットワークの名前が入っています。

初期 `networks` データベースは、Solaris インストールプログラムが設定します。ただし、既存のネットワークトポロジに新たなネットワークを追加する場合は、このデータベースを更新する必要があります。

`/etc/inet/networks` の詳しい構文は、`networks(4)` のマニュアルページで説明されています。基本構文は次のとおりです。

network-name network-number nickname(s) #comment

network-name は、ネットワークの公式名です。

network-number は、InterNIC から割り当てられた番号です。

nickname は、ネットワークの認識のために使用されるその他の名前です。

#comment は、ファイル内のエントリに付加したい任意の注意書きです。

networks ファイルは必要に応じて更新する必要があります。netstat プログラムは、このデータベース内の情報を使用して状態テーブルを作成します。

次のコード例に、/etc/networks ファイルのサンプルを示します。

例 **5–8** /etc/networks ファイル

```
#ident    "@(#)networks    1.4    92/07/14 SMI"    /* SVr4.0 1.1    */
#
# The networks file associates Internet Protocol (IP) network
# numbers with network names. The format of this file is:
#
#     network-name              network-number              nicnames . . .

# The loopback network is used only for intra-machine communication
loopback              127

#
# Internet networks
#
arpanet     10       arpa  # Historical
ucb-ether   46       ucbether
#
# local networks

eng   193.9.0  #engineering
acc   193.9.1  #accounting
prog  193.9.2  #programming
```

protocols データベース

protocols データベースには、システムにインストールされている TCP/IP プロトコルとそれぞれの番号のリストが入っています。このデータベースは、Solaris インストールプログラムが自動的に作成します。このファイルの管理が必要になることはほとんどありません。

protocols データベースには、システムにインストールされている TCP/IP プロトコルの名前が含まれています。詳しい構文については、protocols(4) のマニュアルページを参照してください。次のコード例に、/etc/inet/protocols ファイルのサンプルを示します。

例 **5–9** /etc/inet/protocols ファイル

```
#
# Internet (IP) protocols
#
ip    0   IP    # internet protocol, pseudo protocol number
icmp  1   ICMP  # internet control message protocol
tcp   6   TCP   # transmission control protocol
udp  17   UDP   # user datagram protocol
```

services データベース

services データベースには、TCP サービスと UDP サービスの名前と、それぞれのよく知られているポート番号のリストが入っています。このデータベースは、ネットワークサービスを呼び出すプログラムにより使用されます。Solaris インストールプログラムは、services データベースを自動的に作成します。通常は、このデータベースは管理作業が必要になることはありません。

詳しい構文は、services(4) のマニュアルページに記載されています。次のコード例に、典型的な /etc/inet/services ファイルからの抜粋を示します。

例 **5–10** /etc/inet/services ファイル

```
#
# Network services
#
echo      7/udp
echo      7/tcp
discard   9/udp     sink null
discard   11/tcp
daytime   13/udp
daytime   13/tcp
netstat   15/tcp
ftp-data  20/tcp
ftp       21/tcp
telnet    23/tcp
time      37/tcp    timeserver
time      37/udp    timeserver
name      42/udp    nameserver
whois     43/tcp    nickname
```

ブート処理

注 – 起動スクリプトの名前は、Solaris リリースごとに変更されることがあります。

1. ホストでオペレーティングシステムを起動します。
2. カーネルが、ブート処理の一部として `/sbin/init` を実行します。
3. `/sbin/init` が、`/etc/rcS.d/S30rootusr.sh` 起動スクリプトを実行します。
4. この起動スクリプトは、ディスクレスとデータレスの操作のための最小限のホスト構成とネットワーク構成の確立など、いくつかのシステム起動処理を行います。また、このスクリプトは、`/usr` ファイルシステムをマウントします。
   a. ローカルデータベースファイルに、必要な構成情報 (ホスト名と IP アドレス) が含まれている場合は、スクリプトはそれを使用します。
   b. ローカルホスト構成ファイル内に必要な情報がない場合は、`/etc/rcS.d/S30rootusr.sh` は、RARP を使用してホストの IP アドレスを入手します。
5. ドメイン名、ホスト名、デフォルトのルーターアドレスがローカルファイルに含まれている場合は、マシンはそれらを使用します。ローカルファイルに構成情報が含まれていない場合は、システムは bootparams プロトコルを使用して、ホスト名、ドメイン名、デフォルトのルーターアドレスを入手します。必要な情報が、ホストと同じネットワーク上にあるネットワーク構成サーバーから入手可能でなければなりません。この時点ではまだインターネットワーク通信が存在していないので、この条件が必要になります。
6. `/etc/rcS.d/S30rootusr.sh` が作業を完了し、その他のいくつかのブート手続きが実行されると、次に `/etc/rc2.d/S69inet` が実行されます。このスクリプトは、ネームサービス (NIS、NIS+、または DNS) の開始の前に完了しておく必要のある起動処理を実行します。これらの処理には、IP の構成、ドメイン名のルーティングと設定などがあります。
7. `S69inet` の処理が完了すると、`/etc/rc2.d/S71rpc` が実行されます。このスクリプトは、NIS、NIS+、DNS のいずれかのネームサービスを起動します。
8. `/etc/rc2.d/S71rpc` の実行の後で、`/etc/rc2.d/S72inetsvc` が実行されます。このスクリプトは、ネームサービスの存在の有無に応じて異なるサービスを起動します。`S72inetsvc` は `inetd` デーモンも起動します。このデーモンは、`telnet` などのユーザーサービスを管理します。

ブート処理についての詳細は、『*Solaris* のシステム管理 *(*基本編*)*』を参照してください。

ルーティングプロトコル

Solaris オペレーティングシステムは 2 つのルーティングプロトコルをサポートしています。それは、RIP (Routing Information Protocol) と ICMP RDISC (Router Discovery Protocol) です。RIP と RDISC は、どちらも標準 TCP/IP プロトコルです。

ルーティング情報プロトコル (RIP)

RIP はルーティングデーモン `in.routed` により実現されるもので、このデーモンはマシンのブート時に自動的に起動されます。`s` オプションを指定した `in.routed` をルーターで実行すると、`in.routed` は、到達可能なすべてのネットワークへのルートをカーネルルーティングテーブルに組み入れ、すべてのネットワークインタフェースに対して「到達可能性」を通知します。

`q` オプションを指定した `in.routed` をホストで実行した場合は、`in.routed` はルーティング情報を抽出しますが、到達可能性は通知しません。ホストでは、ルーティング情報は次の 2 つの方法で抽出できます。

- `S` フラグ (大文字の "S" は「省スペースモード」の意) を指定しない。 `in.routed` は、ルーターで実行するときとまったく同じようにフルルーティングテーブルを作成します。
- `S` フラグを指定する。 `in.routed` は、使用可能なルーターについてデフォルトのルートを 1 つずつ示す最小カーネルテーブルを作成します。

ICMP ルーター検索 (RDISC) プロトコル

ホストは、RDISC を使用してルーターからルーティング情報を入手します。したがって、ホストが RDISC を実行しているときは、各ルーターは、ルーティング情報の交換のために、RIP などのような別のプロトコルも実行している必要があります。

RDISC は `in.rdisc` により実装されます。`in.rdisc` は、ルーターとホストの両方で実行している必要があります。通常は、`in.rdisc` をホストで実行すると、同じく `in.rdisc` を実行している各ルーターをデフォルトのルートに加えます。`in.rdisc` を実行しているホストは、RIP だけを実行しているルーターは検索しないので、注意してください。また、ルーターが `in.rdisc` (`in.routed` ではなく) を実行しているときは、ルーターごとに異なる優先項目を持つように構成すると、ホストができるだけ効率的なルーターを選択できるようになります。`rdisc(1M)` のマニュアルページを参照してください。

マシンがルーターかどうかを決定する方法

あるマシンがホストまたはルーターのどちらであるかを決定するのは、マシンのブート時に実行される `/etc/rc2.d/S69inet` 起動スクリプトです。この決定に伴って、ルーティングプロトコル (RIP と RDISC) を、ルーターモードで実行するかホストモードで実行するかも決まります。

`/etc/rc2.d/S69inet` スクリプトは、次の 2 つの条件が満たされているとき、マシンがルーターであると判断します。

- `/etc/hostname.`*interface* ファイルが 2 つ以上ある
- `ifconfig` コマンドにより、複数のインタフェースが “up” として構成されている (`ifconfig(1M)` のマニュアルページを参照してください)。

インタフェースが 1 つしか見つからない場合は、このスクリプトはそのマシンがホストであると判断します。70 ページの「ルーターの両方のネットワークインタフェースの構成」を参照してください。`/etc/hostname.`*interface* ファイル以外の方法で構成されているインタフェースは、判断の対象にされません。

IPv4 アドレスの構成部分

TCP/IP を実行する各ネットワークは、それぞれ一意なネットワーク番号を持っている必要があります。そのネットワーク上のすべてのマシンは、それぞれ一意な IP アドレスを持っている必要があります。ネットワークを登録し、ネットワーク番号を入手するには、その前に、IP アドレスの構造を理解しておく必要があります。この節では、IPv4 アドレスについて説明します。IPv6 アドレスについては、288 ページの「IPv6 アドレス指定」を参照してください。

IPv4 アドレスは、特定のマシンのネットワークインタフェースを一意のものとして識別する 32 ビットの番号です。IPv4 アドレスは一般に 10 進数で表され、ピリオドで区切った 4 つの 8 ビットフィールドの形式をとります。個々の 8 ビットフィールドは、それぞれ IPv4 アドレスの 1 バイトを表します。このような形式で IPv4 アドレスのバイトを表す方式を「ドット化 10 進形式」と呼びます。

IPv4 アドレスのバイトは、さらに、ネットワーク部とホスト部の 2 つの部分に分かれます。図 5–3 に、129.144.50.56 という典型的な IPv4 アドレスの構成部分を示します。

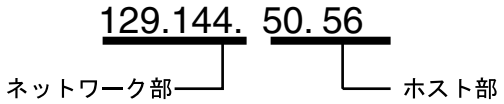


図 **5–3** IPv4 アドレスの構成部分

ネットワーク部

ネットワーク部は、ネットワークに割り当てられている一意な番号を示します。また、割り当てられているネットワーククラスも識別します。 図 5–3 では、ネットワーク部は IPv4 アドレスの 2 バイトを占めています。

ホスト部

IPv4 アドレスのこの部分は、管理者が各ホストに割り当てる番号です。ホスト番号は、ネットワーク上でこのマシンを一意なものとして識別します。ネットワーク上の各ホストについて、アドレスのネットワーク部は同じで、ホスト部はそれぞれ異なる必要があるという点に注意してください。

サブネット番号 (省略可能)

多数のホストを持つローカルネットワークは、いくつかのサブネットに分割されることがあります。ネットワークをサブネット化することにした場合は、サブネットにサブネット番号を割り当てる必要があります。IPv4 アドレスのホスト番号部の一部のビットをネットワーク識別子として使用することで、IPv4 アドレス空間の有効率を最大限にすることができます。ネットワーク識別子として使用した場合、アドレスの指定した部分がサブネット番号になります。サブネット番号は、ネットマスクを使って作成します。ネットマスクは、IPv4 アドレスのネットワーク部とサブネット部を選択するビットマスクです。詳細については、96 ページの「IPv4 アドレス用のネットワークマスクの作成」を参照してください。

ネットワーククラス

ネットワーク上での IPv4 アドレス指定に関する計画の第 1 ステップは、最も妥当なネットワーククラスを決定することです。このステップが完了したら、重要な第 2 ステップ、つまり InterNIC アドレス指定機関からのネットワーク番号の入手に進むことができます。

現在、TCP/IP ネットワークには 3 つのクラスがあります。32 ビットの IPv4 アドレス空間は、ネットワーク部のビット数が多かったり少なかったりするなど、クラスによって使い方が異なります。3 つのクラスとは、クラス A、クラス B、クラス C です。

クラス A ネットワーク番号

クラス A ネットワーク番号では、IPv4 アドレスの最初の 8 ビットが「ネットワーク部」として使用されます。残りの 24 ビットは、次の図に示すように、IPv4 アドレスのホスト部です。

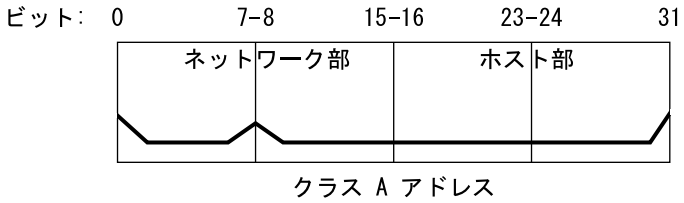


図 **5–4** クラス A アドレスのバイト割り当て

クラス A ネットワーク番号の最初のバイトに割り当てられる値の範囲は、1〜127 です。たとえば、75.4.10.4 という IPv4 アドレスがあるとします。最初のバイトの 75 という値は、このホストがクラス A ネットワーク内にあることを示しています。残りのバイトの 4.10.4 はホストアドレスを形成します。クラス A の番号の場合、InterNIC が割り当てるのは、最初の 1 バイトだけです。残りの 3 バイトをどのように使用するかは、そのネットワーク番号の所有者の自由です。クラス A のネットワークとして存在可能なのは 127 個だけです。この範囲内の各番号が、それぞれ最大 16,777,214 個のホストを収容できます。

クラス B ネットワーク番号

クラス B ネットワーク番号では、16 ビットがネットワーク番号に使用され、16 ビットがホスト番号に使用されます。クラス B ネットワーク番号の最初のバイトの値の範囲は、128〜191 です。129.144.50.56 の番号の場合、最初の 2 バイトの 129.144 は InterNIC により割り当てられるネットワークアドレスです。残りの 2 バイトの 50.56 はホストアドレスで、これはネットワーク番号の所有者が任意に割り当てることができます。図 5–5 に、クラス B のアドレスを示します。

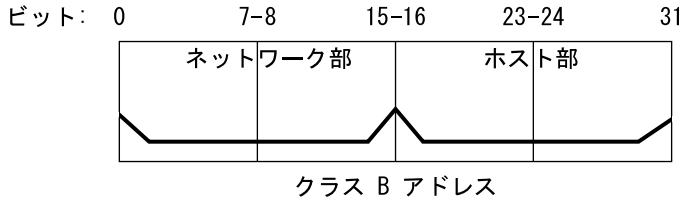


図 **5–5** クラス B アドレスのバイト割り当て

一般に、クラス B は、多数のホストを備えたネットワークを持つ組織に割り当てられます。

クラス C ネットワーク番号

クラス C ネットワーク番号では、24 ビットがネットワーク番号に使用され、8 ビットがホスト番号に使用されます。クラス C ネットワーク番号は、ホスト数が少ない、つまり最大ホスト数が 254 台程度のネットワークに適しています。クラス C ネットワーク番号は、IPv4 アドレスの最初の 3 バイトを占めます。ネットワーク番号の所有者が自由に割り当てることができるのは、4 番目のバイトだけです。図 5–6 に、クラス C アドレスのバイトを示します。

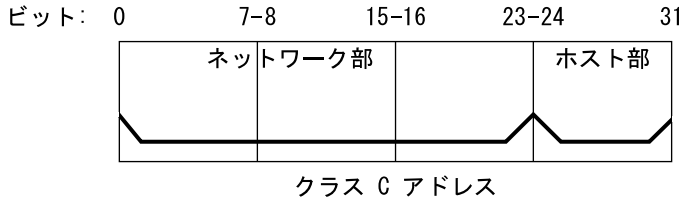


図 **5–6** クラス C アドレスのバイト割り当て

クラス C ネットワーク番号の最初のバイトの値の範囲は、192〜223 です。第 2 と第 3 のバイトの値の範囲は、どちらも 1〜255 です。典型的なクラス C アドレスは、たとえば 192.5.2.5 のようになります。最初の 3 バイトの 192.5.2 がネットワーク番号です。最後のバイト、つまり 5 がホスト番号です。

第 6 章

DHCP (トピック)

第 7 章

Solaris DHCP (概要)

この章では、動的ホスト構成プロトコル (DHCP) を紹介し、このプロトコルの基本概念、およびネットワーク上で使用した場合の利点について説明します。

この章では、以下の内容について説明します。

DHCP プロトコルについて

TCP/IP ネットワーク上のホストシステムは、その起動時に DHCP プロトコルによって、ネットワークに対し自動的に構成されます。DHCP では、クライアント/サーバーメカニズムが使用されます。サーバーは、クライアントの構成情報を格納、管理し、クライアントの要求に応じてその構成情報を提供します。構成情報には、クライアントの IP アドレスと、クライアントが使用可能なネットワークサービス情報が含まれます。

DHCP は、従来の BOOTP プロトコルをベースに機能拡張されたプロトコルです。BOOTP は、TCP/IP ネットワーク経由のブートを可能にすることを目的に設計されました。DHCP では、クライアントとサーバー間のメッセージに対し BOOTP と同じフォーマットが使用されますが、メッセージには BOOTP よりも多くの情報が含まれています。 この追加された情報は、クライアントに対するネットワーク構成データです。

DHCP の主な利点は、リースによって IP アドレスの割り当てを管理し、IP アドレスを、使用されなくなった時点で回収し、他のクライアントに再割り当てすることができることです。これによって、1 つのサイトで使用する IP アドレスプールは、すべてのクライアントに常時アドレスを割り当てた場合に比べて、小さくなります。

Solaris DHCP を使用した場合の利点

DHCP は、TCP/IP ネットワークの設定やネットワークの日々の管理に伴う、システム管理者やネットワーク管理者の手間を軽減します。なお、Solaris DHCP は IPv4 でのみ動作することに注意してください。

Solaris DHCP には、以下の利点があります。

- IP アドレス管理 – DHCP の主な利点は、IP アドレスをより簡単に管理できることです。DHCP を備えていないネットワークでは、管理者が手動で IP アドレスを割り当てる必要があります。管理者が手動で IP アドレスを割り当てる場合には、各クライアントに一意の IP アドレスを割り当て、各クライアントを個別に構成する必要があります。クライアントが別のネットワークに移動する場合には、管理者はそのクライアントのために手動で修正を加える必要があります。DHCP が使用可能な場合は、管理者が介在しなくても、DHCP サーバーが IP アドレスを管理し、割り当てます。クライアントは、別のサブネットに移動する際に新しいネットワークに適した新しいクライアント情報を DHCP サーバーから取得するため、手動による再構成は必要ありません。
- ネットワーククライアント構成の一元化 – ネットワーク管理者は、特定のクライアント、あるいは特定のクライアントタイプに特化した構成を作成し、その情報を 1 箇所に、つまり DHCP データストアでまとめて集中管理することができます。管理者は、クライアント構成を変更するためにクライアントにログインする必要はありません。DHCP データストア内の情報を変更するだけで、複数のクライアントに対する変更を実行できます。
- BOOTP クライアントのサポート – BOOTP サーバーと DHCP サーバーはどちらも、クライアントからのブロードキャストを待機して、応答します。DHCP サーバーは、DHCP クライアントからの要求だけではなく、BOOTP クライアントからの要求にも応答できます。BOOTP クライアントは、IP アドレスと、ブートに必要な情報をサーバーから受け取ります。
- ローカルおよびリモートクライアントのサポート – BOOTP は、あるネットワークから別のネットワークへのメッセージリレー (中継) 機能を備えています。DHCP は、さまざまな方法で BOOTP リレー機能を使用します。ほとんどのネットワークルーターは、BOOTP リレーエージェントとして機能するように構成できます。そのように構成されたネットワークルーターは、要求側クライアントのネットワーク上に存在しないサーバーに BOOTP 要求を渡します。同じ方法で、DHCP 要求をリレーすることも可能です。これは、ルーターには DHCP 要求と BOOTP 要求の区別がないためです。また、BOOTP リレー機能をサポートするルーターが使用できない場合には、Solaris DHCP サーバーを BOOTP リレーエージェントとして動

作するように構成することもできます。

- ネットワークブート機能 – クライアントは、DHCP を使用すると、RARP (逆アドレス解決プロトコル) や `bootparams` を使用しなくても、ネットワーク上のサーバーからブートに必要な情報を取得できます。DHCP サーバーは、IP アドレス、ブートサーバー、ネットワーク構成情報を含む、クライアントが動作するのに必要なすべての情報をクライアントに提供することができます。DHCP ネットワークブート要求は、サブネットを越えてリレーできるので、DHCP ネットワークブート機能を使用すれば、ネットワーク内のブートサーバー数を削減できます。RARP でのブートには、サブネットごとにブートサーバーが必要です。
- 大規模ネットワークのサポート - 何百万という DHCP クライアントをもつネットワークでは Solaris DHCP を使用できます。 DHCP サーバーは、マルチスレッド機能を使って多数のクライアント要求を同時に処理するとともに、大量データの処理に適したデータストアをサポートします。データストアへのアクセスは別々の処理モジュールによって行われるため、個々のサイトでは、DHCP データの保存に使用する独自のデータベースのサポートを追加することができます。

DHCP の動作

システム管理者はまず、DHCP サーバーをインストールし、構成する必要があります。構成作業の際、システム管理者は、クライアントがネットワーク上で機能するために必要なネットワーク情報を入力します。この情報が正しく設定されると、クライアントはネットワーク情報を要求し、受け取ることができます。

図 7–1 は、DHCP サービスにおける一連のイベントを示したものです。丸の中の番号は、図の後に続く説明の箇条書き番号を示しています。

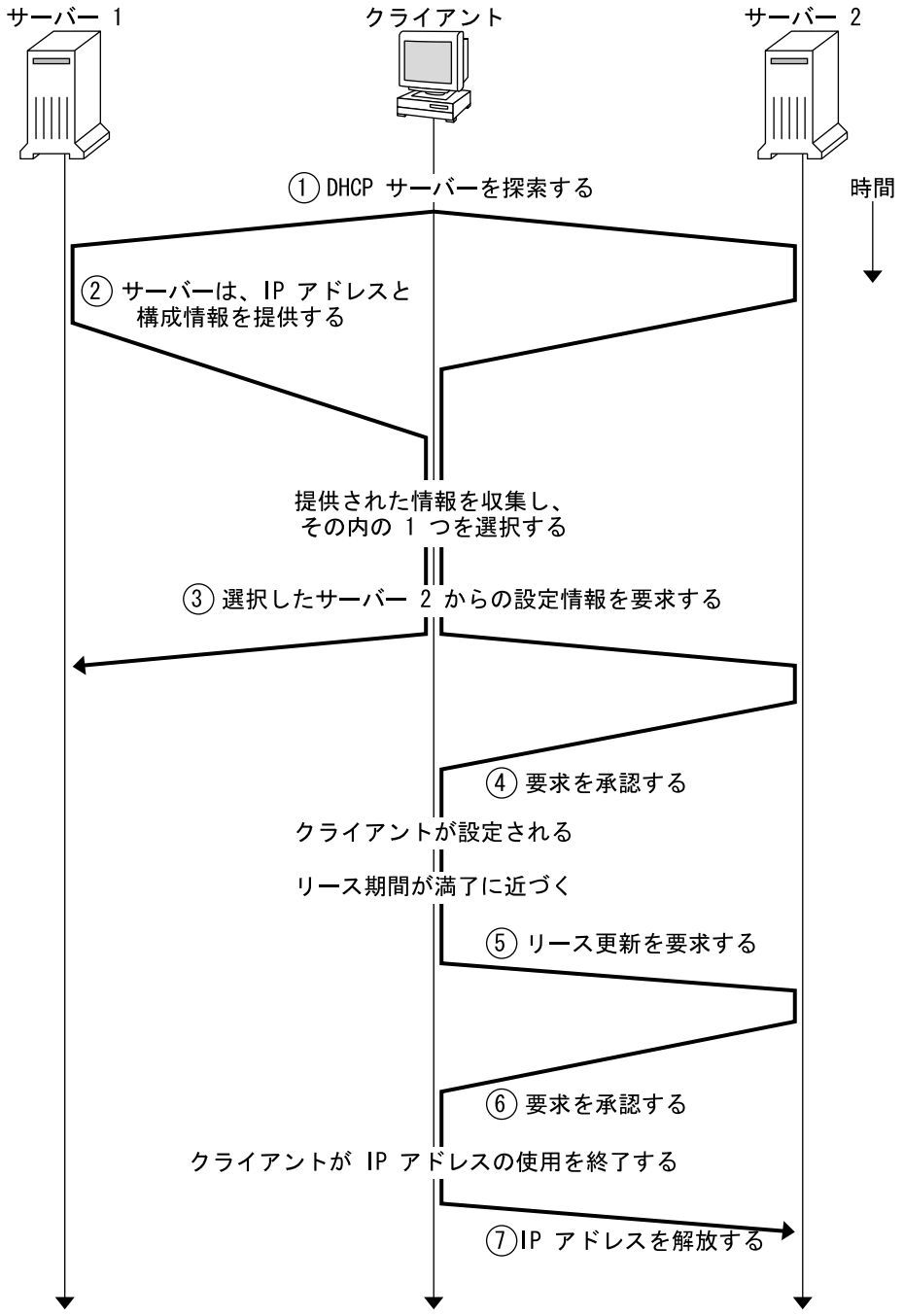


図 **7–1** DHCP サービスにおける一連のイベント

説明:

1. クライアントは、ローカルサブネット上で制限付きブロードキャストアドレス (255.255.255.255) に検索メッセージをブロードキャストすることで、DHCP サーバーを検索します。ルータが存在し、BOOTP リレーエージェントとして動作するように構成されている場合、要求は異なるサブネット上の別の DHCP サーバーに渡されます。クライアントのブロードキャストにはクライアント固有の ID が含まれています。Solaris DHCP 実装では、この ID はクライアントの MAC (Media Access Control) アドレスから抽出されます。 Ethernet ネットワークでは、MAC アドレスは Ethernet アドレスと同じです。

   検索メッセージを受け取った DHCP サーバーは、次の情報からクライアントのネットワークを特定します。

   - この要求がどのネットワークインタフェースから入ってきたか。これによってサーバーは、クライアントが、インタフェースが接続されているネットワーク上にあるのか、あるいはそのネットワークに接続された BOOTP リレーエージェントを使用しているのかがわかります。
   - BOOTP リレーエージェントの IP アドレスが要求に含まれているか。要求がリレーエージェントを通過する際に、リレーエージェントは要求ヘッダーにリレーエージェントのアドレスを挿入します。サーバーがリレーエージェントのアドレスを検出すると、サーバーは、そのアドレスのネットワーク部分がクライアントのネットワークアドレスを示していることを認識します。これは、リレーエージェントがクライアントのネットワークに接続されている必要があるからです。
   - クライアントのネットワークは、サブネット化されているか。サーバーは、リレーエージェントのアドレス、または要求を受け取ったネットワークインタフェースのアドレスが示すネットワークのサブネットマスクを `netmasks` テーブルから見つけます。サーバーは、使用されているサブネットマスクを認識すると、ネットワークアドレスのどの部分がホスト部分であるかを特定し、クライアントに適切な IP アドレスを選択できます。(ネットマスクについては、`netmasks(4)` を参照)。

2. DHCP サーバーは、クライアントのネットワークを特定すると、適切な IP アドレスを選択し、そのアドレスがまだ使用されていないことを確認します。次に、選択した IP アドレスと、クライアントの構成に使用可能なサービス情報を含むオファーメッセージをブロードキャストし、クライアントに応答します。各サーバーは、提供予定の IP アドレスを一時的に予約します。この状態は、クライアントがその IP アドレスを使用するかどうかをサーバーが確認できるまで続きます。
3. クライアントは、提供されるサービスの番号とタイプに基づいて最善のオファーを選択し、そのオファーを行なったサーバーの IP アドレスを使用するという要求をブロードキャストします。ブロードキャストにより、応答したすべての DHCP サーバーは、クライアントが 1 つのサーバーをすでに選択したことを認識し、選択されなかったサーバーは、それらが提供する予定だった IP アドレスの予約を取り消すことができます。
4. 選択されたサーバーは、クライアントの IP アドレスを割り当て、その情報を DHCP データストアに格納し、クライアントに承認 (ACK) を送信します。承認メッセージには、クライアントのためのネットワーク構成パラメータが含まれています。クライアントは、その IP アドレスが他のシステムに使用されていないこと

を ping コマンドを使って確認してから、ブート処理を続けてネットワークに参加します。

5. クライアントはリース期間を監視し、規定のリース期間が経過した場合には、リース期間を延長するために、選択したサーバーに対して新たな要求メッセージを送信します。
6. リース期間が、管理者が規定したローカルリースポリシーに合っている場合、要求を受け取る DHCP サーバーは、そのリース期間を延長します。サーバーが 20 秒以内に応答しない場合、クライアントは、他の DHCP サーバーのいずれかがリース期間を延長できるように要求をブロードキャストします。
7. クライアントは、その IP アドレスが必要なくなると、IP アドレスを解放することをサーバーに通知します。この処理は、通常のシャットダウンの際に実行され、また手動で実行することも可能です。

Solaris DHCP サーバー

Solaris DHCP サーバーは、ホストシステム上の Solaris オペレーティング環境ではデーモンとして動作します。Solaris DHCP サーバーは、2 つの基本機能を備えています。

- IP アドレスの管理 – Solaris DHCP サーバーは、IP アドレスの範囲を制御し、常時または定義した期間、IP アドレスをクライアントに割り当てます。DHCP サーバーはリースメカニズムを使って、クライアントが一時的なアドレスを使用できる期間を決めます。アドレスは、不要になるとプールに返され、再割り当てされます。DHCP サーバーは、DHCP ネットワークテーブル内にクライアントへの IP アドレス結合情報を保持し、複数のクライアントが同じアドレスを使用しないようにします。
- クライアントにネットワーク構成情報を提供 – Solaris DHCP サーバーは、クライアントの IP アドレスを割り当て、ホスト名やブロードキャストアドレス、ネットワークサブネットマスク、デフォルトゲートウェイ、ネームサービスといったネットワーク構成情報をクライアントに提供します。ネットワーク構成情報は、サーバーの dhcptab データベースから取得されます。

また、Solaris DHCP サーバーは以下の追加機能を実行するように構成することも可能です。

- BOOTP クライアント要求への応答 – Solaris DHCP サーバーは、BOOTP サーバーを検索する BOOTP クライアントからのブロードキャストを待機し、BOOTP クライアントに IP アドレスとブートパラメータを提供します。管理者は、これらの情報をあらかじめ静的に構成しておく必要があります。DHCP サーバーは、BOOTP サーバーとしても DHCP サーバーとしても機能することができます。
- 要求のリレー – Solaris DHCP サーバーは、他のサブネット上の適切なサーバーに BOOTP 要求と DHCP 要求をリレーします。DHCP サーバーは BOOTP リレーエージェントとして構成された場合、DHCP サービスや BOOTP サービスを提供できなくなります。

- DHCP クライアントにネットワークブート情報を提供 – Solaris DHCP サーバーは、DHCP クライアントがネットワーク経由でブートするために必要な情報を DHCP クライアントに提供できます。この情報には、IP アドレスやブートパラメータ、ネットワーク構成情報などがあります。
- ホスト名を指定したクライアントに代わって DNS テーブルを更新 – DHCP サービスを求めるクライアントの要求に Hostname オプションと値が含まれている場合には、DHCP サーバーが、クライアントに代わって DNS を更新することができます。

DHCP サーバーの管理

スーパーユーザーは、DHCP マネージャや 126 ページの「DHCP コマンド行ユーティリティ」 に記載されているコマンド行ユーティリティを使って、DHCP サーバーの起動、終了、構成を行うことができます。通常、DHCP サーバーは、システムのブート時に自動的に起動され、システムのシャットダウン時に自動的に終了するように構成されています。したがって、通常は、サーバーの起動や終了を手動で行う必要はありません。

DHCP データストア

Solaris DHCP サーバーが使用するすべてのデータは、データストアと呼ばれるプレーンテキストファイル、NIS+ テーブル、バイナリ形式ファイルに格納されます。管理者は、DHCP サービスを構成するときに、どの形式のデータストアを使用するかを選択します。データストアの形式の違いについては、145 ページの「データストアの選択」を参照してください。データストアのフォーマットは、DHCP マネージャまたは `dhcpconfig` コマンドを使って変換できます。

さらに、個々のサーバーで異なるデータストアフォーマットを使用している場合でも、それぞれのデータストアで動作するエクスポートユーティリティやインポートユーティリティを使用すれば、DHCP サーバーのデータストアにあるデータを別のデータストアに移動することができます。 DHCP マネージャや `dhcpconfig` コマンドを使用して、データストアの内容全体またはその一部をエクスポートまたはインポートすることができます。

注 – Solaris DHCP (サーバーツールと管理ツール) とデータベース間のインタフェースになる独自のコードモジュールを開発する場合には、DHCP データ領域のデータベースやファイルのフォーマットはどのようなものでもかまいません。詳細は、『*Solaris DHCP* サービス開発ガイド』を参照してください。

Solaris DHCP データストアには、次の 2 種類のテーブルがあります。このテーブルの内容を表示、管理するには、DHCP マネージャまたはコマンド行ユーティリティを使用します。

- dhcptab テーブル – クライアントに提供することが可能な構成情報が入っています。
- DHCP ネットワークテーブル – テーブル名が示すネットワーク上にある DHCP クライアントや BOOTP クライアントの情報が入っています。たとえば、ネットワーク 134.20.0.0 のテーブル名には 134_20_0_0 が含まれています。

dhcptab テーブル

dhcptab テーブルには、クライアントが DHCP サーバーから取得できるすべてのデータが入っています。DHCP サーバーは、起動されるたびに dhcptab テーブルをスキャンします。dhcptab のファイル名は、使用されるデータストアによって異なります。たとえば、NIS+ データストア SUNWnisplus によって作成された dhcptab は SUNWnisplus1_dhcptab になります。

DHCP プロトコルは、クライアントに渡すことができる情報の標準的な項目を多数定義しています。これらの項目は、パラメータ、シンボル、またはオプションと呼ばれます。DHCP プロトコルでは、オプションは数値コードとテキストラベルで定義されており、値は与えられていません。例として、一般的に使用される標準オプションの一部を示します。

表 **7–1** DHCP 標準オプションの例

コード	ラベル	説明
1	Subnet	サブネットマスク IP アドレス
3	Router	ルーターの IP アドレス
6	DNSserv	DNS サーバーの IP アドレス
12	Hostname	クライアントホスト名を表すテキスト文字列
15	DNSdmain	DNS ドメイン名

オプションの中には、管理者がサーバーの構成中に情報を提供すると、自動的に値が割り当てられるものがあります。また、管理者は後で、他のオプションに値を明示的に割り当てることもできます。オプションとその値はクライアントに渡され、構成情報を形成します。たとえば、オプションと値のペアである DNSdmain=Georgia.Peach.COM は、クライアントの DNS ドメイン名を Georgia.Peach.COM に設定します。

オプションは、マクロとして知られているコンテナ内で他のオプションと共にグループ化することができ、これによりクライアントへ容易に情報を渡すことができます。マクロの中には、サーバー構成時に自動的に作成され、構成時に値が割り当てられるオプションを含むものがあります。また、マクロには他のマクロを含めることもできます。

dhcptab ファイルのフォーマットについては、dhcptab(4) のマニュアルページを参照してください。 DHCP マネージャでは、「オプション (Options)」タブや「マクロ (Macros)」タブに示されるすべての情報は dhcptab ファイルから得られます。オプションについては 128 ページの「オプションについて」を、マクロについては 129 ページの「マクロについて」 をそれぞれ参照してください。

dhcptab テーブルをテキストエディタで編集しないでください。オプションやマクロの作成、削除、変更には、dhtadm コマンドまたは DHCP マネージャを使用する必要があります。

DHCP ネットワークテーブル

DHCP ネットワークテーブルは、クライアントの識別子を IP アドレスと、各アドレスに関連した構成パラメータに対応付けます。ネットワークテーブルのフォーマットについては、dhcp_network(4) のマニュアルページを参照してください。 DHCP マネージャでは、「アドレス (Addresses)」タブに示されるすべての情報はネットワークテーブルから得られます。

DHCP マネージャ

DHCP マネージャは、DHCP サービスに関連するすべての管理作業を行うためのグラフィカルツールです。DHCP マネージャを使用するには、スーパーユーザーになる必要があります。このツールを使用すると、サーバーだけでなく、サーバーが使用するデータも管理することができます。サーバー上では DHCP マネージャを下記の場合に使用することができます。

- DHCP サーバーを構成および構成解除する場合
- DHCP サーバーを起動、停止、および再起動する場合
- DHCP サービスを有効または無効にする場合
- サーバーの設定をカスタマイズする場合

さらに、DHCP マネージャでは、IP アドレスやネットワーク構成マクロ、ネットワーク構成オプションに関して次のことができます。

- DHCP 管理下にあるネットワークの追加や削除
- DHCP 管理下にある IP アドレスの表示、追加、変更、削除、解放
- ネットワーク構成マクロの表示、追加、変更、削除
- 標準以外のネットワーク構成オプションの表示、追加、変更、削除

DHCP マネージャでは、DHCP データストアに関して次のことができます。

- データを新しいデータストアフォーマットに変換する。
- DHCP データをある DHCP サーバーから別のサーバーに移動する。 データを最初のサーバーからエクスポートし、次のサーバーにインポートする必要があります。

DHCP マネージャでは、実行できる手順についての詳細なオンラインヘルプも利用できます。

DHCP コマンド行ユーティリティ

すべての DHCP 管理機能は、コマンド行ユーティリティを使用しても実行することができます。コマンド行ユーティリティを実行するには、スーパーユーザーとして、または DHCP 管理プロファイルに割り当てられているユーザーでログインしている必要があります。これについては、171 ページの「DHCP コマンドへのユーザーアクセスの設定」 を参照してください。

次の表に、各ユーティリティとその使用目的を示します。

表 **7–2** DHCP コマンド行ユーティリティ

コマンド名	説明と使用目的
`in.dhcpd`	DHCP サービスデーモン。数個のランタイムオプションの設定を可能にするコマンド行引数を提供する。
`dhcpconfig`	DHCP サーバーの構成や構成解除に使用する。 このユーティリティでは、DHCP マネージャの多くの機能をコマンド行から実行することができる。このユーティリティは主に、一部の構成機能を自動化したいときにスクリプト中で使用する。 `dhcpconfig` は、サーバーシステムのネットワークトポロジファイルから情報を収集し、初期構成に必要な情報を作成する。
`dhtadm`	DHCP クライアント用の構成オプションとマクロの追加、削除、変更に使用する。このユーティリティによって `dhcptab` が間接的に編集され、`dhcptab` のフォーマットが正しく保たれる。 `dhcptab` ファイルを直接編集してはならない。
`pntadm`	DHCP ネットワークテーブルの管理に使用する。このユーティリティでは、IP アドレスやネットワークを DHCP 管理下に追加したり、そこから削除したり、指定する IP アドレスのネットワーク構成を変更したり、DHCP 管理下にある IP アドレスやネットワークの情報を表示したりできる。

役割によるアクセス制御 (RBAC) - DHCP コマンドを使用する場合

`dhcpconfig`、`dhtadm`、`pntadm` コマンドのセキュリティは、役割によるアクセス制御 (RBAC, Role-Based Access Control) の設定値に基づいて決められます。デフォルトでは、これらのコマンドを実行できるのはスーパーユーザーだけです。 これらのコマンドを別のユーザー名で使用したい場合は、この名前を DHCP プロファイルに割り当てる必要があります。これについては、 171 ページの「DHCP コマンドへのユーザーアクセスの設定」を参照してください。

DHCP サーバーの構成

DHCP サーバーを動作させたいシステム上で DHCP マネージャを初めて実行するときは、DHCP サーバーを構成します。DHCP マネージャのサーバー構成ダイアログに、1 つのネットワーク上で DHCP サーバーを使用可能にして実行するために必要な基本情報を入力するように要求するメッセージが表示されます。既存のシステムファイルからいくつかのデフォルト値を取得することができます。そのネットワークに対してシステムを構成していない場合には、デフォルト値はありません。DHCP マネージャは下記の情報を入力するように促します。

- そのサーバーの役割 : DHCP サーバーまたは BOOTP リレーエージェントのいずれか
- データストアの形式 (ファイル、バイナリファイル、NIS+、または独自のもの)
- データストアの構成パラメータ (選択したデータストアの形式によって異なる)
- ホストレコードの更新に使用するネームサービス (使用する場合) (`/etc/inet/hosts`、NIS+、または DNS)
- リース期間と、クライアントがリース期間を更新できるようにするかどうか
- DNS サーバーの DNS ドメイン名および IP アドレス
- DHCP サービス用に構成する最初のネットワークのネットワークアドレスとサブネットマスク
- ネットワークのタイプ : LAN または PPP (ポイントツーポイント)
- ルーターの検索、または特定のルーターの IP アドレス
- NIS サーバーの NIS ドメイン名および IP アドレス
- NIS+ サーバーの NIS+ ドメイン名および IP アドレス

DHCP サーバーは `dhcpconfig` コマンドを使用しても構成することができます。このユーティリティは既存のシステムファイルから自動的に情報を収集し、有用な初期構成を提供します。そのため、`dhcpconfig` コマンドを実行する前に既存のシステムファイルが正しいことを確認しておく必要があります。`dhcpconfig` がどのファイルから情報を入手するかについては、`dhcpconfig(1M)` のマニュアルページを参照してください。

IP アドレスの割り当て

Solaris DHCP サーバーは、下記のタイプの IP アドレス割り当て機能をサポートしています。

- 手動割り当て – DHCP サーバーは、特定の DHCP クライアントに対して管理者が選択した、専用の IP アドレスを割り当てます。このアドレスは変更したり他のクライアントに割り当てたりすることはできません。
- 自動または常時割り当て – DHCP サーバーは有効期限のない IP アドレスを割り当て、管理者がその割り当てを変更するか、あるいは、クライアントがそのアドレスを解放するまで、そのアドレスを永続的にそのクライアントに使用します。

- 動的割り当て – DHCP サーバーは IP アドレスを要求しているクライアントに、一定期間このアドレスをリースします (貸し出します)。この期間が過ぎると、サーバーはこのアドレスを回収し、他のクライアントに割り当てることができます。このアドレスの使用期間はサーバーに構成されているリース期間によって決まります。

ネットワーク構成情報

管理者は、どのような情報を DHCP クライアントに提供するかを決める必要があります。DHCP サーバーを構成するときにはネットワークの基本的な情報を指定しますが、後で、クライアントに提供したい情報を追加することもできます。

DHCP サーバーは、オプションと値の対、およびマクロの形で、`dhcptab` データベースにネットワーク構成情報を保存します。オプションはクライアントに供給するネットワークデータのキーワードです。値はオプションに割り当てられ、DHCP メッセージでクライアントに渡されます。たとえば、NIS サーバーのアドレスは、DHCP サーバーによって割り当てられた値 (IP アドレスのリスト) を持つ `NISservs` というオプションを使用して渡されます。マクロは、クライアントに供給したい任意の個数のオプションをグループ化するための便利な方法です。管理者は、DHCP マネージャを使って、オプションをグループ化するマクロを作成し、それらのオプションに値を割り当てることができます。グラフィカルユーザーインタフェースでないツールを使用する場合は、DHCP 構成テーブル管理用ユーティリティ `dhtadm` を使ってオプションやマクロを処理することができます。

オプションについて

Solaris DHCP では、オプションとはクライアントに渡されるネットワーク情報です。DHCP の資料では、オプションはシンボルやタグとも呼ばれる場合もあります。オプションは、数値コードやテキストラベルで定義されます。オプションには、それが DHCP サービスで使用されるときに値を受け取ります。

DHCP プロトコルは、一般的に指定されているネットワークデータに対して多数の標準オプションを定義しています。 それらオプションにはたとえば、`Subnet`、`Router`、`Broadcast`、`NIS+dom`、`Hostname`、および `LeaseTim` があります。その他の標準オプションについては、`dhcp_inittab` マニュアルページを参照してください。マクロにこれらのオプションを指定する際には、標準オプションのキーワードを変更することはできませんが、ネットワークに関連するオプションに値を割り当てることができます。

標準オプションで指定できないデータに対しては、新しいオプションを作ることができます。作成するオプションは下記のいずれかのカテゴリに分類されるものでなければなりません。

- 拡張 – この DHCP サーバーにはまだ実装されていないが、標準 DHCP オプションとしてすでに予約されています。使用したい標準オプションがわかっているが、DHCP サーバーをグレードアップしたくない場合に使用することができます。

- サイト – 使用しているサイトに固有なオプションのために予約されています。システム管理者がこれらのオプションを作成します。
- ベンダー – ハードウェアまたはベンダープラットホームなどの特定クラスのクライアントにだけ適用するオプションのために予約されています。Solaris DHCP の実装には、Solaris クライアント用の多数のベンダーオプションが含まれています。たとえば、オプション `SrootIP4` は、ネットワークからブートされるクライアントがそのルートファイルシステムとして使用すべきサーバーの IP アドレスを指定します。

第 10 章に、オプションを作成、変更、および削除する手順が説明されています。

マクロについて

Solaris DHCP サービスでは、マクロはネットワーク構成オプション、およびシステム管理者がこれらのオプションに割り当てた値の集まりです。マクロは、オプションをグループ化し、特定のクライアントまたはクライアントタイプにオプションをまとめて渡すために作成します。たとえば、特定のサブネット上のすべてのクライアントを対象としたマクロには、サブネットマスク、ルーター IP アドレス、ブロードキャストアドレス、NIS+ ドメイン、およびリース期間のためのオプションと値のペアを含めることができます。

DHCP サーバーによるマクロ処理

DHCP サーバーがマクロを処理するときは、そのマクロに定義されているネットワークオプションと値を、クライアントへの DHCP メッセージに含めます。サーバーは、特定のタイプのクライアントに対し一部のマクロを自動的に処理します。

マクロを自動的に処理するためには、マクロの名前が、次の表に示すカテゴリのいずれかに従っている必要があります。

表 7–3 自動処理のためのマクロのカテゴリ

マクロのカテゴリ	説明
クライアントクラス	このマクロ名は、クライアントマシンのタイプやオペレーティングシステムによって指定されたクライアントの種類と一致する。たとえば、`SUNW.Ultra-1` という名前のマクロがサーバーに存在する場合、ハードウェア実装が `SUNW,Ultra-1` であるクライアントは、自動的に `SUNW.Ultra-1` マクロの値を受け取る。
ネットワークアドレス	このマクロ名は、DHCP が管理するネットワーク IP アドレスと一致する。たとえば、サーバーのマクロの名前が `10.53.224.0` の場合、10.53.224.0 ネットワークに接続されているクライアントはいずれも自動的に `10.53.224.0` マクロ内の値を受け取る。

表 **7–3** 自動処理のためのマクロのカテゴリ　(続き)

マクロのカテゴリ	説明
クライアント ID	このマクロ名は、通常は Ethernet または MAC アドレスから導出された、クライアント用の一意の識別子と一致する。たとえば、`08002011DF32` という名前のマクロがサーバーに存在する場合、(Ethernet アドレス `8:0:20:11:DF:32` から得られる) クライアント ID `08002011DF32` を持つクライアントは、`08002011DF32` という名前のマクロにある値を自動的に受け取る。

表 7–3 に示されているカテゴリのいずれも使用しない名前を持つマクロは、下記のいずれかの条件が満たされた場合にのみ処理することができます。

- マクロが IP アドレスに割り当てられる場合
- マクロが、自動的に処理される他のマクロに含まれる場合
- マクロが、IP アドレスに割り当てられている他のマクロに含まれている場合

注 **–** サーバーを構成する場合、デフォルトでは、そのサーバーの名前と一致する名前の付いたマクロが作られます。このサーバーマクロは、自動処理が行われる名称タイプのいずれとも一致しないため、いずれのクライアントに対しても自動的に処理されません。後でサーバー上で IP アドレスを作成する場合、その IP アドレスは、サーバーのデフォルトのマクロを使用するように割り当てられます。

マクロ処理の順序

DHCP クライアントが DHCP サービスを要求するときは、DHCP サーバーはどのマクロがそのクライアントに一致するかを決定します。このサーバーは、処理の順序を決めるためのマクロのカテゴリを使用して、より一般的なものから特定のものへと、順にマクロを処理します。マクロは下記の順序で処理されます。

1. クライアントクラスマクロ – 最も一般的なカテゴリ
2. ネットワークアドレスマクロ – クライアントクラスよりは特定なマクロ
3. IP アドレスに割り当てられたマクロ – ネットワークアドレスよりは特定されたマクロ
4. クライアント ID マクロ – 1 クライアントだけに適用される最も特定されたカテゴリ

他のマクロに含まれているマクロはそのマクロの一部として処理されます。

複数のマクロに同じオプションが含まれている場合は、最も特定されたカテゴリのマクロ内のオプションに設定されている値が一番最後に処理されるため、その値が使用されます。たとえば、ネットワークアドレスに、24 時間の値を持つリース期間オプションが入っていて、クライアント ID マクロに 8 時間の値を持つリース期間オプションが入っている場合は、そのクライアントは 8 時間のリース期間を受け取ります。

Solaris DHCP クライアント

「クライアント」という用語は、ネットワーク上でクライアントとしての役割を実行している物理的なマシンについて言及するために使用される場合がありますが、ここで説明している DHCP クライアントはソフトウェアエンティティです。Solaris DHCP クライアントは、そのネットワーク構成を DHCP サーバーから受け取るように構成されているシステムの Solaris オペレーティング環境で動作するデーモン (`dhcpagent`) です。他のベンダーの DHCP クライアントも Solaris DHCP サーバーのサービスを使用することができます。ただし、この節では Solaris DHCP クライアントについてのみ説明します。

この節の説明では 1 つのネットワークインタフェースを想定していることに注意してください。137 ページの「複数のネットワークインタフェースを備えた DHCP クライアントシステム」 では、2 つ以上のネットワークインタフェースを備えたホストの重要な問題について説明しています。

DHCP クライアントのインストール

Solaris DHCP クライアントは、Solaris オペレーティング環境のインストール時に、DHCP を使用してネットワークインタフェースを構成するように指定すると、システム上にインストールされ使用可能な状態になります。DHCP を使用するために Solaris クライアントに対して必要な作業はこれだけです。

Solaris オペレーティング環境がすでに動作しているシステムで DHCP を使ってネットワーク構成情報を取得したい場合には、164 ページの「Solaris DHCP クライアントの構成と構成解除」 を参照してください。

DHCP クライアントの起動

`dhcpagent` デーモンは、システムのブートに関与する他のプロセスに必要な構成情報を取得します。 そのため、システム起動スクリプトは、ブートプロセスの初期段階に `dhcpagent` を起動し、DHCP サーバーからネットワーク構成情報が到着するのを待ちます。

`/etc/dhcp.`*interface* ファイル (たとえば、Sun Enterprise Ultra™ システム上の `/etc/dhcp.hme0`) が存在していれば、起動スクリプトは、指定されたインタフェース上で DHCP が使用されることを認識します。`dhcp.`*interface* ファイルを検出すると、起動スクリプトは `dhcpagent` デーモンを起動します。

起動された `dhcpagent` は、ネットワークインタフェースの構成を行う指示を受信するまで待機します。起動スクリプトは `ifconfig` *interface* `dhcp start` コマンドを出して、`dhcpagent` に DHCP を起動するように指示します (119 ページの「DHCP の

動作」 を参照)。 dhcp.*interface* ファイル内にコマンドが含まれている場合、それらのコマンドは ifconfig の dhcp start オプションに追加されます。dhcp オプションと共に使用されるオプションについては、ifconfig(1M) のマニュアルページを参照してください。

Solaris DHCP クライアントはネットワーク構成情報をどのように管理するか

DHCP サーバーから情報パケットが取得されると、dhcpagent はネットワークインタフェースの構成、立ち上げを行い、そのインタフェースを IP アドレスのリース期間中制御します。dhcpagent デーモンは、メモリーに保持された内部テーブル中に構成データを保持します。システム起動スクリプトは dhcpinfo コマンドを使用して dhcpagent デーモンのテーブルから構成オプションの値を抽出します。それらの値は、システムを構成し、システムがネットワークに加わることができるようにするために使用されます。

エージェントは、一定時間 (通常はリース期間の半分) が過ぎるまで何もせずに待機した後でリースの延長を DHCP サーバーに要求します。dhcpagent デーモンは、インタフェースが停止していたり、IP アドレスが変更されているのを検出すると、ifconfig から指示があるまでそのインタフェースの制御を行いません。また、dhcpagent デーモンは、インタフェースが適切に動作し、IP アドレスが変更されていないことを検出すると、リースの更新要求をサーバーに送信します。リースを更新できない場合、dhcpagent デーモンはリース期間の満了時にそのインタフェースを停止します。

DHCP のクライアントの管理

通常のシステム動作時には、Solaris DHCP クライアントの管理は必要ありません。Solaris DHCP クライアントはシステムブート時に自動的に起動し、リースについてサーバーとネゴシエーションを行い、シャットダウン時に停止します。dhcpagent デーモンを手動で起動または停止することはできません。ただし、必要な場合は、クライアントシステムのスーパーユーザーとして ifconfig コマンドを使用、クライアントにおけるネットワークインタフェースの管理に関与することができます。

DHCP クライアントで使用する ifconfig コマンドオプション

ifconfig コマンドでは、次の処理を行うことができます。

- DHCP クライアントの起動 – ifconfig *interface* dhcp start コマンドでは、DHCP クライアントと DHCP サーバー間の対話を開始して、IP アドレスと新しい構成オプション群を取得します。 このコマンドは、IP アドレスを追加したり、サブネットマスクを変更する場合など、情報を変更してそれをクライアントですぐに

使用したいときに便利です。

- ネットワーク構成情報だけの要求 – `ifconfig` *interface* `dhcp inform` コマンドは、`dhcpagent` が IP アドレスを除くネットワーク構成パラメータを要求するようにします。このコマンドは、ネットワークインタフェースが有効な IP アドレスを持っているが、クライアントシステムが更新されたネットワークオプションを必要としているような場合に便利です。たとえば、DHCP を IP アドレスの管理には使用しないが、ネットワーク上のホストの構成には使用したいような場合です。
- リースの延長要求 – `ifconfig` *interface* `dhcp extend` コマンドは、`dhcpagent` がリース期間の延長を要求するようにします。この処理は自動的に実行されますが、リース期間を変更し、新しいリース期間を次のリース更新を待たずにクライアントでただちに使用したい場合は、手動でこのコマンドを実行できます。
- **IP** アドレスの解放 – `ifconfig` *interface* `dhcp release` コマンドは、`dhcpagent` がネットワークインタフェースで使用されている IP アドレスを解放するようにします。この処理はリース満了時に自動的に実行されます。長いリース期間が設定されている場合でネットワークインタフェースを長期間停止したり、ネットワークからシステムを切り離す場合には、このコマンドを使用します。
- IP アドレスの放棄 – `ifconfig` *interface* `dhcp drop` コマンドは、`dhcpagent` が DCHP サーバーへ通知せずに、ネットワークインタフェースを放棄するようにします。この処理により、クライアントは次回リブート時に同じ IP アドレスを使用することができます。
- ネットワークインタフェースに対する `ping` の実行 – `ifconfig` *interface* `dhcp ping` コマンドは、インタフェースが DHCP の制御下にあるかどうかをテストします。
- ネットワークインタフェースの DHCP 構成状態の表示 – `ifconfig` *interface* `dhcp status` コマンドは、DHCP クライアントの現在の状態を表示します。この表示には、次の情報が含まれています。
  - クライアントに IP アドレスがバインドされているかどうか
  - 送信、受信、および拒否された要求の数
  - 一次インタフェースかどうか
  - リースが取得された時刻、リースが期限切れになった時刻、リースの更新予定時刻と実際に更新された時刻 入力例

```
# ifconfig hme0 dhcp status
Interface  State         Sent  Recv  Declined  Flags
hme0       BOUND           1     1         0   [PRIMARY]
(Began,Expires,Renew)=
(08/16/2000 15:27, 08/18/2000 13:31, 08/17/2000 15:24)
```

DHCP クライアント用のパラメータファイル

クライアントシステム上の `/etc/default/dhcpagent` ファイルには、`dhcpagent` デーモンに対する調整可能なパラメータが含まれています。テキストエディタを使用して、クライアントの動作に影響を与えるパラメータを変更することができます。このファイルには詳しい説明が記載されているので、 `dhcpagent` のマニュアルページと併せて、このファイルも参照してください。

DHCP クライアントのシャットダウン

DHCP を実行しているシステムが正常にシャットダウンするときは、`dhcpagent` デーモンが現在の構成情報を `/etc/dhcp/`*interface*`.dhc` ファイルに書き込みます。この場合、リースは解放されるのではなく放棄されるので、DHCP サーバーは、IP アドレスが実際には使用されていないことを認識できません。

システムのリブート時にリースがまだ有効であると、リブート前に使用していたものと同じ IP アドレスとネットワーク構成情報を使用するために、DHCP クライアントは簡略化された要求を送信します。DHCP サーバーがこれを許可した場合、クライアントはシステムのシャットダウン時にディスクに書き込んだ情報を使用することができます。サーバーがこの情報の使用をクライアントに許可しない場合は、クライアントは前述の DHCP プロトコルシーケンスを開始し、新しいネットワーク構成情報を取得します。

DHCP クライアントシステムとネームサービス

Solaris システムでは、DNS、NIS、NIS+、およびローカルファイル (`/etc/inet/hosts`) のネームサービスがサポートされています。これらのネームサービスを使用するためには、ある程度の事前構成が必要です。 使用するネームサービスを指定するために、ネームサービススイッチ構成ファイル (`nsswitch.conf`(4) を参照) を正しく設定する必要もあります。

ネームサービスのクライアントとしてシステムを構成しないと、DHCP クライアントシステムでネームサービスを使用することはできません。

次の表は、DHCP に関連する問題をネームサービスごとに要約したものです。この表には、各ネームサービスに対してクライアントを設定する上で役立つマニュアルへのリンクが示されています。

表 **7–4** DHCP クライアントシステムに対するネームサービスクライアント設定情報

ネームサービス	クライアントの設定に関する注意

NIS	Solaris DHCP を使ってクライアントシステム上に Solaris オペレーティング環境をインストールする場合には、構成マクロに `NISservs` オプションと `NISdmain` オプションを指定すれば、NIS サーバーの IP アドレスと NIS ドメイン名をクライアントに渡すことができます。これによって、クライアントは自動的に NIS クライアントになります。
	
	DHCP クライアントシステムで Solaris オペレーティング環境がすでに動作している場合、DHCP サーバーが NIS 情報をクライアントに送信しても、クライアントシステムが自動的に NIS クライアントとして構成されるわけではありません。
	
	DHCP クライアントシステムに NIS 情報を送信するように DHCP サーバーが構成されている場合には、クライアントで次の `dhcpinfo` コマンドを実行すれば、これらの値を見ることができます。
	
	`# /sbin/dhcpinfo NISdmain`
	
	`# /sbin/dhcpinfo NISservs`
	
	NIS ドメイン名と NIS サーバーの値は、システムを NIS クライアントとして構成するときに使用します。
	
	Solaris DHCP クライアントシステムでは NIS クライアントを標準の方法で設定します。これについては、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「NIS サービスの設定と構成」を参照してください。
	
	注 – スクリプトを作成すれば、`dhcpinfo` や `ypinit` を使って、DHCP クライアントシステムにおける NIS クライアントの構成を自動的に行うことができます。

NIS+	予約されていない IP アドレス (アドレスは常に同じであるとは限らない) を DHCP クライアントシステムが受け取る場合には、非標準的な方法で DHCP クライアントシステムを NIS+ クライアントとして設定する必要があります。これについては、238 ページの「NIS+ クライアントとしての DHCP クライアントの設定」を参照してください。この手順が必要な理由は、NIS+ ではサービス要求を認証するためのセキュリティ手段が使用されるためです。セキュリティ手段は IP アドレスによって異なります。 DHCP クライアントシステムが手動で IP アドレスを割り当てられている場合は (クライアントのアドレスは常に同じ)、非標準的な方法で NIS+ クライアントを設定できます。これについては『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)* 』の「NIS+ クライアントマシンの設定」を参照してください。

表 **7–4** DHCP クライアントシステムに対するネームサービスクライアント設定情報　(続き)

ネームサービス	クライアントの設定に関する注意
/etc/inet/hosts	ネームサービスとして /etc/inet/hosts を使用する DHCP クライアントシステムには、/etc/inet/hosts ファイルを設定する必要があります。 DHCP クライアントシステム自身の /etc/inet/hosts ファイルには、そのホスト名が DHCP ツールによって追加されます。 ただし、同じネットワークにある他のシステムの /etc/inet/hosts ファイルには、このホスト名を手動で追加する必要があります。さらに、DHCP サーバーシステムが名前を解決するために /etc/inet/hosts を使用する場合は、このシステムにもクライアントのホスト名を手動で追加する必要があります。
DNS	DHCP クライアントシステムが DNS ドメイン名を DHCP から取得する場合には、クライアントシステムの /etc/resolv.conf ファイルは自動的に構成されます。/etc/inet/hosts ファイルを使用するシステムで DNS を実際に使用するためには、/etc/nsswitch.conf ファイルの hosts 行に dns を追加する必要があります。DNS クライアントの詳細については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』の「DNS クライアントの設定」を参照してください。 クライアントシステムでローカル名の解決に NIS または NIS+ を使用する場合は、次の点に注意してください。 ■ NIS – NIS サーバーで DNS 転送が可能な場合は (デフォルトでは可能)、NIS クライアントシステムでも DNS を使用できます。 この場合には、DNS クライアントとしての設定は必要ありません。NIS サーバーで DNS 転送が可能でない場合は、DNS クライアントになると、クライアントシステムで DNS を使用することができます。DNS クライアントになる方法については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)* 』の「クライアントの追加と削除」を参照してください。クライアントが DNS ドメイン名を DHCP サーバーから取得する場合には、DNS クライアントに必要な /etc/resolv.conf ファイルは自動的に構成されます。したがって、この場合には、nsswitch.conf ファイルの構成だけが必要です。 ■ NIS+ – nsswitch.conf ファイルを編集して hosts 行に dns を追加すれば、 DNS を使用するように NIS+ クライアントシステムを構成することができます。

クライアントホスト名の登録

DHCP サービスで使用する IP アドレスのホスト名を DHCP サーバーが生成するようにすると、DHCP サーバーがこれらのホスト名を NIS+、/etc/inet/hosts、または DNS ネームサービスに登録できます。 ホスト名の登録を NIS で行うことはできません。NIS には、NIS マップの更新や伝達をプログラムで行うためのプロトコルが備わっていないからです。

注 – DNS サーバーと DHCP サーバーが同じシステムで動作している場合のみ、DHCP サーバーは、生成したホスト名を DNS に登録することができます。

DHCP クライアントがそのホスト名を指定し、DHCP サーバーが動的な更新をすることができるように DNS サーバーが構成されている場合には、DNS サーバーと DHCP サーバーが異なるシステムで動作していても、DHCP サーバーがクライアントに代わって DNS を更新することができます。この機能の使用方法については、179 ページの「DHCP サーバーによる動的 DNS 更新の有効化」 を参照してください。

次の表は、DHCP クライアントシステムのホスト名の登録についてネームサービスごとに示したものです。

表 **7–5** ネームサービスへのクライアントホスト名の登録

	ホスト名を登録する人または物	
ネームサービス	**DHCP** が生成したホスト名	**DHCP** クライアントが指定したホスト名
NIS	NIS 管理者	NIS 管理者
NIS+	DHCP ツール	DHCP ツール
`/etc/inet/hosts`	DHCP ツール	DHCP ツール
DNS	DHCP ツール (DNS サーバーが DHCP サーバーと同じシステムで動作している場合)	DHCP サーバー (動的 DNS 更新が可能として構成されている場合)
	DNS 管理者 (DNS サーバーが異なるシステムで動作している場合)	DNS 管理者 (DHCP サーバーがそのように構成されていない場合)

Solaris DHCP クライアントは、DHCP 要求に特定のホスト名を指定できます。ただし、DHCP クライアントがそのように構成されている必要があります。構成方法については、181 ページの「特定のホスト名に応答するように Solaris クライアントを有効にする方法」を参照してください。Solaris 以外のクライアントでこの機能がサポートされているかどうかについては、それぞれのマニュアルを参照してください。

複数のネットワークインタフェースを備えた DHCP クライアントシステム

DHCP クライアントデーモンは、それぞれが独自の IP アドレスとリース期間を持つ、複数のインタフェースを、1 つのシステム上で同時に管理することができます。DHCP に対して複数のネットワークインタフェースが構成されている場合、クライアントはそれらのインタフェースを構成するために個別の要求を発行し、各インタ

フェースに対して個別のネットワーク構成オプションのセットを維持します。この場合、パラメータは個別に保存されますが、パラメータの中にはその性質上広域的なものがあります。そのようなパラメータは、特定のネットワークインタフェースではなく、システム全体に適用されます。

たとえば、ホスト名、NIS ドメイン名、時間帯などのオプションは広域パラメータであり、各インタフェースに対して同じ値を取ります。ただし、DHCP 管理者が入力した情報に誤りがあるために、これらの値が異なっている場合があります。広域パラメータの問い合わせに対して応答が 1 つだけ返されるようにするために、一次ネットワークインタフェース用のパラメータだけが要求されます。一次インタフェースとして取り扱いたいインタフェースには、`/etc/dhcp`*interface* ファイルに `primary` という語を挿入することができます。

第 8 章

DHCP サービスの使用計画 (手順)

DHCP サービスは、既存のネットワークで使用することも、これから構築するネットワークで使用することもできます。ネットワークを構築中である場合は、DHCP サービスの設定を行う前に第 3 章を参照してください。既存のネットワークを使用する場合は、そのままこの章をお読みください。

この章では、ネットワークに DHCP サービスを設定する前に行うべき作業について説明します。この章の説明は、DHCP マネージャを使用することを前提にしていますが、DHCP サービスの設定はコマンド行ユーティリティ `dhcpconfig` を使って行うこともできます。

この章では、以下の内容について説明します。

DHCP サービスを使用するためのネットワークの準備 (作業マップ)

DHCP の使用に先立ってネットワークを設定する際には、まず情報を収集し、サーバーをどのように構成するかを決める必要があります。 以下の作業マップを使って、DHCP を使用するためにネットワークで準備する必要がある作業を確認してください。

タスク	説明	参照先
ネットワークトポロジをマッピングする	ネットワークで提供するサービスとその場所を決める	140 ページの「ネットワークトポロジのマッピング」
必要な DHCP サーバーの数を決める	予想される DHCP クライアント数を元にして必要な DHCP サーバーの数を決める	141 ページの「DHCP サーバー数の決定」
システムファイルと `netmasks` テーブルを更新する	ネットワークトポロジを正確に反映する	142 ページの「システムファイルとネットマスクテーブルの更新」

ネットワークトポロジのマッピング

ネットワークの物理的な構造またはレイアウトを示すマップをまだ作成していない場合は、それを作成します。このマップには、ルーターやクライアントの場所と、ネットワークサービスを提供するサーバーの場所を明示してください。ネットワークトポロジを示すこのマップは、どのサーバーから DHCP サービスを提供し、どのような構成情報をクライアントに提供するかを決める上で必要です。

ネットワークの計画についての詳細は、 第 3 章を参照してください。

DHCP 構成プロセスは、サーバーのシステムファイルとネットワークファイルから、いくつかのネットワーク情報を検索することができます。142 ページの「システムファイルとネットマスクテーブルの更新」では、これらのファイルについて説明しています。ただし、クライアントに他のサービス情報を提供したい場合には、サーバーのマクロに入力する必要があります。ネットワークトポロジを点検する際には、クライアントが認識する必要があるサーバーの IP アドレスを控えておいてください。次に、ネットワーク上にあるにもかかわらず、DHCP 構成プロセスが検出できないネットワークサービスの例を示します。

- タイムサーバー
- ログサーバー
- プリントサーバー
- インストールサーバー
- ブートサーバー
- スワップサーバー
- X Window フォントサーバー
- TFTP サーバー

避けなければならないネットワークトポロジ

DHCP は、複数の IP ネットワークが、複数のネットワークハードウェアインタフェースや複数の論理インタフェースを介して同じネットワークハードウェア媒体を共有するネットワーク環境では正しく動作しません。同じ物理 LAN で複数の IP

ネットワークが動作していると、DHCP クライアントの要求はすべてのネットワークハードウェアインタフェースに送信されます。そのため、クライアントは、すべての IP ネットワークに同時に接続されているものとみなされます。

DHCP は、適切な IP アドレスをクライアントに割り当てられるように、クライアントのネットワークアドレスを特定できる必要があります。同じハードウェア媒体に複数のネットワークが存在していると、サーバーはクライアントのネットワークを特定できないため、IP アドレスを割り当てることができません。

DHCP はどのネットワーク上でも使用することができますが、複数のネットワーク上では使用できません。この条件がユーザーのニーズと合わない場合は、ネットワークを再構成する必要があります。再構成の方法としては、次のものが考えられます。

- 可変長サブネットマスク (variable length subnet mask: VLSM) を使用して、手持ちの IP アドレス空間を有効活用します。これにより、同じ物理ネットワーク上で複数の LAN を動作させる必要がなくなります。VLSM と Classless Inter-Domain Routing (CIDR) についての詳細は、RFC-1519 を参照してください。
- スイッチ上のポートを構成し、デバイスを別の物理 LAN に割り当てます。これにより、Solaris DHCP の要件である、1 つの LAN から 1 つの IP ネットワークへのマッピングが維持されます。ポートの構成については、スイッチに関する技術資料を参照してください。

DHCP サーバー数の決定

DHCP クライアントをサポートするために必要なサーバーの数は、データストアに何を使用するかによって異なります。次の表は、1 つの DHCP サーバーで DHCP/BOOTP クライアントをいくつまでサポートできるかをデータストア別に示したものです。

表 **8–1** 予想される最大クライアント数

データストア	最大クライアント数
テキストファイル	10,000
NIS+	40,000
バイナリファイル	100,000

この最大数は一般的な指針であり、絶対的な数ではありません。DHCP サーバーのクライアントの能力は、クライアントが 1 秒間にいくつのトランザクションを処理する必要があるかに大きく依存します。一方、サーバーがサポートできるクライアントの数は、クライアントのリース期間と使用パターンで大きく変わります。たとえば、リースが 12 時間に設定され、ほとんどのユーザーが夜にシステムを停止し、朝の同じ時間にシステムを開始するとします。この場合、多数のクライアントが同時にリースを要求するので、サーバーは、毎朝ピークトランザクションを処理できなければな

りません。したがってこのような環境では、DHCP サーバーは、リース期間がこれより長い環境や、ケーブルモデムのように常時接続されているデバイスで構成される環境に比べて、少ないクライアントしかサポートできません。

各データストアについては、145 ページの「データストアの選択」 を参照してください。

システムファイルとネットマスクテーブルの更新

構成処理の間、DHCP マネージャまたは dhcpconfig ユーティリティは、サーバー上のさまざまなシステムファイルを走査し、サーバーの構成に使用できる情報を収集します。

DHCP マネージャや dhcpconfig を使ってサーバーの構成を行う前に、システムファイルの内容が最新の状態になっていることを確認してください。サーバーの構成を行なった後にエラーに気が付いた場合は、DHCP マネージャまたは dhtadm を使って、サーバー上のマクロを修正する必要があります。

次の表は、DHCP サーバーの構成中に収集されるいくつかの情報と、情報の提供元を示します。サーバーで DHCP を構成する前に、これらの情報が適切に設定されていることを確認してください。サーバーの構成後にシステムファイルを変更する場合は、この変更を反映するためにサービスを再構成する必要があります。

表 **8–2** DHCP 構成のための情報

インフォメーション	送信元	コメント
時間帯	システムの日時、時間帯の設定値	日時と時間帯は Solaris のインストール時に初期設定される。日時を変更するには date コマンドを使用し、時間帯を変更するには、/etc/TIMEZONE ファイルの TZ 変数を編集する。
DNS パラメータ	/etc/resolv.conf	DHCP サーバーは、/etc/resolv.conf ファイルから DNSドメイン名や DNS サーバーアドレスなどの DNS パラメータを検索する。 resolv.conf の詳細については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

表 **8–2** DHCP 構成のための情報　(続き)

インフォメーション	送信元	コメント
NIS または NIS+ パラメータ	システムのドメイン名、`nsswitch.conf`、NIS、NIS+	DHCP サーバーは、`domainname` コマンドを使ってサーバーのドメイン名を取得し、`nsswitch.conf` ファイルを使ってドメインベースの情報をどこから検索するかを決める。サーバーが NIS または NIS+ クライアントの場合、DHCP サーバーは NIS または NIS+ サービスを参照し、NIS/NIS+ サーバーの IP アドレスを取得する
デフォルトルーター	システムのルーティングテーブル、管理者による入力	DHCP サーバーはネットワークルーティングテーブルを検索し、ローカルネットワークに接続されているクライアントのデフォルトルーターを見つける。同じネットワーク上にないクライアントについては、管理者にこの情報を入力するように要求する。
サブネットマスク	ネットワークインタフェース、`netmasks` テーブル	DHCP サーバーは、自身のネットワークインタフェースを参照して、ローカルクライアント用のネットマスクとブロードキャストアドレスを特定する。この要求がリレーエージェントからすでに転送されてきている場合には、リレーエージェントのネットワークにある `netmasks` テーブル内のサブネットマスクを参照する
ブロードキャストアドレス	ネットワークインタフェース、`netmasks` テーブル	ローカルネットワークの場合には、DHCP サーバーは、ネットワークインタフェースからブロードキャストアドレスを取得する。リモートネットワークでは、サーバーは BOOTP リレーエージェントの IP アドレスとリモートネットワークのネットマスクを使用して、そのネットワーク用のブロードキャストアドレスを計算する

DHCP サーバーの構成前に必要な選択 (作業マップ)

この節では、ネットワークに最初の DHCP サーバーを構成する前に決定する必要がある事柄について説明します。この作業マップを使って、決定する必要がある事柄を確認してください。

タスク	説明	参照先
DHCP サーバーを選択する	DHCP サービスを実行するためのシステム要件をサーバーが満たしているかどうか判断する	144 ページの「DHCP を使用するためのサーバーの選択」
データストアを選択する	データストアの選択肢を比較して、サイトに最も適したデータストアを決定する	145 ページの「データストアの選択」
リースポリシーを設定する	サイトに適したリースを決定するために、IP アドレスのリースについて確認する	146 ページの「リースポリシーの設定」
ルーターのアドレスを指定するか、ルーターを検索するかを選択する	DHCP クライアントが特定のルーターを使用するか、ルーターを検索するかを決定する	147 ページの「DHCP クライアントのためのルーターの決定」

DHCP を使用するためのサーバーの選択

ネットワークトポロジを念頭に置き、次のガイドラインに従って、DHCP サーバーを設定するホストを選択します。

サーバーとしての要件は次のとおりです。

- Solaris 2.6、Solaris 7、Solaris 8、または Solaris 9 オペレーティング環境が動作している。多数のクライアントをサポートする必要がある場合は、Solaris 8 7/01 オペレーティング環境以降のバージョンをインストールする必要があります。
- DHCP を使用するクライアントがあるすべてのネットワークに、直接ネットワーク経由、または BOOTP リレーエージェントを介してアクセス可能である。
- ルーティングを使用するように構成されている。
- ネットワークトポロジを反映した `netmasks` テーブルが正しく構成されている。

データストアの選択

DHCP データは、テキストファイル、バイナリファイル、または NIS+ ディレクトリサービスに保存できます。次の表は、各データストアの特徴とそれが最も適している環境を示したものです。

表 **8–3** データストアの比較

データストア	性能	保守	共有	推奨環境
バイナリファイル	高性能、大容量	少ない保守、データベースサーバーが不要。内容は、DHCP マネージャ、`dhtadm`、`pntadm` で表示する必要がある。ファイルの定期的なバックアップが必要。	コンテナを DHCP サーバーの間で共有することはできない	多数のネットワークからなり、ネットワークごとに数千のクライアントがいる中規模から大規模の環境。小規模から中規模の ISP に適している
NIS+	中程度の性能と容量。NIS+ サービスの性能と容量に依存する	DHCP サーバーシステムが NIS+ クライアントとして構成されていなければならない。NIS+ サービスの保守が必要。内容は、DHCP マネージャ、`dhtadm`、`pntadm` で表示する必要がある。`nisbackup` による定期的なバックアップが必要。	DHCP データは NIS+ に分散される。複数のサーバーから同じコンテナにアクセスできる	ネットワーク当たり 5000 クライアントまでの小規模から中規模の環境
テキストファイル	中程度の性能、少ない容量	少ない保守、データベースサーバーが不要。ASCII ファイルであるため、DHCP マネージャ、`dhtadm` または `pntadm` を使用しなくても見ることができる。ファイルの定期的なバックアップが必要。	コンテナを DHCP サーバーの間で共有できる。ただし、DHCP データが、NFS マウントポイントを通してエクスポートされる 1 つのファイルシステムに格納されていなければならない	ネットワーク当たり数百から 1000 クライアントで、合計が 10,000 クライアント未満の小規模な環境

NIS+ とは異なり、NIS はデータストアオプションとしては推奨されません。これは、高速な増分更新がサポートされていないためです。ネットワークで NIS が使用されている場合は、データストアとしてテキストファイルまたはバイナリファイルを使用することをお勧めします。

リースポリシーの設定

リースとは、DHCP サーバーが特定の IP アドレスの使用を DHCP クライアントに許可する期間のことです。管理者は、サーバーの初期構成時に、サイト全体に適用するリースポリシーを指定する必要があります。このポリシーには、リース期間やクライアントがこのリースを更新できるかどうかを指定します。 サーバーは提供された情報を使用して、構成時に作成するデフォルトマクロ内のオプションの値を設定します。管理者は、作成する構成マクロでオプションを使用することによって、特定のクライアントや特定のクライアントタイプごとに、異なるリースポリシーを設定することもできます。

リース期間は、リースが有効な時間数、日数、または週数として指定されます。クライアントに IP アドレスが割り当てられると (あるいは、クライアントが、すでに割り当てられている IP アドレスのリースを再度ネゴシエーションすると)、クライアントの DHCP 肯定応答のタイムスタンプにリース期間の時間数が加算され、リース満了日時が計算されます。たとえば、DHCP 肯定応答のタイムスタンプが 2001 年 9 月 16 日 9:15 AM で、リース期間が 24 時間の場合、リース満了時間は 2001 年 9 月 17 日 9:15 AM になります。リース満了日時はクライアントの DHCP ネットワークレコード中に保存され、DHCP マネージャまたは `pntadm` を使って表示されます。

リース期間には、期限切れの IP アドレスを速やかに再利用できるように比較的小さな値を設定します。ただし、リース期間は、DHCP サービスが使用できなくなっても、その DHCP サービスが動作するシステムの修理が終わるまでクライアントが動作を継続できるような長さでなければなりません。一般には、サーバーの予想停止時間の 2 倍を指定します。たとえば、故障部品を検出、交換し、サーバーをリブートするのに 4 時間かかるとすれば、8 時間をリース期間に指定します。

リースネゴシエーションオプションは、リースが満了する前に、クライアントが提供されたリースについてサーバーとネゴシエーションできるかどうかを決めるものです。リースのネゴシエーションが可能な場合には、クライアントがリースの残り時間を常に監視し、リース期間の半分が経過すると、リース期間を元の値に復元する要求を DHCP サーバーに送ります。IP アドレスの数より多くのシステムが存在するために IP アドレスの使用時間を制限したい場合には、リースのネゴシエーションを無効にすることができます。しかし、IP アドレスの数が十分にある場合は、リースネゴシエーションを有効にすべきです。これによって、NFS や telnet セッションなどの TCP 接続を中断するおそれがあるネットワークインタフェースの停止や新しいリースの取得を、クライアントに強制する必要がなくなります。管理者は、サーバー構成時に、リースネゴシエーションをサイト全体に対して有効にすることができます。あるいは、構成マクロの `LeaseNeg` オプションを使用すれば、特定のクライアントやクライアントタイプに対してのみ有効にすることができます。

注 – ネットワークでサービスを提供するシステムはそれ自身の IP アドレスを保持すべきであり、短期的なリースに依存すべきではありません。このようなシステムで DHCP を使用する場合は、常時リースにより IP アドレスを割り当てるのではなく、予約済みの IP アドレスを手動で割り当てるべきです。これによって、このシステムの IP アドレスが使用されなくなったときには、それを検出することができます。

DHCP クライアントのためのルーターの決定

クライアントが自身のローカルネットワークの外側にあるネットワークと通信する場合には、ルーターが必要です。クライアントは、このルーターの IP アドレスを知っている必要があります。

管理者は、DHCP サーバーの構成時に、クライアントが使用するルーターの IP アドレスを指定する必要があります。あるいは、DHCP マネージャを使用する場合には、クライアント自身がルーター検索プロトコルを使ってルーターを検出するように指定することもできます。

そのネットワークのクライアントがルーター検索機能をサポートする場合には、ルーターが 1 つしかなくてもルーター検索プロトコルを使用すべきです。ルーター検索プロトコルを使用すると、クライアントはネットワーク内でのルーター変更に容易に対応できます。たとえば、ルーターに故障が発生したため、新しいアドレスを持つルーターに置き換えられた場合でも、クライアントは新しいアドレスを自動的に検出できます。つまり、新しいルーターアドレスを知るために新しいネットワーク構成を取得する必要はありません。

IP アドレスの管理に必要な選択 (作業マップ)

DHCP サービスの設定の一環として、サーバーが管理する IP アドレスに関する要素を決定します。ネットワークに複数の DHCP サーバーが必要な場合、アドレス管理をどのように分担させるかを決定し、各サーバーにアドレス管理のそれぞれの役割を割り当てるようにします。次の作業マップを使って、IP アドレスの管理に必要な選択を行います。

タスク	説明	インフォメーション
サーバーが管理する IP アドレスを指定する	DHCP サーバーが管理する IP アドレスの数と範囲を決定する	148 ページの「IP アドレスの数と範囲」
サーバーがクライアントのホスト名を自動的に生成するかどうかを決定する	クライアントホスト名について学び、それを使うかどうかを決定する	148 ページの「クライアントホスト名の生成」
クライアントに割り当てる構成マクロを決定する	クライアントに適したマクロを選択できるように、クライアント構成マクロについて確認する	149 ページの「デフォルトのクライアント設定マクロ」
使用するリースタイプを決定する	DHCP クライアントに最適なリースタイプを決定するために、リースタイプについて確認する	149 ページの「動的リースタイプと常時リースタイプ」

IP アドレスの数と範囲

DHCP マネージャを使用すると、サーバーの初期構成時に、総アドレス数とブロックの開始アドレスを指定することにより、そのブロック分の IP アドレス、またはその範囲内の IP アドレスを DHCP の管理下に追加することができます。DHCP マネージャは、この情報から連続するアドレスのリストを作成し、追加します。アドレスが連続していない複数のブロックがある場合は、初期構成の後に DHCP マネージャのアドレスウィザードを再起動して他のアドレスを追加することができます。

IP アドレスの構成を行う前に、アドレスを追加する最初のブロックにあるアドレスの数と、その範囲内の開始のアドレスの IP アドレスを控えておいてください。

クライアントホスト名の生成

DHCP 本来の動的な特性により、IP アドレスはそれを使用するシステムのホスト名に恒久的に関連付けられる訳ではありません。DHCP 管理ツールでは、各 IP アドレスに対応するクライアント名を生成できます。クライアント名には、接頭辞 (ルート名) とダッシュ、それにサーバーから割り当てられる数字が使用されます。たとえば、ルート名が `charlie` なら、クライアント名は `charlie-1`、`charlie-2`、`charlie-3` のようになります。

デフォルトでは、生成されたクライアント名は、それを管理する DHCP サーバーの名前で始まります。これは、複数の DHCP サーバーが存在する環境で便利です。特定の DHCP サーバーがどのクライアントを管理しているのかを DHCP ネットワークテーブルから簡単に知ることができるからです。ただしルート名は、任意の名前に変更できます。

IP アドレスを構成する前に、管理ツールを使ってクライアント名を生成するかどうかと、生成する場合はそのクライアント名に使用するルート名を決めてください。

生成されるクライアント名は、構成時にオプションを選択すれば、/etc/inet/hosts、DNS、または NIS+ 内の IP アドレスに対応付けることができます。詳細は、136 ページの「クライアントホスト名の登録」を参照してください。

デフォルトのクライアント設定マクロ

Solaris DHCP で、マクロは複数のネットワーク構成オプションとその設定値の集まりです。DHCP サーバーは、マクロを使って、どのようなネットワーク構成情報を DHCP クライアントに送信するかを決めます。

管理ツールは、DHCP サーバーの構成時に、システムファイルから情報を収集するだけでなく、プロンプトやコマンド行オプションを通して管理者から直接情報を収集します。 この情報から次のマクロを作成します。

- クライアントネットワークの IP アドレスに対応する名前をもつネットワークアドレスマクロ。このマクロには、ネットワークのどのクライアントでも必要になる情報が含まれています。たとえば、サブネットマスク、ネットワークブロードキャストアドレス、デフォルトルーター、またはルーター検索トークン、さらに、サーバーで NIS/NIS+ を使用する場合には、NIS/NIS+ のドメインとサーバーなどです。 ネットワークに適用可能なその他のオプションも含まれることがあります。
- Locale という名前のロケールマクロ。 このマクロには、時間帯を指定するためのユニバーサル時間 (UTC) からの時間差 (秒単位) が含まれています。
- サーバーのホスト名と同じ名前をもつサーバーマクロ。このマクロには、リースポリシー、時間サーバー、DNS ドメイン、DNS サーバーに関する情報の他に、構成プログラムがシステムファイルから入手したその他の情報が含まれていることがあります。このマクロには、Locale マクロが含まれています。

ネットワークアドレスマクロは、そのネットワーク上に配置されているすべてのクライアントに対して自動的に処理されます。ロケールマクロはサーバーマクロに含まれるため、サーバーマクロを処理する際に処理されます。

最初のネットワークの IP アドレスを構成する際に、これらのアドレスを使用するすべての DHCP クライアントに対して使用するクライアント構成マクロを選択する必要があります。デフォルトではサーバーマクロが選択されます。このサーバーマクロには、このサーバーを使用するすべてのクライアントに必要な情報が含まれています。クライアントは、サーバーマクロに含まれるオプションより前に、ネットワークアドレスマクロに含まれるオプションを受け取ります。マクロの処理順序については、130 130 ページの「マクロ処理の順序」を参照してください。

動的リースタイプと常時リースタイプ

構成しようとするアドレスにリースポリシーが適用されるかどうかは、リースタイプで決まります。DHCP マネージャでは、最初のサーバーの構成時に、追加するアドレスに動的リースを使用するか、常時リースを使用するかを選択できます。dhcpconfig コマンドによる構成では、動的リースが使用されます。

アドレスが動的リースを持つ場合、DHCP サーバーは、そのアドレスをクライアントに割り当て、リース期間を延長し、さらに、そのアドレスが使用されなくなったときは、検出、回収することにより、そのアドレスを管理することができます。アドレスが常時リースを持つ場合は、DHCP サーバーはそのアドレスを 1 つのクライアントだけに割り当てます。そのクライアントは、明示的にそのアドレスを解放するまでアドレスを保持します。アドレスが解放されると、サーバーはアドレスを他のクライアントに割り当てることができます。そのアドレスは、常時リースとして構成されている限り、リースポリシーの対象となることはありません。

IP アドレスの範囲を構成した場合、選択したリースタイプはその範囲内のすべてのアドレスに適用されます。DHCP の利点を最大限に活かすためには、大部分のアドレスに対して動的リースを使用する必要があります。必要な場合には、後で個々のアドレスを常時リースに変更できますが、常時リースの総数は最小限に抑えるようにしてください。

予約済みアドレスとリースタイプ

アドレスは、特定のクライアントに手動で割り当てることにより予約することができます。予約されたアドレスは、関連付けられた常時リースまたは動的リースを持つことができます。予約済みアドレスに常時リースが割り当てられている場合には、以下のようになります。

- そのアドレスに結合されているクライアント以外のクライアントにそのアドレスを割り当てることはできません。
- DHCP サーバーがこのアドレスを別のクライアントに割り当てることはできません。
- DHCP サーバーがこのアドレスを再利用することはできません。

予約済みアドレスに動的リースが割り当てられている場合には、そのアドレスが結合されているクライアント以外のクライアントにそのアドレスを割り当てることはできません。しかしこの場合でも、クライアントは、アドレスが予約済みでないかのように、リース期間を監視し、リース延長のネゴシエーションを行う必要があります。これにより、管理者は、ネットワークテーブルを参照するだけで、クライアントがそのアドレスを使用しているかどうかを監視できます。

初期構成時には、すべての IP アドレスに対して予約済みアドレスを生成することはできません。これは、予約済みアドレスが特定のアドレスに対してのみ使用するためのものだからです。

複数の DHCP サーバーを使用するための計画

複数の DHCP サーバーを構成して IP アドレスを管理する場合には、次のガイドラインに従ってください。

- 各サーバーがそれぞれのアドレス範囲を受け持ち、またアドレス範囲が重複しないように、IP アドレスのプールを分割します。
- 可能であれば、データストアとして NIS+ を選択します。そうでない場合は、テキストファイルを選択し、データストアへの絶対パスとして共有ディレクトリを指定します。バイナリファイルのデータストアを共有することはできません。
- アドレスの所有権が正しく割り当てられるように、またサーバーベースのマクロが自動的に作成されるように、個々のサーバーを個別に構成します。
- 指定された時間間隔で `dhcptab` テーブルのオプションとマクロを走査するようにサーバーを設定します。これによって、すべてのサーバーが最新の情報を使用します。この設定には、DHCP マネージャを使って `dhcptab` の自動読み取りをスケジュールします。詳細は、182 ページの「DHCP サービスの性能オプションのカスタマイズ」を参照してください。
- すべてのクライアントからすべての DHCP サーバーにアクセスできるようにします。これによって、個々のサーバーがそれぞれを相互にサポートすることができます。たとえば、有効な IP アドレスリースを持つクライアントが構成の検証またはリースの延長を行おうとしているとします。このときに、クライアントのアドレスを所有する一次サーバーにクライアントが 20 秒間アクセスを試みても応答がないと、他のサーバーがクライアントに応答します。さらに、あるクライアントが特定のアドレスを要求しても、そのアドレスを所有するサーバーが応答しない場合にも、他のいずれかのサーバーが要求を処理します。クライアントは、要求したアドレスとは異なるアドレスを受け取ります。

リモートネットワーク構成の計画

初期構成が完了すると、リモートネットワーク内の IP アドレスを DHCP の管理下に置くことができます。ただし、システムファイルはサーバー内にないため、DHCP マネージャや `dhcpconfig` はデフォルト値を提供するための情報を検索することができません。したがって、管理者が情報提供する必要があります。リモートネットワークの構成を行う前に、次の情報を用意してください。

- リモートネットワークの IP アドレス
- リモートネットワークのサブネットマスク。これは、ネームサービスの `netmasks` テーブルから取得することができます。ネットワークがローカルファイルを使用する場合は、そのネットワーク内のシステム上にある

/etc/netmasks を参照してください。ネットワークが NIS+ を使用する場合は、niscat netmasks.org_dir コマンドを使用します。ネットワークが NIS を使用する場合には、ypcat -k netmasks.byaddr コマンドを使用します。netmasks テーブルが、管理対象としたいすべてのサブネットに関するトポロジ情報をすべて含んでいることを確認してください。

- ネットワークタイプ – クライアントがネットワークに接続する際、ローカルエリアネットワーク (LAN) 接続を使用するか、ポイントツーポイントプロトコル (PPP) を使用するか。
- ルーティング – クライアントがルーター検索機能を使用できるか。使用できない場合は、クライアントが使用するルーターの IP アドレスを指定する必要があります。
- NIS ドメインと NIS サーバー (使用する場合)
- NIS+ ドメインと NIS+ サーバー (使用する場合)

DHCP ネットワークを追加する手順については、187 ページの「DHCP ネットワークの追加」を参照してください。

DHCP を設定するためのツールの選択

これまでの各節で説明した情報の収集や準備が終わったら、DHCP サーバーを構成します。GUI 対応の DHCP マネージャ、またはコマンド行ユーティリティの dhcpconfig を使用して、サーバーの構成を行うことができます。DHCP マネージャでオプションを選択し、データを入力すると、そのデータから DHCP サーバーが使用する dhcptab テーブルとネットワークテーブルが作成されます。dhcpconfig ユーティリティでは、コマンド行オプションを使ってデータを入力する必要があります。

DHCP マネージャの機能

DHCP マネージャは Java ベースのグラフィカルツールであり、DHCP 構成ウィザードを提供します。DHCP 構成ウィザードは DHCP サーバーとして構成されていないシステム上で DHCP マネージャを最初に実行したときに自動的に起動されます。DHCP 構成ウィザードの一連のダイアログボックスでは、サーバーの構成に不可欠な次の情報を入力する必要があります。データストア形式、リースポリシー、DNS/NIS/NIS+ サーバーとドメイン、ルーターのアドレスなど。ただし、この情報のうちの一部はウィザードがシステムファイルから入手します。したがって、管理者は、情報が正しいかどうかを確認したり、正しくない場合は訂正します。

すると、DHCP サーバーデーモンがサーバーシステム上で起動され、ネットワークのための IP アドレスを構成するために、追加アドレスウィザードを起動するよう求められます。最初は、サーバーのネットワークだけが DHCP 用に構成され、その他のサーバーオプションにはデフォルト値が与えられます。初期構成が完了した後で DHCP マネージャを再度起動すると、ネットワークを追加したり、他のサーバーオプションを変更したりできます。

dhcpconfig 機能

dhcpconfig ユーティリティは、DHCP サーバーの構成や構成解除だけでなく、新しいデータストアへの変換や他の DHCP サーバーとのデータのインポート/エクスポートを行うことができます。dhcpconfig ユーティリティを使用して DHCP サーバーを構成する場合、このユーティリティは142 ページの「システムファイルとネットマスクテーブルの更新」で説明しているシステムファイルから情報を取得します。DHCP マネージャを使用する場合と異なり、dhcpconfig がシステムファイルから得る情報を表示したり確認したりすることはできませんので、dhcpconfig を実行する前にシステムファイルを更新しておく必要があります。コマンド行オプションを使用すると、dhcpconfig がデフォルトでシステムファイルから得る値を無効にすることができます。dhcpconfig ユーティリティは、スクリプト中で使用できます。詳細は、dhcpconfig のマニュアルページを参照してください。

DHCP マネージャと dhcpconfig の比較

下の表に、2 つのサーバー構成ツールの相違点を示します。

表 **8–4** DHCP マネージャと dhcpconfig コマンドの比較

機能	**DHCP** マネージャ	**dhcpconfig (**オプションの指定**)**
システムから収集されたネットワーク情報	システムファイルから収集された情報を表示し、必要な場合は変更することができる	コマンド行オプションを使ってネットワーク情報を指定できる
ユーザーの構成	デフォルト値を使用し、必須ではないサーバーオプションのプロンプトを省略することによって、構成作業を高速化できる。必須ではないオプションは、初期構成後に変更できる	構成処理は最も速いが、多くのオプションを使って値を指定する必要がある場合がある

次の章では、サーバーの構成方法について DHCP マネージャの場合と dhcpconfig ユーティリティの場合をそれぞれ説明します。

第 9 章

DHCP サービスの構成 (手順)

ネットワーク上で DHCP サービスを構成するには、まず 1 番目の DHCP サーバーを構成して起動します。他のサーバーは後で追加できます。データが共有データをサポートする場合、共有された場所から同じデータにアクセスできます。この章では、DHCP サーバーを構成して、ネットワークと関連する IP アドレスを DHCP の管理下に置く手順について説明します。 サーバーの構成解除についても解説します。

また、作業ごとに DHCP マネージャを使用する手順と `dhcpconfig` ユーティリティを使用する手順を説明します。この章では、以下の内容について説明します。

DHCP サーバーの構成と構成解除 (DHCP マネージャ)

この節では、DHCP マネージャを使用して DHCP サーバーを構成および構成解除する手順について説明します。 なお、DHCP マネージャを使用するには、CDE などの X Window System が動作している必要があります。

DHCP を構成していないサーバー上で DHCP マネージャを実行すると、次の画面が表示され、DHCP サーバーまたは BOOTP リレーエージェントのどちらを構成するかを指定できます。

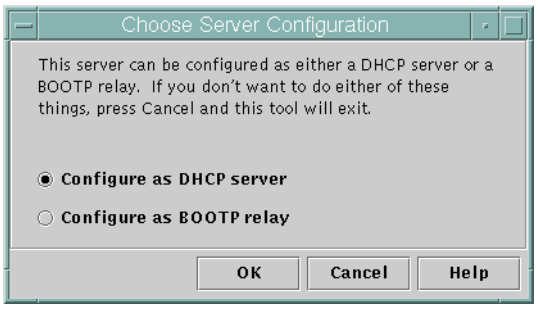


図 **9–1** 「サーバー構成の選択 (Choose Server Configuration)」ダイアログボックス

DHCP サーバーの構成

DHCP サーバーを構成するとき、DHCP マネージャは DHCP 構成ウィザードを起動して、サーバーを構成するために必要な情報を入力するように要求します。図 9–2 に示すような、ウィザードの初期画面が表示されます。

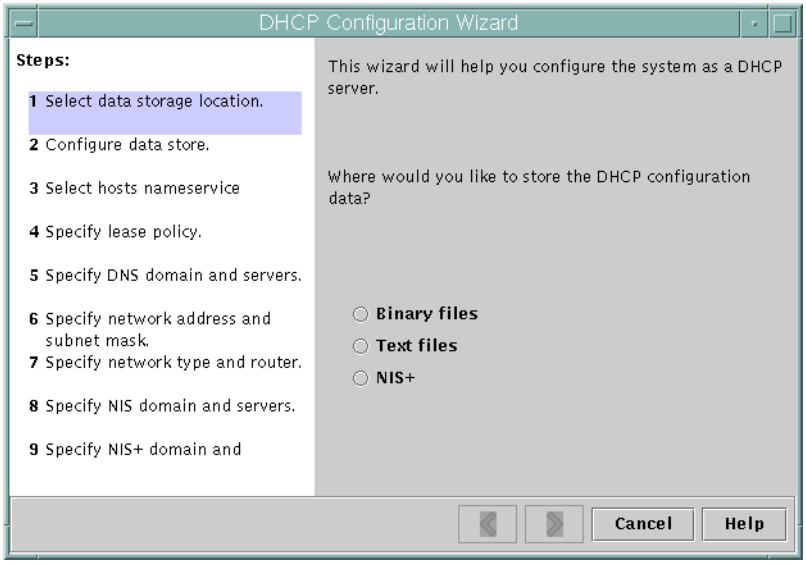


図 **9–2** DHCP 構成ウィザードの初期画面

ウィザードの質問に答えると、DHCP マネージャは 表 9–1 に示されている項目を作成します。

表 **9–1** DHCP サーバーの構成時に作成される項目

項目	説明	目次
サービス構成ファイル、/etc/inet/dhcpsvc.conf	サーバー構成オプションのキーワードおよび値を記録する	データストア形式とその場所、システムのブート時に DHCP デーモンを起動するために in.dhcpd に指定するオプション
dhcptab テーブル	まだ存在しない場合、DHCP マネージャは dhcptab テーブルを生成する	値が割り当てられたマクロとオプション
オプションとして指定する Locale マクロ	ユニバーサル時間 (UTC) とローカルな時間帯との時間差 (秒単位) が含まれる	UTCoffst オプション
サーバーのノード名と一致するように名前が設定されたサーバーマクロ	DHCP サーバーを構成した管理者の入力によって決定された値を持つオプションが含まれる。オプションは、サーバーが所有するアドレスを使用するすべてのクライアントに適用される	Locale マクロと次オプション ■ Timeserv。サーバーの一次 IP アドレスを指し示すように設定されている ■ LeaseTim と、ネゴシエーション可能なリースを選択している場合には LeaseNeg ■ DNSdmain および DNSserv (DNS が構成されている場合) ■ Hostname。このオプションに値を設定してはならない。このオプションが存在すると、ホスト名はネームサービスから取得される必要があることを意味する
ネットワークアドレスマクロ。その名前はクライアントのネットワークアドレスと同じ	DHCP サーバーを構成した管理者の入力によって決定された値を持つオプションが含まれる。オプションは、マクロ名で指定されたネットワーク上に存在するすべてのクライアントに適用される	次のオプション ■ Subnet ■ Router または RDiscvyF ■ Broadcst (ネットワークが LAN の場合) ■ MTU ■ NISdmain および NISservs (NIS が構成されている場合) ■ NIS+dom および NIS+serv (NIS+ が設定されている場合)
ネットワークのためのネットワークテーブル	ネットワークの IP アドレスが作成されるまで、空のテーブルとして作成される	IP アドレスを追加するまで、なし

▼ DHCP サーバーを構成する方法 (DHCP マネージャ)

1. **DHCP** サーバーとして使用するシステムを選択します。

   144 ページの「DHCP サーバーの構成前に必要な選択 (作業マップ)」のガイドラインに従います。

2. データストア、リースポリシー、ルーター情報について決定します。

   144 ページの「DHCP サーバーの構成前に必要な選択 (作業マップ)」のガイドラインに従います。

3. サーバーシステム上でスーパーユーザーになります。

4. 次のコマンドを入力します。

   ```
   #/usr/sadm/admin/bin/dhcpmgr &
   ```

5. 「**DHCP** サーバーとして構成 **(Configure as DHCP Server)**」オプションを選択します。

   サーバーを構成するための DHCP 構成ウィザードが起動します。

6. 計画作成段階で決めた事項に基づいて、オプションを選択するか、要求された情報を入力します。

   わからないことがある場合は、ウィザードウィンドウ内の「ヘルプ (Help)」をクリックして Web ブラウザを開き、DHCP 構成ウィザードのヘルプを表示します。

7. 要求された情報の入力が終了したら、「完了 **(Finish)**」をクリックしてサーバー構成を完了します。

8. アドレス起動ウィザードウィンドウで「はい **(Yes)**」をクリックし、サーバーのアドレスを構成します。

   アドレスウィザードを使用すると、どのアドレスを DHCP の制御下に置くかを指定できます。

9. 計画作成段階で決めた事柄に基づいて、質問に答えます。

   詳細は、147 ページの「IP アドレスの管理に必要な選択 (作業マップ)」を参照してください。わからないことがある場合は、ウィザードウィンドウ内の「ヘルプ (Help)」をクリックして Web ブラウザを開き、アドレス追加ウィザードのヘルプを表示します。

10. 選択した項目を確認し、「完了 **(Finish)**」をクリックしてネットワークテーブルにアドレスを追加します。

    指定した範囲内にある各アドレスのレコードで、ネットワークテーブルが更新されます。

ネットワークウィザードを使用すると、DHCP サーバーにさらにネットワークを追加することができます。187 ページの「DHCP ネットワークの追加」を参照してください。

BOOTP リレーエージェントの構成

BOOTP リレーエージェントを構成するときは、DHCP マネージャは次の動作を行います。

- 要求をリレーすべき DHCP サーバーの IP アドレスを入力するように求める
- `/etc/inet/dhcpsvc.conf` を編集して、BOOTP リレーサービスに必要なオプションを指定する

次に、BOOTP リレーエージェントの構成を選択した場合に表示される画面を示します。

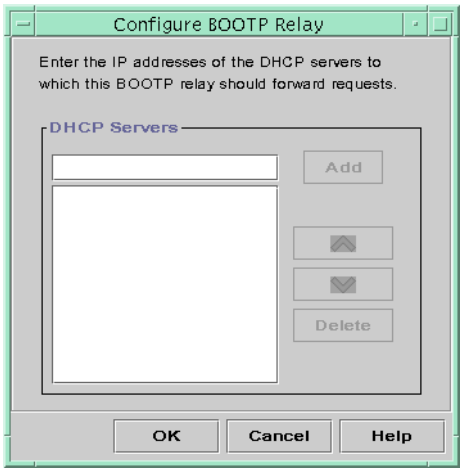


図 **9–3** 「BOOTP リレーの構成 (Configure BOOTP Relay)」ダイアログボックス

▼ BOOTP リレーエージェントを構成する方法 (DHCP マネージャ)

1. **BOOTP** リレーエージェントとして使用するシステムを選択します。
   144 ページの「DHCP を使用するためのサーバーの選択」を参照してください。
2. サーバーシステム上でスーパーユーザーになります。
3. 次のコマンドを入力します。

   #/usr/sadm/admin/bin/dhcpmgr &

   システムが DHCP サーバーまたは BOOTP リレーエージェントとして構成されていない場合は、DHCP 構成ウィザードが起動します。システムがすでに DHCP サーバーとして構成されている場合には、そのサーバーの構成解除をしなければ、そのサーバーを BOOTP リレーエージェントとして構成することはできません。160

ページの「DHCP サーバーと BOOTP リレーエージェントの構成解除」を参照してください。

4. 「**BOOTP** リレーとして構成 **(Configure as BOOTP Relay)**」を選択します。
「BOOTP リレーの構成 (Configure BOOTP Relay)」ダイアログボックスが表示されます。

5. この **BOOTP** リレーエージェントが受信した **BOOTP** または **DHCP** 要求を処理するように構成されている、**1** つ以上の **DHCP** サーバーの **IP** アドレスまたはホスト名を入力し、「追加 **(Add)**」をクリックします。

6. 「了解 **(OK)**」をクリックして、ダイアログボックスを終了します。
DHCP マネージャ はアプリケーションを終了するための「ファイル (File)」メニューと、サーバーを管理するための「サービス (Service)」メニューだけを表示します。その他のメニューオプションは、DHCP サーバー上でのみ有効なため、ここでは使用できません。

DHCP サーバーと BOOTP リレーエージェントの構成解除

DHCP サーバーまたは BOOTP リレーエージェントを構成解除するときは、DHCP マネージャは次の動作を行います。

- DHCP デーモン (`in.dhcpcd`) プロセスを停止する
- デーモンの起動に関する情報とデータストアの場所を記録している `/etc/inet/dhcpsvc.conf` ファイルを削除する

次に、DHCP サーバーの構成解除を選択した場合の画面を示します。

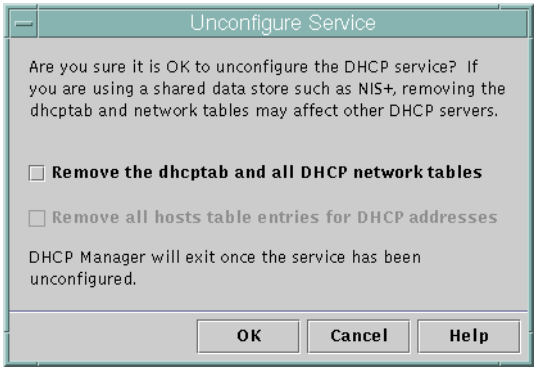


図 **9–4** 「サービスの解除 (Unconfigure Service)」ダイアログボックス

構成解除したサーバー上の DHCP データ

DHCP サーバーを構成解除するときには、dhcptab テーブルと DHCP ネットワークテーブルをどうするかを決定する必要があります。サーバー間でデータを共有している場合は、dhcptab と DHCP の各ネットワークテーブルを削除しないでください。DHCP サーバーの構成を解除することによって、ネットワーク全体に渡って DHCP を使用することができなくなるからです。データの共有は、NIS+ またはエクスポートしたローカルファイルシステムを使用して行うことができます。/etc/inet/dhcpsvc.conf ファイルには、使用されるデータストアとその場所が記録されています。

データを削除するためのいずれのオプションも選択しなければ、データをそのままの形で残し、DHCP サーバーを構成解除できます。サーバーを構成解除し、データをそのままの形で残す場合は、DHCP サーバーを無効にします。

構成解除しようとしているサーバーが所有している IP アドレスを別の DHCP サーバーに所有させる場合、現在のサーバーを構成解除する前に、DHCP データを別の DHCP サーバーに移動しておく必要があります。 詳細については、243 ページの「DHCP サーバー間での構成データの移動 (作業マップ)」を参照してください。

データを削除したい場合は、dhcptab およびネットワークテーブルを削除するためのオプションを選択します。DHCP アドレス用のクライアント名を作成している場合、このようなエントリを hosts テーブル (DNS、/etc/inet/hosts、または NIS+) から削除することも選択できます。

BOOTP リレーエージェントを構成解除する前に、DHCP サーバーへ要求を転送するために、このエージェントを使用しているクライアントが存在しないことを確認してください。

▼ DHCP サーバーまたは BOOTP リレーエージェントを構成解除する方法 (DHCP マネージャ)

1. スーパーユーザーになります。
2. 次のコマンドを入力します。

   ```
   #/usr/sadm/admin/bin/dhcpmgr &
   ```

3. 「サービス **(Service)**」メニューから、「構成解除 **(Unconfigure)**」を選択します。
   「サービスの解除 (Unconfigure Service)」ダイアログボックスが表示されます。サーバーが BOOTP リレーエージェントの場合、このダイアログボックスでリレーエージェントを構成解除することを確認できます。サーバーが DHCP サーバーの場合、DHCP データをどうするかを決定し、このダイアログボックスで選択する必要があります。図 9–4 を参照してください。
4. **(**省略可能**)** データを削除するためのオプションを選択します。

サーバーが共有データ (NIS+ 経由で共有されるデータ、または NFS 経由で共有されるファイル) を使用する場合、データを削除するオプションは選択しないでください。サーバーが共有データを使用しない場合、データを削除するオプションの 1 つまたは両方を選択します。

データの削除については、161 ページの「構成解除したサーバー上の DHCP データ」を参照してください。

5. 「了解 **(OK)**」をクリックします。

DHCP サーバーの構成と構成解除 (dhcpconfig コマンド)

この節では、dhcpconfig とコマンド行オプションを使用して、DHCP サーバーまたは BOOTP リレーエージェントを構成または構成解除する手順について説明します。

▼ DHCP サーバーを構成する方法 (dhcpconfig -D)

1. **DHCP** サーバーとして使用したいシステムを選択します。

   144 ページの「DHCP サーバーの構成前に必要な選択 (作業マップ)」のガイドラインに従います。

2. データストア、リースポリシー、ルーター情報について決定します。

   144 ページの「DHCP サーバーの構成前に必要な選択 (作業マップ)」のガイドラインに従います。

3. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。

4. 次の書式でコマンドを入力します。

   #/usr/sbin/dhcpconfig -D -r *datastore* **-p** *location*

   datastore には、SUNWfiles、SUNWbinfiles、または SUNWnisplus の 1 つを指定します。

   location には、(データストアによって異なる) DHCP データを保存したい場所を指定します。 SUNWfiles および SUNWbinfiles の場合、UNIX の絶対パス名で指定する必要があります。SUNWnisplus の場合、完全指定の NIS+ ディレクトリにする必要があります。

   dhcpconfig ユーティリティは、サーバーマシンのシステムファイルとネットワークファイルを使用して、DHCP サーバーを構成するために使用する値を決定します。デフォルトの値を変更できる dhcpconfig コマンドのその他のオプ

ションについては、dhcpconfig のマニュアルページを参照してください。

5. **1** つまたは複数のネットワークを **DHCP** サービスに追加します。
   ネットワークを追加する手順については、188 ページの「DHCP ネットワークを追加する方法 (dhcpconfig)」を参照してください。

▼ BOOTP リレーエージェントを構成する方法 (dhcpconfig -R)

1. **BOOTP** リレーエージェントとして使用したいシステムを選択します。
   144 ページの「DHCP サーバーの構成前に必要な選択 (作業マップ)」のガイドラインに従います。
2. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
3. 次のコマンドを入力します。

   ```
   # /usr/sbin/dhcpconfig -R addresses
   ```

   addresses には、要求を転送したい DHCP サーバーの IP アドレス (コンマで区切られたリスト) を指定します。

▼ DHCP サーバーまたは BOOTP リレーエージェントを構成解除する方法 (dhcpconfig -U)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. **DHCP** サーバーまたは **BOOTP** リレーエージェントとして動作させるシステム上で、次のコマンドを入力します。

   ```
   # /usr/sbin/dhcpconfig -U
   ```

   サーバーが共有データ (NIS+ 経由で共有されるデータ、またはNFS 経由で共有されるテキストファイル) を使用しない場合、-x オプションも使用すると、dhcptab とネットワークテーブルを削除できます。サーバーが共有データを使用する場合、-x オプションは使用しないでください。-h オプションを使用すると、ホスト名をホストテーブルから削除できます。 dhcpconfig のオプションの詳細については、dhcpconfig のマニュアルページを参照してください。

   データの削除については、161 ページの「構成解除したサーバー上の DHCP データ」を参照してください。

Solaris DHCP クライアントの構成と構成解除

CD-ROM から Solaris オペレーティング環境をインストールすると、DHCP を使用してネットワークインタフェースを構成するかどうかを尋ねるプロンプトが表示されます。これに対して、DHCP を使用するとして応答すると、Solaris のインストール中に、使用しているシステム上で DHCP クライアントソフトウェアが使用可能になります。DHCP を使用するために Solaris クライアントに対して必要な作業はこれだけです。

クライアント上ですでに Solaris オペレーティング環境が稼動しているが、DHCP をまだ使用していない場合は、Solaris システムの構成を解除し、いくつかのコマンドを実行して、システムのブート時に DHCP を使用するようにシステムをセットアップします。

クライアントが Solaris クライアントでない場合、構成手順については、クライアントのマニュアルを参照してください。

▼ Solaris DHCP クライアントを構成する方法

以下の手順が必要なのは、Solaris をインストールする際に DHCP を使用可能にしなかった場合だけです。

1. クライアントシステムでスーパーユーザーになります。
2. システムが対話式構成ではなく事前構成を使用する場合、**sysidcfg** ファイルを編集して、**dhcp** サブキーを **network_interface** キーワードに追加します。

   たとえば、`network_interface=le0 {dhcp}` のようにします。詳細については、`sysidcfg(4)` のマニュアルページを参照してください。
3. 次のコマンドを入力して、システムを構成解除およびシャットダウンします。

   ```
   # sys-unconfig
   ```

   このコマンドで削除される構成情報についての詳細は、`sys-unconfig(1M)` のマニュアルページを参照してください。
4. シャットダウンが完了したら、システムを再起動します。

   システムのリブート時に、システム構成情報を入力するように求めるプロンプトが、`sysidtool` プログラムから出力されます。詳細については、`sysidtool (1M)` のマニュアルページを参照してください。
5. **DHCP** を使用してネットワークインタフェースを構成するようにプロンプトが表示されたら、**Yes** を選択します。

sysidcfg ファイルを使用してシステムを事前構成する場合、network_interface キーワードを挿入して、dhcp を従属キーワードとして指定します。 たとえば、network_interface=le0 {dhcp} のようにします。

▼ Solaris DHCP クライアントを構成解除する方法

1. クライアントシステム上でスーパーユーザーになります。
2. **sysidcfg** ファイルを使用してクライアントを事前構成する場合、**dhcp** サブキーを **network_interface** キーワードから削除します。
3. 次のコマンドを入力して、システムを構成解除およびシャットダウンします。

   ```
   # sys-unconfig
   ```

   このコマンドで削除される構成情報についての詳細は、sys-unconfig(1M) のマニュアルページを参照してください。
4. シャットダウンが完了したら、システムを再起動します。

   システムの構成を解除してあるので、システムのリブート時に、システムの構成情報を入力するように sysidtool プログラムから要求されます。詳細については、sysidtool(1M) のマニュアルページを参照してください。
5. **DHCP** を使用してネットワークインタフェースを構成するように要求するプロンプトが表示されたら、**No** を選択します。

   sysidcfg を使用して構成を指定する場合、プロンプトは表示されません。

第 10 章

DHCP の管理 (手順)

この章では、Solaris DHCP サービスを管理するときに便利な作業について説明します。この章では、サーバー、BOOTP リレーエージェント、およびクライアントに関する作業を説明します。各作業ごとに、DHCP マネージャを使用する手順と DHCP コマンド行ユーティリティを使用する手順を説明します。DHCP コマンド行ユーティリティについての詳細は、マニュアルページを参照してください。

この章に進む前に、DHCP サービスとネットワークの初期構成を済ませておく必要があります。 第 9 章では、DHCP の構成について説明しています。

この章では、次の内容について説明します。

DHCP マネージャ

DHCP マネージャは、DHCP サービスで管理作業を実行するために使用する GUI (Graphical User Interface) です。

DHCP マネージャウィンドウ

DHCP マネージャのウィンドウの表示は、管理プログラムが実行されているサーバーの構成が DHCP サーバーであるか BOOTP リレーエージェントであるかによって異なります。

サーバーが DHCP サーバーとして構成されている場合、DHCP マネージャはタブ形式のウィンドウを使用します。このウィンドウでは、作業に応じたタブを選択します。DHCP マネージャには次のタブがあります。

- アドレス – DHCP が管理しているすべてのネットワークと IP アドレスをリストする。「アドレス (Addresses)」タブから、ネットワークや IP アドレスを個別にまたはまとめて、追加または削除できる。また、各ネットワークや IP アドレスの属性を変更したり、アドレスをまとめて同時に同じ属性に変更したりできる。DHCP マネージャを起動すると、「アドレス (Addresses)」タブが開かれる。
- マクロ – DHCP 構成データベース (`dhcptab`) で利用できるすべてのマクロと、それらのマクロに含まれるオプションをリストする。「マクロ (Macros)」タブからマクロを作成または削除したり、オプションを追加してそれらのオプションに値を設定することでマクロを変更できる。
- オプション – この DHCP サーバーについて定義されたすべてのオプションをリストする。このタブで表示されるオプションは、DHCP プロトコルで定義された標準的なオプションではない。「拡張 (Extended)」、「ベンダー (Vendor)」、または「サイト (Site)」のクラスを持つ、標準オプションを拡張したもの。標準オプションは変更できないため、このタブには表示されない。

次に、DHCP サーバー上で起動した場合の DHCP マネージャウィンドウを示します。

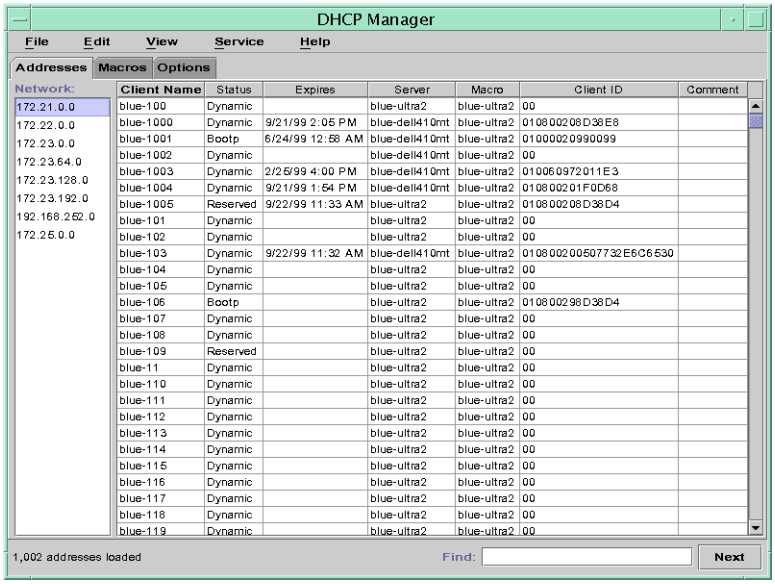


図 **10–1** DHCP サーバーシステム上の DHCP マネージャ

サーバーが BOOTP リレーエージェントとして構成されているとき、これらのタブの情報は BOOTP リレーエージェントには必要ないので、DHCP マネージャウィンドウにこれらのタブは表示されません。BOOTP リレーエージェントの属性を変更し、DHCP マネージャを使用して DHCP デーモンを停止または起動することだけが可能です。次の図は、BOOTP リレーエージェントとして構成されたシステム上で起動した場合の DHCP マネージャウィンドウです。

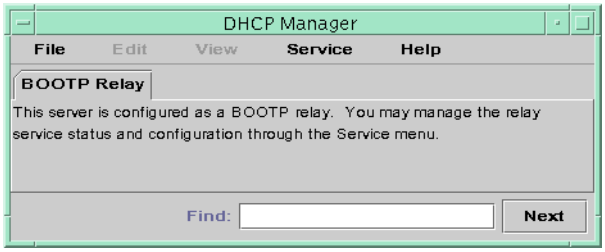


図 **10–2** BOOTP リレーエージェントシステム上の DHCP マネージャ

DHCP マネージャのメニュー

DHCP マネージャのメニューには、次の内容が含まれます。

- 「ファイル **(File)**」– DHCP マネージャを終了する

- 「編集 **(Edit)**」– ネットワーク、アドレス、マクロ、オプションについて管理作業を実行する
- 「表示 **(View)**」– 現在選択されているタブの表示を変更する
- 「サービス **(Service)**」 – DHCP デーモンとデータストアを管理する
- 「ヘルプ **(Help)**」– Web ブラウザを開いて、DHCP マネージャのヘルプを表示する

DHCP マネージャが BOOTP リレーエージェントで実行されている場合、「編集 (Edit)」メニューと「表示 (View)」メニューは使用できません。

すべての DHCP サービス管理機能は、「編集 (Edit)」メニューと「サービス (Service)」メニューで実行されます。「編集 (Edit)」メニューにあるコマンドを使用して、選択されているタブに応じて、ネットワーク、アドレス、マクロ、オプションの作成、削除、変更を行うことができます。また、「アドレス (Addresses)」タブが選択されている場合、「編集 (Edit)」メニューはウィザードも表示します。このウィザードは、ネットワークと複数の IP アドレスを容易に作成できるダイアログのセットです。「サービス (Service)」メニューは、DHCP デーモンを管理するためのコマンドを表示します。これらのコマンドを使用すると、サーバーを起動または停止したり、有効または無効にしたり、構成を変更したり、構成を解除したりできます。「サービス (Service)」メニューには、データストアを変換したり、データをサーバーにエクスポートまたはインポートするためのコマンドもあります。

DHCP マネージャの起動と停止

DHCP マネージャはスーパーユーザーとして DHCP サーバーで実行する必要がありますが、X Window System リモート表示機能を使用すると、他の UNIX システムからリモートで表示できます。

▼ DHCP マネージャを起動および停止する方法

1. **(省略可能) DHCP** サーバーシステムでスーパーユーザーになります。
2. リモートで **DHCP** サーバーシステムにログインしている場合、次の手順でローカルのシステムに **DHCP** マネージャを表示することができます。
   a. ローカルシステムで次のように入力します。

      # **xhost +***server-name*

   b. リモートの **DHCP** サーバーシステムで次のように入力します。

      # **DISPLAY=***local-hostname;***export DISPLAY**
3. 次のコマンドを入力します。

   # **/usr/sadm/admin/bin/dhcpmgr &**

DHCP マネージャウィンドウが開き、サーバーが DHCP サーバーとして構成されている場合には「アドレス (Addresses)」タブを表示します。BOOTP リレーエージェントとして構成されている場合には、タブは表示されません。

4. **DHCP** マネージャを停止するには、「ファイル **(File)**」メニューから「終了 **(Exit)**」を選択します。

   DHCP マネージャウィンドウが閉じます。

DHCP コマンドへのユーザーアクセスの設定

スーパーユーザーにならなくても、 `dhcpconfig`、`dhtadm`、`pntadm` コマンドを実行できるようにするには、これらのコマンドに対して役割によるアクセス制御 (RBAC) を設定する必要があります。RBAC を使用すると、システムで実行することができる処理とユーザーをより正確に定義できます。詳細については、`rbac(5)`、`exec_attr(4)`、 `user_attr(4)` のマニュアルページを参照してください。

次の手順では、ユーザーに DHCP 管理プロファイルを割り当て、そのユーザーが DHCP コマンドを実行できるようにする方法を説明します。

▼ DHCP コマンドへのユーザーアクセスを与える方法

1. **DHCP** サーバーシステム上でスーパーユーザーになります。
2. ファイル **`/etc/user_attr`** を編集して、**DHCP** サービスを管理できるようにしたいユーザーごとに、次の形式のエントリを追加します。

   username`::::type=normal;profiles=DHCP Management`

   たとえば、ユーザー `ram` には、次のエントリを追加します。

   ```
   ram::::type=normal;profiles=DHCP Management
   ```

DHCP サービスの起動と停止

DHCP サービスの起動と停止には、DHCP デーモンの動作に影響する可能性がある処理をいくつか実行する必要があります。希望する結果を得るための正しい手順を選択するには、DHCP サービスの起動と停止、有効と無効、および構成と構成解除を理解しておく必要があります。次に、これらの用語について説明します。

- 起動、停止、再起動コマンドは、現在のセッションのデーモンだけに影響します。つまり、DHCP サービスを停止すると現在実行中のデーモンは終了しますが、システムを再起動すると終了したデーモンは再び起動します。サービスを停止しても、DHCP データテーブルは影響されません。
- 有効コマンドと無効コマンドは、現在のセッションと将来のセッションのデーモンに影響します。DHCP サービスを無効にすると、現在実行中のデーモンは終了し、サーバーを再起動しても終了したデーモンは起動しません。DHCP デーモンがシステム起動時に自動的に起動するように設定しておく必要があります。DHCP データテーブルは影響されません。DHCP サービスを有効または無効にできるのは、DHCP マネージャだけです。
- 構成解除コマンドは、デーモンをシャットダウンし、システムの再起動時にデーモンが起動されないようにし、DHCP データテーブルを削除できるようにします。構成解除については、第 9 章を参照してください。

注 – サーバーに複数のネットワークインタフェースがある場合にすべてのネットワークでは DHCP サービスを提供したくない場合、185 ページの「DHCP サービスを監視するネットワークインタフェースの指定」を参照してください。

この節では、DHCP サービスを起動および停止、有効および無効にするときの手順について説明します。

▼ DHCP サービスを起動および停止する方法 (DHCP マネージャ)

1. **DHCP** サーバーシステム上でスーパーユーザーになります。
2. **DHCP** マネージャを起動します。
   この手順については、170 ページの「DHCP マネージャを起動および停止する方法」を参照してください。
3. 次の操作の **1** つを選択します。
   a. 「サービス **(Service)**」メニューから「起動 **(Start)**」を選択して、**DHCP** サービスを起動します。
   b. 「サービス **(Service)**」メニューから「停止 **(Stop)**」を選択して、**DHCP** サービスを停止します。
      DHCP デーモンは、手動で再開されるかまたはシステムが再起動するまで停止します。
   c. 「サービス **(Service)**」メニューから「再開 **(Restart)**」を選択して、**DHCP** サービスを停止しすぐに再起動します。

▼ DHCP サービスを起動および停止する方法 (コマンド行)

1. サーバー上でスーパーユーザーになります。
2. 次の操作の **1** つを選択します。

   a. **DHCP** サービスを開始するには、次のコマンドを入力します。

      ```
      # /etc/init.d/dhcp start
      ```

      `/etc/inet/dhcpsvc.conf` に設定された構成パラメータを使用して、DHCP デーモンが起動します。

   b. **DHCP** サービスを停止するには、次のコマンドを入力します。

      ```
      # /etc/init.d/dhcp stop
      ```

      DHCP デーモンは、手動で再開されるかまたはシステムが再起動するまで停止します。

▼ DHCP サービスを有効または無効にする方法 (DHCP マネージャ)

1. **DHCP** マネージャを起動します。
2. 次の操作の **1** つを選択します。

   a. 「サービス **(Service)**」メニューから「有効 **(Enable)**」を選択して、**DHCP** サービスをすぐに起動し、システム起動時に **DHCP** サービスが自動的に起動するように構成します。

   b. 「サービス **(Service)**」メニューから「無効 **(Disable)**」を選択して、**DHCP** サービスをすぐに停止し、システム起動時に **DHCP** サービスが自動的に起動しないように構成します。

DHCP サービスオプションの変更 (作業マップ)

DHCP サービスの一部の追加機能について値を変更できます。これらの機能の一部は、DHCP マネージャを使用した初期構成の際には表示されなかったものです。`dhcpconfig` を使用してサーバーを構成した場合、サーバーはこれらのオプションに

関してデフォルト値を使用します。DHCP マネージャの「サービスオプションの変更 (Modify Service Options)」ダイアログボックスを使用するか、`in.dhcpd` コマンドでオプションを指定して、サービスオプションを変更できます。

次の作業マップに、サービスオプションに関する作業と、使用する手順を示します。

タスク	説明	参照先
ログオプションの変更	詳細ログを使用可能または使用不可にし、DHCP トランザクションのログを使用可能または使用不可にし、`syslog` 機能を選択して DHCP トランザクションログに使用する	177 ページの「詳細 DHCP ログメッセージを生成する方法 (DHCP マネージャ)」 177 ページの「詳細 DHCP ログメッセージを生成する方法 (コマンド行)」 177 ページの「DHCP トランザクションログを有効または無効にする方法 (DHCP マネージャ)」 178 ページの「現在のセッションについて DHCP トランザクションログを有効または無効にする方法 (コマンド行)」 179 ページの「DHCP トランザクションを別の `syslog` ファイルに記録する方法」
DNS 更新オプションの変更	ホスト名を提供するクライアント用の DNS エントリをサーバーが追加することを使用可能または使用不可にし、サーバーが DNS を更新するときに費やすことができる最大時間を決定する	180 ページの「DHCP クライアント用に動的 DNS 更新を有効にする方法」
重複 IP アドレス検出の使用可能または使用不可	DHCP サーバー が IP アドレスをクライアントに提供する前に IP アドレスが使用されていないことを確認することを、使用可能または使用不可にする	183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (DHCP マネージャ)」 183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (コマンド行)」
DHCP サーバーの構成情報の読み込みに関するオプションの変更	指定された間隔での `dhcptab` の自動読み込みを使用可能または使用不可にする。また、読み込み間隔を変更する	183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (DHCP マネージャ)」 183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (コマンド行)」
リレーエージェントホップ数の変更	DHCP デーモンでドロップされる前に要求をやり取りできるネットワーク数を増減する	183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (DHCP マネージャ)」 183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (コマンド行)」

タスク	説明	参照先
提供される IP アドレスがキャッシュされている時間の変更	新しいクライアントに IP アドレスを提供する前に DHCP サービスが提供された IP アドレスを予約する秒数を増減する	183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (DHCP マネージャ)」 183 ページの「DHCP サーバー性能オプションをカスタマイズする方法 (コマンド行)」

次に、DHCP マネージャの「サービスオプションの変更 (Modify Service Options)」ダイアログボックスを示します。

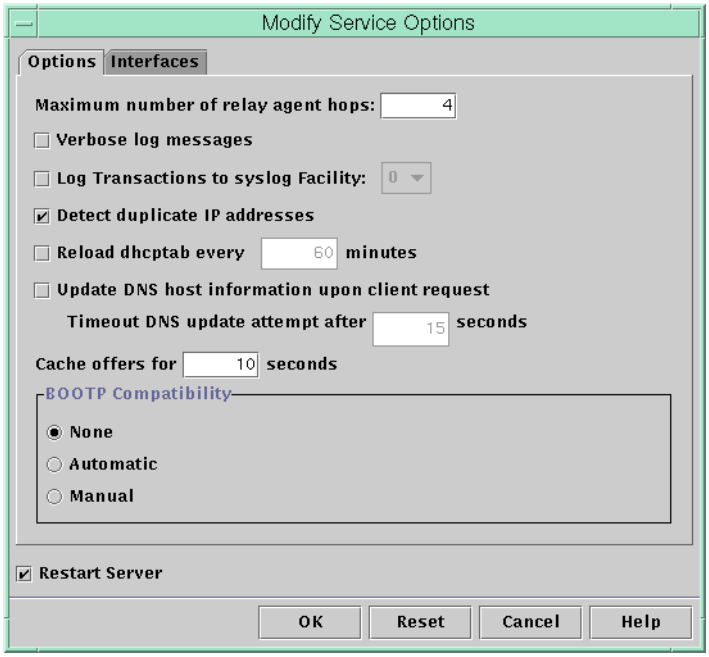


図 **10–3** 「サービスオプションの変更 (Modify Service Options)」ダイアログボックス

DHCP ログオプションの変更

DHCP サービスは、DHCP サービスメッセージと DHCP トランザクションを `syslog` に記録できます。`syslog` についての詳細は、`syslogd(1M)` および `syslog.conf(4)` のマニュアルページを参照してください。

`syslog` に記録された DHCP サービスメッセージには、次のものがあります。

- エラーメッセージ。DHCP サービスがクライアントまたは管理者の要求を完了するのを妨げる条件を、管理者に通知する

- 警告と通知。DHCP サービス完了を妨げはしないが、正常終了しなかった状態を管理者に通知する

DHCP デーモンの詳細オプションを使用して、報告される情報を増やすことができます。詳細メッセージ出力は、DHCP に関する問題の障害追跡に役立つ場合があります。177 ページの「詳細 DHCP ログメッセージを生成する方法 (DHCP マネージャ)」を参照してください。

もう 1 つの有用な障害追跡方法は、トランザクションの記録です。トランザクションは、DHCP サーバーや BOOTP リレーとクライアントとの間のすべての交換に関する情報を提供します。DHCP トランザクションには、次のものがあります。

- ASSIGN – IP アドレスの割り当て
- ACK – サーバーは、クライアントが提供された IP アドレスを受け入れることを認め、構成パラメータを送る
- EXTEND – リース期間の延長
- RELEASE – IP アドレスの解放
- DECLINE – クライアントはアドレス割り当てを拒否している
- INFORM – クライアントはネットワーク構成パラメータを要求しているが IP アドレスは要求していない
- NAK – サーバーは、クライアントに対して、すでに使用された IP アドレスの使用要求を認めない
- ICMP_ECHO – サーバーは、可能性のある IP アドレスが他のホストですでに使用中であることを検出する

BOOTP リレートランザクションには、次のものがあります。

- RELAY-CLNT – DHCP クライアントから DHCP サーバーへリレーされるメッセージ
- RELAY-SRVR – DHCP サーバーから DHCP クライアントへリレーされるメッセージ

トランザクションのログは、デフォルトでは使用不可になっています。トランザクションの記録が使用可能になると、デフォルトでは `local0` `syslog` 機能が使用されます。DHCP トランザクションメッセージは、通知の `syslog` 重要度付きで生成されるため、デフォルトでは他の通知が記録されるファイルにトランザクションが記録されます。ただし、トランザクションはローカルの機能を使用するため、`syslog.conf` ファイルを編集して別のログファイルを指定すると、他の通知とは別々にトランザクションメッセージを記録できます。

トランザクションの記録を使用可能または使用不可にできます。177 ページの「DHCP トランザクションログを有効または無効にする方法 (DHCP マネージャ)」で説明しているように、0 から 7 までの異なる `syslog` 機能を指定できます。また、サーバーシステムの `syslog.conf` ファイルを編集する場合、179 ページの「DHCP トランザクションを別の `syslog` ファイルに記録する方法」で説明しているように、`syslogd` に指示して DHCP トランザクションメッセージを別なファイルに保管することもできます。

▼ 詳細 DHCP ログメッセージを生成する方法 (DHCP マネージャ)

1. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。
2. 「詳細ログメッセージ **(Verbose Log Messages)**」を選択します。
3. 「サーバーの再起動 **(Restart Server)**」が選択されていない場合は、選択します。
4. 「了解 **(OK)**」をクリックします。

   このセッション以降、このオプションを再設定するまで、デーモンは詳細モードで動作します。メッセージを表示するのに時間がかかるため、詳細モードではデーモンの効率が低下する場合があります。

▼ 詳細 DHCP ログメッセージを生成する方法 (コマンド行)

1. **DHCP** サーバーシステム上でスーパーユーザーになります。
2. 次のコマンドを入力して、**DHCP** デーモンを停止してから、詳細モードで再起動します。

   ```
   # /etc/init.d/dhcp stop
   # /usr/lib/inet/in.dhcpd -v options
   ```

   options には、デーモンを起動するために通常使用するオプションを指定します。

   デーモンは、このセッションについて詳細モードで実行されます。

   メッセージを表示するのに時間がかかるため、詳細モードではデーモンの効率が低下する場合があります。

▼ DHCP トランザクションログを有効または無効にする方法 (DHCP マネージャ)

この手順では、以後すべての DHCP サーバーセッションに関するトランザクションログを有効または無効にします。

1. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。
2. 「**syslog** へのログトランザクション **(Log Transactions to Syslog Facility)**」を選択します。

   トランザクションログを無効にするには、このオプションの選択を解除します。
3. **(**省略可能**)** ローカル機能を **0** から **7** まで選択して、トランザクションログに使用します。

デフォルトでは、DHCP トランザクションは、システム通知が記録される場所へ記録されます。この場所は `syslogd` の構成によって決まります。DHCP トランザクションを他のシステム通知とは別の場所に記録したい場合は、179 ページの「DHCP トランザクションを別の `syslog` ファイルに記録する方法」を参照してください。

トランザクションログを有効にすると、メッセージファイルのサイズは急速に大きくなります。

4. 「サーバーの再起動 **(Restart Server)**」が選択されていない場合は、選択します。
5. 「了解 **(OK)**」をクリックします。

   このセッション以降、このダイアログボックスで無効にするまで、デーモンは選択された `syslog` 機能にトランザクションを記録します。

▼ 現在のセッションについて DHCP トランザクションログを有効または無効にする方法 (コマンド行)

1. **DHCP** サーバーシステム上でスーパーユーザーになります。
2. 現在のセッションについてログを有効にするには、次のコマンドを入力します。

   ```
   # /etc/init.d/dhcp stop
   # /usr/lib/inet/in.dhcpd -l syslog-local-facility
   ```

   syslog-local-facility には、0 から 7 までの数字を指定します。このオプションを省略すると、デフォルトで 0 が使用されます。177 ページの「DHCP トランザクションログを有効または無効にする方法 (DHCP マネージャ)」を参照してください。

注 – トランザクションログを使用不可にするには、`in.dhcpd` 起動時に `-l` オプションを省略します。

デフォルトでは、DHCP トランザクションは、システム通知が記録される場所へ記録されます。この場所は `syslogd` の構成によって決まります。DHCP トランザクションを他のシステム通知とは別の場所に記録したい場合は、179 ページの「DHCP トランザクションを別の `syslog` ファイルに記録する方法」を参照してください。

トランザクションログを有効にすると、メッセージファイルのサイズは急速に大きくなります。

▼ DHCP トランザクションを別の syslog ファイルに記録する方法

1. **DHCP** サーバーシステム上でスーパーユーザーになります。
2. サーバーシステムの **/etc/syslog.conf** ファイルを編集し、次の書式の行を追加します。

   `local`*n*`.notice` *path-to-logfile*

   n にはトランザクションログ用に指定した `syslog` 機能番号を指定します。*path-to-logfile* には、トランザクションを記録するファイルへの絶対パスを指定します。

   たとえば、次のような行を追加できます。

   ```
   local0.notice /var/log/dhcpsrvc
   ```

   `syslog.conf` ファイルの詳細については、`syslog.conf(4)` のマニュアルページを参照してください。

DHCP サーバーによる動的 DNS 更新の有効化

DHCP クライアントにリースされた IP アドレスにホスト名がマップされているとき、ホスト名を供給するように DHCP サーバーを構成している場合は、DHCP サーバーは割り当てられている名前をクライアントに通知します。DHCP クライアントが独自のホスト名を供給するように DHCP サーバーを構成している場合は、DHCP サーバーは DHCP クライアントの代わりに DNS 更新を行います。

DNS はインターネット用に基本的なネームサービスを提供します。DNS 更新が行われると、他のシステムは DHCP クライアントシステムを名前で参照できます。

自身のホスト名を供給する DHCP クライアントのホスト名で DNS サービスを更新するように DHCP サービスを構成できます。 システム名が DNS で登録されているとき、システムはドメインの外からも見えます。DNS 更新機能を有効にするには、DNS サーバー、DHCP サーバー、および DHCP クライアントをすべて正しく設定する必要があります。要求された名前は、ドメイン内にある他のシステムが使用していてはいけません。

DHCP サーバーの DNS 更新機能が動作するのは、次の条件がすべて真であるときです。

- DNS サーバーが RFC 2136 をサポートする
- BIND ベースの DNS ソフトウェアは、DHCP または DNS のサーバーシステムのどちらにあるかにかかわらず、バージョンが v8.2.2 であり、パッチレベルがレベル 5 以降である
- DNS サーバーがDHCP サーバーからの動的 DNS 更新を受け入れるように構成されている
- DHCP サーバーが動的 DNS 更新を行うように構成されている
- DNS サポートが、DHCP サーバー上の DHCP クライアントのネットワーク用に構成されている

- DHCP クライアントが、その DHCP 要求メッセージで要求されたホスト名を供給するように構成されている
- 要求されたホスト名が、DHCP 所有のアドレスに対応するか、対応するアドレスを持っていない

▼ DHCP クライアント用に動的 DNS 更新を有効にする方法

注 – 動的 DNS 更新は本来セキュリティ上のリスクであることに注意してください。

デフォルトでは、Solaris DNS デーモン (`in.named`) は動的更新を許可しません。 動的 DNS 更新の承認が与えられるのは、DNS サーバーシステム上にある `named.conf` 構成ファイルの適切なゾーン内において、`allow-update` キーワードに要求したホストの IP アドレスが割り当てられている場合です。他のセキュリティは提供されません。動的 DNS 更新を有効にするときには、この機能のユーザーに対する便利さとセキュリティリスクのバランスを注意深く考慮する必要があります。

---

1. **DNS** サーバーで、スーパーユーザーとして **`/etc/named.conf`** ファイルを編集します。
2. 適切なドメインの **`zone`** セクションを見つけて、**`allow-update`** キーワードに **DHCP** サーバーの **IP** アドレスを追加します。

   たとえば、DHCP サーバーのアドレスが 10.0.0.1 と 10.0.0.2 である場合、`dhcp.domain.com` ゾーン用の `named.conf` ファイルを次のように変更します。

   ```
   zone "dhcp.domain.com" in {
               type master;
               file "db.dhcp";
               allow-update { 10.0.0.1; 10.0.0.2; };
   };

   zone "10.IN-ADDR.ARPA" in {
               type master;
               file "db.10";
               allow-update { 10.0.0.1; 10.0.0.2; };
   };
   ```

   DHCP サーバーが A と PTR の両方のレコードを DNS サーバー上で更新できるように、両方のゾーンの `allow-update` を有効にする必要があります。
3. **DHCP** サーバー上で、**DHCP** マネージャを起動します。
4. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。

   「サービスオプションの変更 (Modify Service Options)」ダイアログボックスが開きます。

5. 「クライアント要求により **DNS** ホスト情報を更新 **(Update DNS Host Information Upon Client Request)**」を選択します。

6. **DNS** サーバーからの応答を待ち、時間切れになるまでの秒数を指定し、「了解 **(OK)**」をクリックします。

   通常はデフォルト値です。時間切れに関する問題が発生した場合は、後でこの値を増やすことも可能です。

7. 「マクロ **(Macros)**」タブをクリックして、正しい **DNS** ドメインが指定されていることを確認します。

   `DNSdmain` オプションを渡すには、動的 DNS 更新のサポートを期待するクライアントへの正しいドメイン名と共に指定する必要があります。 デフォルトでは、`DNSdmain` がサーバーマクロ中に指定されています。この値は、各 IP アドレス構成マクロとして使用されます。

8. **DHCP** サービスを要求するときにそのホスト名を指定するように **DHCP** クライアントを設定します。

   Solaris DHCP クライアントを使用する場合は、181 ページの「特定のホスト名に応答するように Solaris クライアントを有効にする方法」を参照してください。
   Solaris DHCP クライアント以外のクライアントを使用する場合は、その DHCP クライアントのマニュアルを参照してください。

▼ 特定のホスト名に応答するように Solaris クライアントを有効にする方法

1. クライアントシステム上で、スーパーユーザーとして **`/etc/default/dhcpagent`** ファイルを編集します。

2. **`/etc/default/dhcpagent`** ファイルでキーワード **`REQUEST_HOSTNAME`** を見つけて、次のように変更します。

   ```
   REQUEST_HOSTNAME=yes
   ```

   キーワードの前にコメント記号 (#) がある場合は、コメント記号を削除します。キーワードがない場合は、キーワードを挿入します。

3. クライアントシステム上で **`/etc/hostname.`***interface* ファイルを編集して、次の行を追加します。

   `inet` *hostname*

   hostname には、使用したいクライアントの名前を指定します。

4. スーパーユーザーとして次のコマンドを入力します。すると、クライアントは再起動時に完全な **DHCP** ネゴシエーションを実行します。

   ```
   # pkill dhcpagent
   # rm /etc/dhcp/interface.dhc
   # reboot
   ```

DHCP サーバーは、ホスト名をクライアントに割り当てる前に、そのホスト名がネットワーク上にある別のシステムによって使用されていないことを確認します。構成によって異なりますが、DHCP サーバーはネームサービスをクライアントのホスト名に更新することもあります。

DHCP サービスの性能オプションのカスタマイズ

DHCP サービスの性能に影響するオプションを変更することができます。これらのオプションについて、次の表で説明します。

表 **10–1** DHCP サービスの性能に影響するオプション

サーバーオプション	説明	**/etc/inet/dhcpsvc.conf** 内のエントリ
BOOTP リレーエージェントホップ数	一定数以上の BOOTP リレーエージェントを通過すると、その要求はドロップされます。デフォルトのリレーエージェントホップの最大数は、4 つです。要求が複数のリレーエージェントを通過してから DHCP サーバーに到達するようにネットワークを設定していない限り、この 4 という数を超えることはありません。	RELAY_HOPS= *integer*
提供前の IP アドレスの利用可能性の確認	サーバーはデフォルトで、IP アドレスをクライアントに提供する前に、そのアドレスがまだ使用されていないことを確認します。この機能を使用不可にして、提供にかかる時間を減少させることができますが、IP アドレスを重複して使用する危険が発生します。	ICMP_VERIFY=TRUE/FALSE
指定された間隔での dhcptab の自動読み込み	指定した間隔 (分単位) で dhcptab を自動的に読み込むようにサーバーを設定することができます。ネットワークの構成情報を頻繁に変更せず、複数の DHCP サーバーを持っていない場合は、dhcptab を自動的に再読み込みする必要はありません。また、DHCP マネージャには、データ変更後にサーバーに dhcptab を再読み込みさせるようにするオプションもあります。	RESCAN_INTERVAL=*min*
提供された IP アドレスを予約する時間の長さ	サーバーは、IP アドレスをクライアントに提供した後、そのキャッシュに書き込みます。キャッシュに書き込まれている間、サーバーはそのアドレスを再び提供することはしません。提供した IP アドレスがキャッシュに書き込まれている秒数を変更することができます。デフォルトは 10 秒です。低速のネットワークでは、このキャッシュ時間を延長する必要があります。	OFFER_CACHE_TIMEOUT=*sec*

次の手順では、これらのオプションを変更する方法を説明します。

▼ DHCP サーバー性能オプションをカスタマイズする方法 (DHCP マネージャ)

1. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。
2. 要求が通過できる **BOOTP** リレーエージェントの数を変更するには、異なるリレーエージェントホップの最大数を指定します。
3. **IP** アドレスが使用されていないことを **DHCP** サーバーで確認してからクライアントにそのアドレスを提供するようにするには、「重複 **IP** アドレスの検出 **(Detect Duplicate IP Addresses)**」を選択します。
4. 指定された間隔で **DHCP** サーバーに **`dhcptab`** を読み込ませるには、「**dhcptab** の読み込み周期 **(Reload dhcptab Every *n* Minutes)**」を選択して、その間隔を分数で入力します。
5. サーバーが **IP** アドレスを提供した後にそのアドレスを予約しておく期間を変更するには、「キャッシュの更新 **(Cache Offers for *n* Seconds)**」フィールドに秒数を入力します。
6. 「サーバーの再起動 **(Restart Server)**」が選択されていない場合は、選択します。
7. 「了解 **(OK)**」をクリックします。

▼ DHCP サーバー性能オプションをカスタマイズする方法 (コマンド行)

この手順でオプションを変更する場合、変更されたオプションが影響するのは現在のサーバーセッションだけです。 DHCP サーバーシステムを再起動すると、DHCP サーバーは、サーバー構成中に指定された設定を使用して起動します。この設定を将来のセッションにも適用したい場合は、DHCP マネージャを使用してオプションを変更する必要があります。

1. **DHCP** サーバーシステム上でスーパーユーザーになります。
2. 次のコマンドを入力します。

```
# /etc/init.d/dhcp stop
# /usr/lib/inet/in.dhcpd options
```

この場合、*options* は次のようになります。

`-h` *relay-hops*	デーモンが DHCP または BOOTP のデータグラムをドロップする前に発生することができるリレーエージェントホップの最大数を指定する

`-n`	重複 IP アドレスの自動検出を使用不可にする。この設定は推奨されない
`-t` *dhcptab_rescan_interval*	DHCP サーバーが `dhcptab` 情報を自動的に読み込み直す間隔を分で指定する
`-o` *seconds*	DHCP サーバーが DHCP クライアントを検索するために提供した IP アドレスをキャッシュに書き込んでおく秒数を指定する。デフォルトは 10 秒

たとえば次のコマンドは、ホップ数を 2 に設定し、重複 IP アドレスの検出を使用不可にし、自動再読み込み間隔を 30 秒に設定し、キャッシュ時間を 20 秒にしています。

```
# /usr/lib/inet/in.dhcp -h 2 -n -t 30 -o 20
```

DHCP ネットワークの追加、変更、削除 (作業マップ)

DHCP サーバーを構成する際に、DHCP サービスを使用するために少なくとも 1 つのネットワークを構成する必要があります。いつでもネットワークを追加することができます。

次の作業マップに、DHCP ネットワークを利用する際に必要な作業とその手順を示します。

タスク	説明	参照先
サーバーネットワークインタフェースでの DHCP サービスの使用可能と使用不可	デフォルトの動作では、DHCP 要求に関するすべてのネットワークインタフェースを監視するが、変更できる	186 ページの「DHCP 監視用のネットワークインタフェースを指定する方法 (DHCP マネージャ)」
DHCP サービスに新しいネットワークを追加	ネットワーク上の IP アドレスを管理するため、ネットワークを DHCP の管理下に置く	188 ページの「DHCP ネットワークを追加する方法 (DHCP マネージャ)」
DHCP に管理されたネットワークのパラメータの変更	特定のネットワークのクライアントに渡される情報を変更する	189 ページの「DHCP ネットワークの構成を変更する方法 (DHCP マネージャ)」 190 ページの「DHCP ネットワークの構成を変更する方法 (`dhtadm`)」

タスク	説明	参照先
DHCP サービスからのネットワークの削除	これ以降、ネットワーク上の IP アドレスが DHCP によって管理されないようにネットワークを削除する	191 ページの「DHCP ネットワークを削除する方法 (DHCP マネージャ)」 192 ページの「DHCP ネットワークを削除する方法 (pntadm)」

DHCP サービスを監視するネットワークインタフェースの指定

デフォルトでは、dhcpconfig および DHCP マネージャの構成ウィザードは両方とも、DHCP サーバーがすべてのサーバーシステムのネットワークインタフェースを監視するように構成します。新しいネットワークインタフェースをサーバーシステムに追加した場合、システムを起動すると、DHCP サーバーがこの新しいネットワークインタフェースを自動的に監視します。そのため、どのネットワークを追加してもそのネットワークインタフェースを通して監視できます。

ただし、DHCP マネージャによって、DHCP サービスでどのネットワークインタフェースを監視して、どのネットワークインタフェースを無視するかを指定することもできます。特定のネットワーク上で DHCP サービスを提供したくない場合、インタフェースを無視すると便利なことがあります。

すべてのインタフェースを無視するように設定してから新しいインタフェースをインストールした場合、サーバーが持つ監視対象インタフェースのリストにそのインタフェースを追加しない限り、DHCP サーバーはそのインタフェースを無視します。インタフェースは DHCP マネージャで指定できます。

この節では、DHCP が監視または無視するネットワークインタフェースを指定できるようにするための手順についても説明します。 この手順では、DHCP マネージャの「サービスオプションの変更 (Modify Service Options)」ダイアログボックスの「インタフェース (Interfaces)」タブを使用します (次図を参照)。

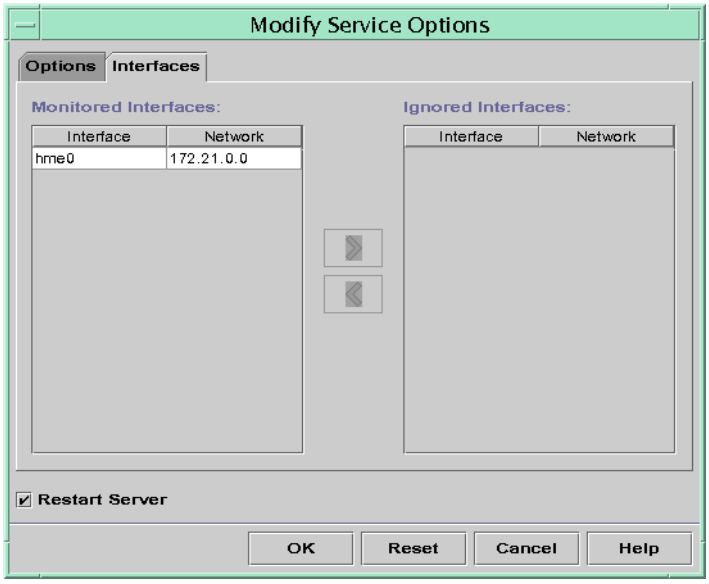


図 **10–4** 「サービスオプションの変更 (Modify Service Options)」ダイアログボックスの「インタフェース (Interfaces)」タブ

▼ DHCP 監視用のネットワークインタフェースを指定する方法 (DHCP マネージャ)

1. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。
   「サービスオプションの変更 (Modify Service Options)」ダイアログボックスが表示されます。
2. 「インタフェース **(Interfaces)**」タブを選択します。
3. 適切なネットワークインタフェースを選択し、矢印ボタンをクリックして、インタフェースを「監視中のインタフェース **(Monitored Interfaces)**」または「削除するインタフェース **(Ignored Interfaces)**」のリストに移動します。
   たとえば、インタフェースを無視するには、「監視中のインタフェース (Monitored Interfaces) リストからそのインタフェースを選択し、右矢印ボタンをクリックして、「削除するインタフェース (Ignored Interfaces)」リストに移動します。
4. 「サーバーの再起動 **(Restart Server)**」が選択されていることを確認して、「了解 **(OK)**」をクリックします。

DHCP ネットワークの追加

DHCP マネージャを使用してサーバーを構成する場合、最初のネットワーク (通常、サーバーシステムの一次インタフェース上にあるローカルのネットワーク) も同時に構成します。さらに他のネットワークを構成したい場合は、DHCP マネージャの DHCP ネットワークウィザードを使用します。

`dhcpconfig -D` を使用してサーバーを構成する場合、DHCP サービスが提供されるすべてのネットワークを手動で構成する必要があります。 詳細については、188 ページの「DHCP ネットワークを追加する方法 (`dhcpconfig`)」を参照してください。

次の図に、DHCP マネージャの DHCP ネットワークウィザードの初期ダイアログボックスを示します。

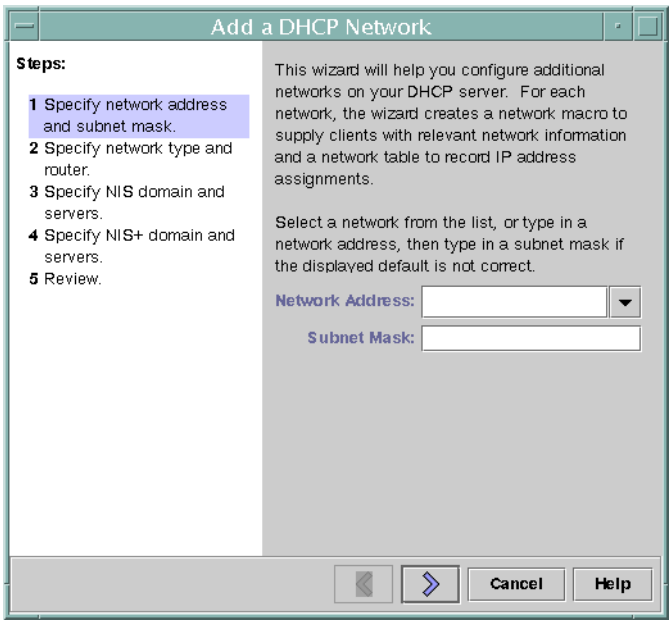


図 **10–5** DHCP マネージャのネットワークウィザード

新しいネットワークを構成すると、DHCP マネージャが次の内容を作成します。

- データストアにネットワークテーブルを作成します。新しいネットワークは、DHCP マネージャの「アドレス (Addresses)」タブにあるネットワークリストに表示されます。
- このネットワークに常駐するクライアントで必要とする情報を含むネットワークマクロを作成します。このネットワークマクロの名前はネットワークの IP アドレスと一致します。ネットワークマクロはデータストア内の `dhcptab` に追加されます。

▼ DHCP ネットワークを追加する方法 (DHCP マネージャ)

1. **DHCP** マネージャの「アドレス **(Addresses)**」タブをクリックします。
   すでに DHCP サービス用に構成されているネットワークがリストされます。
2. 「編集 **(Edit)**」メニューから「ネットワークウィザード **(Network Wizard)**」を選択します。
3. 計画作成段階で決めた事項に基づいて、オプションを選択するか要求された情報を入力します。
   計画については、151 ページの「リモートネットワーク構成の計画」で説明しています。
   ウィザードに関してわからないことがある場合は、ウィザードウィンドウ内のヘルプをクリックして Web ブラウザを開き、ウィザードのヘルプを表示します。
4. 必要な情報を入力し終えた後、「完了 **(Finish)**」をクリックしてネットワークの構成を終了します。
   ネットワークウィザードが、そのネットワークの IP アドレスと一致する名前のネットワークマクロを作成します。DHCP マネージャウィンドウ内にある「マクロ (Macros)」タブをクリックしてそのネットワークマクロを選択すると、ウィザードで入力した情報がそのマクロに含まれているオプションの値として挿入されていることを確認できます。
   ネットワークウィザードは、空のネットワークテーブルを作成します。このテーブルはウィンドウの左側の区画に表示されます。このネットワークのアドレスを追加してからそのネットワークの IP アドレスを DHCP で管理する必要があります。詳細については、 200 ページの「DHCP サービスへのアドレスの追加」を参照してください。

▼ DHCP ネットワークを追加する方法 (dhcpconfig)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. **DHCP** サーバーシステム上で次のコマンドを入力します。

   ```
   # /usr/sbin/dhcpconfig -N network_address
   ```

   network_address には、DHCP サービスに追加したいネットワークの IP アドレスを指定します。`-N` オプションと一緒に使用できるサブオプションについては、`dhcpconfig` のマニュアルページを参照してください。
   サブオプションを使用しない場合、`dhcpconfig` はネットワークファイルを使用して、ネットワークについて必要な情報を取得します。

3. ネットワーク上のクライアントがアドレスを取得できるように、ネットワークの **IP** アドレスを追加します。

   200 ページの「DHCP サービスへのアドレスの追加」を参照してください。

DHCP ネットワークの構成の変更

ネットワークを DHCP サービスに追加した後に、最初に入力した構成情報を変更するには、ネットワークのクライアントに情報を渡すために使用されるネットワークマクロを変更します。

次に、DHCP マネージャの「マクロ (Macros)」タブを示します。

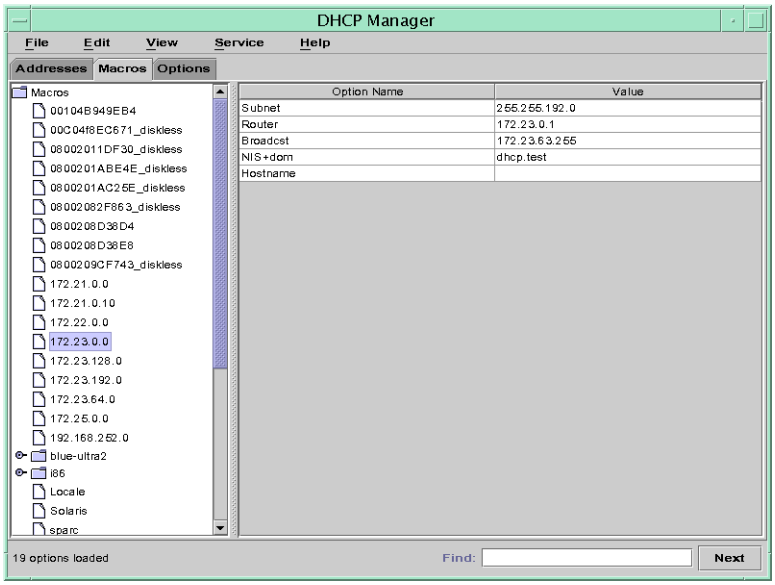


図 **10–6** DHCP マネージャの「マクロ (Macros)」タブ

▼ DHCP ネットワークの構成を変更する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。

   この DHCP サーバーについて定義されたすべてのマクロが左側の区画にリストされます。

2. 構成を変更したいネットワークと名前が一致するネットワークマクロを選択します。

ネットワークマクロ名は、そのネットワークの IP アドレスです。

3. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
「マクロの属性 (Macro Properties)」ダイアログボックスに、マクロに含まれるオプションが示されます。

4. 変更するオプションを選択します。
オプションの名前と値は、ダイアログボックス上部のテキストフィールドに表示されます。

5. そのオプションの新しい値を入力して、「変更 **(Modify)**」をクリックします。
ダイアログボックスで「選択 (Select)」をクリックして、オプションを追加することもできます。マクロの変更についての詳細は、214 ページの「DHCP マクロの変更」を参照してください。

6. 「**DHCP** サーバーに変更を通知する **(Notify DHCP Server of Change)**」を選択して、「了解 **(OK)**」をクリックします。
この変更は `dhcptab` に対して行われます。DHCP サーバーは `dhcptab` を再読み込みするようにシグナルを受け、この変更を有効にします。

▼ DHCP ネットワークの構成を変更する方法 (`dhtadm`)

1. ネットワークのすべてのクライアントに関する情報を含むマクロを特定します。
ネットワークマクロの名前は、ネットワークの IP アドレスと一致します。
この情報が含まれているマクロがわからない場合、`dhtadm -P` コマンドを使用すると、`dhcptab` データベースを表示して、すべてのマクロを表示できます。

2. 次の書式でコマンドを入力して、変更したいオプションの値を変更します。

```
# dhtadm -M -m macro-name -e 'symbol=value'
```

たとえば、`10.25.62.0` のマクロのリース期間を 57600 秒に変更し、NIS ドメインを `sem.example.com` に変更するには、次のコマンドを入力します。

```
# dhtadm -M -m 10.25.62.0 -e 'LeaseTim=57600'
# dhtadm -M -m 10.25.62.0 -e 'NISdmain=sem.example.com'
```

3. スーパーユーザーとして次のコマンドを入力し、**DHCP** デーモンが **`dhcptab`** を再読み込みするようにします。

```
# pkill -HUP in.dhcpd
```

DHCP ネットワークの削除

DHCP マネージャによって、複数のネットワークを同時に削除することができます。削除するネットワークにある DHCP に管理された IP アドレスに関連するホストテーブルのエントリを自動的に削除するオプションもあります。次に、DHCP マネージャの「ネットワークの削除 (Delete Networks)」ダイアログボックスを示します。

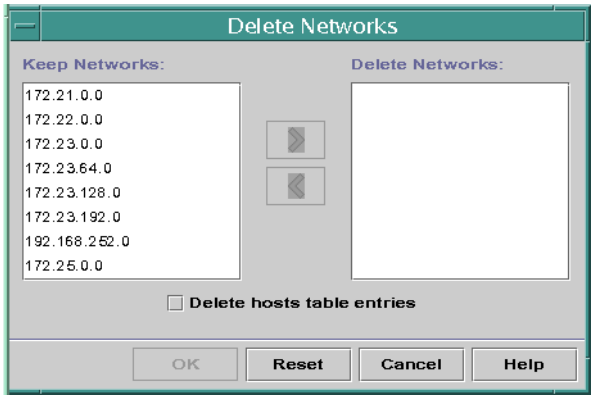


図 **10–7** 「ネットワークの削除 (Delete Networks)」ダイアログボックス

`pntadm` コマンドを使用する場合、ネットワークからそれぞれの IP アドレスのエントリを削除してからそのネットワークを削除する必要があります。一度に 1 つのネットワークだけを削除できます。

▼ DHCP ネットワークを削除する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. 「編集 **(Edit)**」メニューから「ネットワークの削除 **(Delete Networks)**」を選択します。

   「ネットワークの削除 (Delete Networks)」ダイアログボックスが開きます。
3. 「保持するネットワーク **(Keep Networks)**」リストで、削除したいネットワークを選択します。

   `Control` キーを押しながらマウスをクリックすると、複数のネットワークを選択できます。また、Shift キーを押しながらクリックすると、一定範囲のネットワークを選択できます。
4. 右矢印ボタンをクリックして、選択したネットワークを「ネットワークの削除 **(Delete Networks)**」リストに移動します。

5. このネットワークの **DHCP** が管理するアドレスに関するホストテーブルエントリを削除したい場合は、「ホストテーブルエントリも削除 **(Delete Host Table Entries)**」を選択します。

   この手順だけでは、これらのアドレスに関する DNS サーバー上のホスト登録は削除されません。この手順は、ローカルのネームサービスだけに影響します。

6. 「了解 **(OK)**」をクリックします。

▼ DHCP ネットワークを削除する方法 (pntadm)

この手順は、ネットワーク上のアドレスを削除してからそのネットワークを削除します。この手順によって、`hosts` ファイルまたはデータベースからホスト名が確実に削除されます。

1. サーバーシステム上で、スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。

2. 次の書式でコマンドを入力して、ネームサービスから **IP** アドレスとそのホスト名を削除します。

   # pntadm -D -y *IP-address*

   たとえば、アドレス 10.25.52.1 を削除するには、次のコマンドを入力します。

   ```
   # pntadm -D -y 10.25.52.1
   ```

   この `-y` オプションは、ホスト名の削除を指定します。

3. ネットワークのアドレスごとに **pntadm -D -y** コマンドを繰り返し入力します。

   多くのアドレスを削除する場合は、スクリプトを作成すると便利です。

4. すべてのアドレスを削除してから、次のコマンドを入力して、**DHCP** サービスからネットワークを削除します。

   # pntadm -R *network-IP-address*

   たとえば、アドレス 10.25.52.0 を削除するには、次のコマンドを入力します。

   ```
   # pntadm -R 10.25.52.0
   ```

   `pntadm` を使用する方法については、`pntadm` のマニュアルページを参照してください。

DHCP サービスによる BOOTP クライアントのサポート (作業マップ)

DHCP サーバー上で BOOTP クライアントをサポートするには、DHCP サーバーを BOOTP 互換に設定する必要があります。BOOTP 互換の設定内容に応じて、BOOTP クライアントを DHCP サーバーのネットワークテーブルに登録したり、BOOTP クライアントの割り当てに関するいくつかの IP アドレスを予約したりすることができます。

注 – BOOTP アドレスは常時割り当てされます。それらのアドレスを常時リースに明示的に割り当てたかどうかは関係ありません。

次の作業マップに、BOOTP クライアントをサポートするために実行する必要がある作業とその手順を示します。

タスク	説明	参照先
自動 BOOTP サポートの設定	DHCP に管理されたネットワークや、リレーエージェントによって DHCP に管理されたネットワークに接続されたネットワークにあるすべての BOOTP クライアントに IP アドレスを提供する そのため、BOOTP クライアントでアドレスを排他的に使用するためにアドレスのプールを予約する必要がある。このオプションは、サーバーが多くの BOOTP クライアントをサポートする必要がある場合に便利。	194 ページの「すべての BOOTP クライアントのサポートを設定する方法 (DHCP マネージャ)」
手動 BOOTP サポートの設定	DHCP サービスを使用して手動で登録された BOOTP クライアントだけに IP アドレスを提供する そのため、BOOTP クライアント用に指定された特定の IP アドレスにクライアントの ID を結びつける必要がある。このオプションは、BOOTP クライアントが少数の場合や、サーバーを使用できる BOOTP クライアントを制限したい場合に便利。	194 ページの「登録された BOOTP クライアントのサポートを設定する方法 (DHCP マネージャ)」

▼ すべての BOOTP クライアントのサポートを設定する方法 (DHCP マネージャ)

1. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。
   「サービスオプションの変更 (Modify Service Options)」ダイアログボックスが開きます。
2. このダイアログボックスの「**BOOTP** 互換 **(BOOTP Compatibility)**」セクションで、「自動 **(Automatic)**」を選択します。
3. 「サーバーの再起動 **(Restart Server)**」が選択されていない場合は、選択します。
4. 「了解 **(OK)**」をクリックします。
5. **DHCP** マネージャの「アドレス **(Addresses)**」タブを選択します。
6. **BOOTP** クライアント用に予約したいアドレスを選択します。
   最初のアドレスをクリックし、Shift キーを押しながら最後のアドレスをクリックして、一定範囲のアドレスを選択します。
   `Control` キーを押しながら各アドレスをクリックして、重複していない複数のアドレスを選択します。
7. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックスが開きます。
8. 「**BootP**」セクションで、「**BootP** クライアントだけにすべてのアドレスを割り当てる **(Assign All Addresses Only to BOOTP Clients)**」を選択します。
   残りのオプションは「現在の設定を維持 (Keep Current Settings)」に設定しておきます。
9. 「了解 **(OK)**」をクリックします。
   これで、すべての BOOTP クライアントがこの DHCP サーバーからアドレスを取得できるようになりました。

▼ 登録された BOOTP クライアントのサポートを設定する方法 (DHCP マネージャ)

1. 「サービス **(Service)**」メニューから「変更 **(Modify)**」を選択します。
   「サービスオプションの変更 (Modify Service Options)」ダイアログボックスが開きます。
2. このダイアログボックスの「**BOOTP** 互換 **(BOOTP Compatibility)**」セクションで、「手動 **(Manual)**」を選択します。
3. 「サーバーの再起動 **(Restart Server)**」が選択されていない場合は、選択します。

4. 「了解 **(OK)**」をクリックします。
5. **DHCP** マネージャの「アドレス **(Addresses)**」タブを選択します。
6. 特定の **BOOTP** クライアントに割り当てるアドレスを選択します。
7. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「アドレスの属性 (Address Properties)」ダイアログボックスが開きます。
8. 「リース **(Lease)**」タブを選択します。
9. 「クライアント **ID (Client ID)**」フィールドでクライアントの **ID** を入力します。
   Ethernet ネットワーク上で Solaris オペレーティング環境を実行している BOOTP クライアントの ID は、Ethernet のアドレス解決プロトコル (ARP) タイプ (01) にそのクライアントの 16 進 Ethernet アドレスから取り出された文字列が付いたものです。たとえば、Ethernet アドレス 8:0:20:94:12:1e を持つ BOOTP クライアントは、0108002094121E というクライアント ID を使用します。

   ヒント – Solaris クライアントシステム上のスーパーユーザーとして次のコマンドを入力すると、そのインタフェースに関する Ethernet アドレスを取得できます。

   ```
   ifconfig -a
   ```

   ---

10. 「予約 **(Reserved)**」を選択して、このクライアント用に **IP** アドレスを予約します。
11. 「**BOOTP** クライアントのみに割り当てる **(Assign Only to BOOTP Clients)**」を選択します。
12. 「了解 **(OK)**」をクリックします。
    「アドレス (Addresses)」タブでは、BOOTP は「状態 (Status)」フィールドに表示され、入力したクライアント ID は「クライアント ID (Client ID)」フィールドに表示されます。

DHCP サービスで IP アドレスを使用して作業する (作業マップ)

DHCP マネージャまたは `pntadm` コマンドを使用して、IP アドレスの追加、それらのアドレスの属性の変更、DHCP サービスからのアドレスの削除を実行できます。IP アドレスを使用した作業を始める前に、表 10–2 を参照して IP アドレスの属性を確認してください。この表を使用して、DHCP マネージャと `pntadm` を使用するための情報を知ることができます。

注 – この節では、pntadm コマンドを使用するための手順については説明しません。ただし、表 10–2 では、IP アドレスの追加と変更をする際に pntadm を使用して IP アドレスの属性を指定する例を示しています。pntadm についての詳細は、pntadm のマニュアルページも参照してください。

次の作業マップに、IP アドレスを追加、変更、削除する際に実行する必要がある作業とその手順を示します。

タスク	説明	参照先
単一または複数の IP アドレスを DHCP サービスに追加する	DHCP マネージャを使用して DHCP サービスですでに管理されているネットワークに IP アドレスを追加する	202 ページの「単一の IP アドレスを追加する方法 (DHCP マネージャ)」 202 ページの「既存の IP アドレスを複製する方法 (DHCP マネージャ)」 203 ページの「複数のアドレスを追加する方法 (DHCP マネージャ)」 203 ページの「アドレスを追加する方法 (pntadm)」
IP アドレスの属性を変更する	表 10–2 で説明している IP アドレスの属性を変更する	205 ページの「IP アドレスの属性を変更する方法 (DHCP マネージャ)」 206 ページの「IP アドレスの属性を変更する方法 (pntadm)」
DHCP サービスから IP アドレスを削除する	指定された IP アドレスを DHCP から使用できないように設定する	206 ページの「アドレスを使用不可に指定する方法 (DHCP マネージャ)」 207 ページの「アドレスを使用不可に指定する方法 (pntadm)」 208 ページの「DHCP サービスから IP アドレスを削除する方法 (DHCP マネージャ)」 208 ページの「DHCP サービスから IP アドレスを削除する方法 (pntadm)」
固定アドレスを DHCP クライアントに割り当てる	クライアントが要求するたびに同じ IP アドレスを受け取るようにクライアントを設定する	210 ページの「固定 IP アドレスを DHCP クライアントに割り当てる方法 (DHCP マネージャ)」 211 ページの「固定 IP アドレスを DHCP クライアントに割り当てる方法 (pntadm)」

次に、IP アドレスの属性を示します。

表 **10–2** IP アドレスの属性

プロパティ	説明	**pntadm** コマンドで指定する方法
ネットワークアドレス	作業の際に使用する IP アドレスを含むネットワークのアドレス このネットワークアドレスは、DHCP マネージャのアドレスタブにあるネットワークリストに表示される	ネットワークアドレスは、IP アドレスを作成、変更、または削除するために使用する pntadm コマンド行の最後の引数にする必要がある たとえば、ネットワーク 10.21.0.0 に IP アドレスを追加するには次のように入力する **pntadm -A** *ip-address options* **10.21.0.0**
IP アドレス	作成、変更、または削除する IP アドレス この IP アドレスは、DHCP マネージャのアドレスタブの最初の列に表示される	この IP アドレスを操作する場合、pntadm コマンドに必ず -A、-M、-D オプションを使用する たとえば、IP アドレス 10.21.5.12 を変更するには次のように入力する **pntadm -M 10.21.5.12** *options* **10.21.0.0**
クライアント名	ホストテーブルで IP アドレスに割り当てられるホスト名。この名前は、アドレスが作成されるときに、DHCP マネージャによって自動的に生成されることがある。単一のアドレスを作成する場合、その名前を入力することができる。	-h オプションを使用してクライアント名を指定する たとえば、10.21.5.12 にクライアント名 carrot12 を指定するには次のように入力する **pntadm -M 10.21.5.12 -h carrot12 10.21.0.0**
所有サーバー	IP アドレスを管理し、DHCP クライアントの IP アドレス割り当て要求への応答を担当する DHCP サーバー	-s オプションを使用して所有サーバー名を指定する たとえば、サーバー blue2 が 10.21.5.12 を所有するように指定するには、次のように入力する **pntadm -M 10.21.5.12 -s blue2 10.21.0.0**

表 **10–2** IP アドレスの属性　(続き)

プロパティ	説明	**pntadm** コマンドで指定する方法
構成マクロ	dhcptab からネットワーク構成オプションを取得するために DHCP サーバーが使用するマクロ。サーバーを構成してネットワークを追加すると、いくつかのマクロが自動的に作成される。マクロについての詳細は、129 ページの「マクロについて」を参照のこと。DHCP マネージャは、アドレスを生成するとき、サーバーマクロを作成し、各アドレス用の構成マクロとして割り当てる。	-m オプションを使用してマクロ名を指定する たとえば、サーバーマクロ blue2 をアドレス 10.21.5.12 に割り当てるには、次のように入力する **pntadm -M 10.21.5.12 -m blue2 10.21.0.0**
クライアント ID	DHCP サービス内で一意のテキスト文字列。 クライアント ID が 00 の場合、アドレスはどのクライアントにも割り当てられていない。 IP アドレスの属性を変更する際にクライアント ID を指定する場合は、そのアドレスを排他的に使用するために、そのアドレスをクライアントに手動で割り当てる クライアント ID は、DHCP クライアントのベンダーによって決定される。Solaris DHCP クライアント以外のクライアントを使用している場合は、その DHCP クライアントのマニュアルを参照すること	-i オプションを使用してクライアント ID を指定する たとえば、クライアント ID 08002094121E をアドレス 10.21.5.12 に割り当てるには、次のように入力する **pntadm -M 10.21.5.12 -i 0108002094121E 10.21.0.0**

表 **10–2** IP アドレスの属性　(続き)

プロパティ	説明	**pntadm** コマンドで指定する方法
	Solaris DHCP クライアントの場合、クライアント ID はクライアントの 16 進ハードウェアアドレスから取り出されたテキスト文字列。文字列の前には Ethernet の 01 のようなネットワークのタイプを表す ARP コードが付く。ARP コードは、Assigned Numbers 標準の ARP Parameters セクション内にある Internet Assigned Numbers Authority (IANA) によって割り当てられる (http://www.iana.com/numbers.html) たとえば、16 進 Ethernet アドレス 8:0:20:94:12:1e を持つ Solaris クライアントは、クライアント ID 0108002094121E を使用する。クライアントがアドレスを使用している場合、このクライアント ID は DHCP マネージャと `pntadm` で示される。 ヒント **:** Solaris クライアントシステム上のスーパーユーザーとして次のコマンドを入力すると、そのインタフェースに関する Ethernet アドレスを取得できる。 `ifconfig -a`	
予約済み	クライアント ID で示されたクライアントについて、アドレスが排他的に予約されることを指定する。DHCP サーバーはアドレスの返還を要求できない。このオプションを選択した場合、アドレスはクライアントに手動で割り当てる	`-f` オプションを使用して、アドレスの予約または手動を指定する たとえば、あるクライアントについて IP アドレス 10.21.5.12 の予約を指定するには、次のように入力する **`pntadm -M 10.21.5.12 -f MANUAL 10.21.0.0`**

表 **10–2** IP アドレスの属性　(続き)

プロパティ	説明	**pntadm** コマンドで指定する方法
リースのタイプとポリシー	クライアントでの IP アドレスの使用方法を DHCP でどのように管理するかを指定する。リースは、動的または常時。詳細については、149 ページの「動的リースタイプと常時リースタイプ」を参照のこと	`-f` オプションを使用して、アドレスが常時割り当てされるように指定する。デフォルトではアドレスは動的にリースされる たとえば、IP アドレス 10.21.5.12 を常時リース指定するには、次のように入力する **`pntadm -M 10.21.5.12 -f PERMANENT 10.21.0.0`**
リース有効期限	リースが期限切れになる日時。動的リースが指定された場合のみ利用できる。日付は *mm/dd/yyyy* 書式で指定する	`-e` を使用してリースの絶対的な有効期限を指定する たとえば、有効期限を 2002 年 1 月 1 日に指定するには、次のように入力する **`pntadm -M 10.21.5.12 -e 01/01/2002 10.21.0.0`**
BOOTP 設定	BOOTP クライアントに対してアドレスが予約されていることを指定する。BOOTP クライアントのサポートについての詳細は、193 ページの「DHCP サービスによる BOOTP クライアントのサポート (作業マップ)」を参照	`-f` を使用して BOOTP クライアント用のアドレスを予約する たとえば、IP アドレス 10.21.5.12 を BOOTP クライアント用に予約するには、次のように入力する **`pntadm -M 10.21.5.12 -f BOOTP 10.21.0.0`**
使用不可設定	アドレスがクライアントに割り当てられないようにする設定	`-f` を使用して、アドレスを使用不可に指定する たとえば、IP アドレス 10.21.5.12 を使用不能に指定するには、次のように入力する **`pntadm -M 10.21.5.12 -f UNUSABLE 10.21.0.0`**

DHCP サービスへのアドレスの追加

アドレスを追加する前に、それらのアドレスを所有するネットワークを DHCP サービスに追加する必要があります。ネットワークの追加についての詳細は、187 ページの「DHCP ネットワークの追加」を参照してください。

アドレスの追加は DHCP マネージャまたは `pntadm` で行うことができます。

すでに DHCP サービスによって管理されているネットワーク上では、DHCP マネージャを使用すると、次のような複数の方法でアドレスを追加できます。

- 単一の **IP** アドレスの追加 – 単一の新しい IP アドレスを DHCP の管理下に置く
- 既存の **IP** アドレスの複製 – DHCP が管理する既存の IP アドレスの属性をコピーし、新しい IP アドレスとクライアント名を与える
- 一定範囲の複数の **IP** アドレスの追加 – アドレスウィザードを使用して、一連の IP アドレスを DHCP の管理下に置く

次に、「アドレスの作成 (Create Address)」ダイアログボックスを示します。「アドレスの複製 (Duplicate Address)」ダイアログボックスは、テキストフィールドに既存のアドレスの値が表示されていることを除いて「アドレスの作成 (Create Address)」ダイアログボックスと同じです。

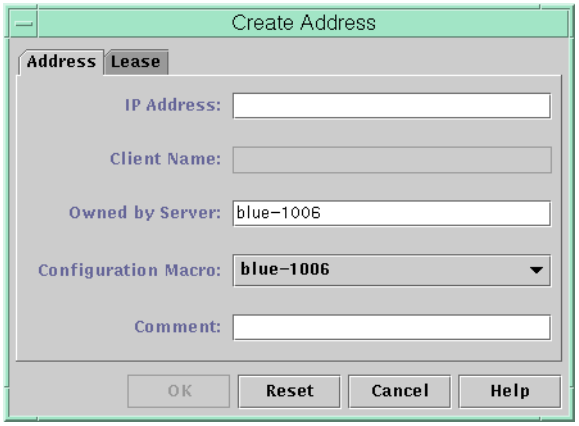


図 **10–8** 「アドレスの作成 (Create Address)」ダイアログボックス

次の図に、一定範囲の IP アドレスを追加するために使用するアドレスウィザードの最初のダイアログを示します。

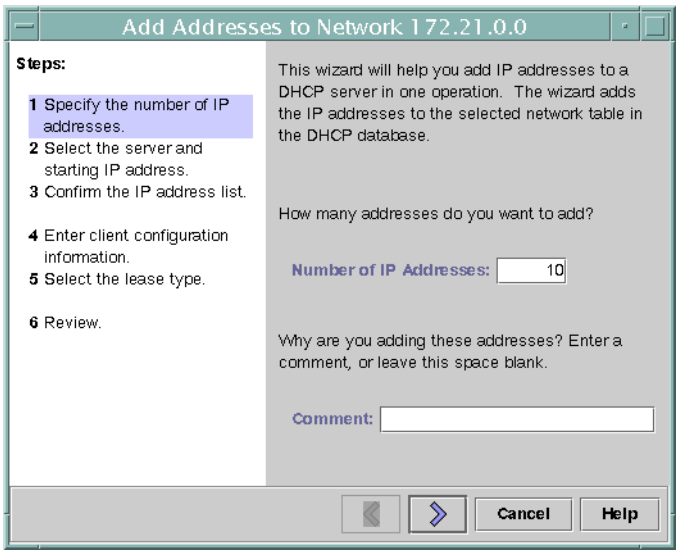


図 **10–9** アドレスウィザード

▼ 単一の IP アドレスを追加する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. 新しい **IP** アドレスを追加するネットワークを選択します。
3. 「編集 **(Edit)**」メニューから「作成 **(Create)**」を選択します。
   「アドレスの作成 (Create Address)」ダイアログボックスが開きます。
4. 「アドレス **(Address)**」と「リース **(Lease)**」タブで、値を選択または入力します。
   設定についての詳細は、表 10–2 を参照してください。
5. 「了解 **(OK)**」をクリックします。

▼ 既存の IP アドレスを複製する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. 新しい **IP** アドレスを配置するネットワークを選択します。

3. 属性の複製を作るアドレスを選択します。
4. 「編集 **(Edit)**」メニューから「複製 **(Duplicate)**」を選択します。
5. そのアドレスの **IP** アドレスとクライアント名を変更します。
   他のオプションのほとんどは同じままにしておく必要がありますが、必要に応じてそれらのオプションを変更することができます。
6. 「了解 **(OK)**」をクリックします。

▼ 複数のアドレスを追加する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. 新しい **IP** アドレスを追加するネットワークを選択します。
3. 「編集 **(Edit)**」メニューから「アドレスウィザード **(Address Wizard)**」を選択します。
   アドレスウィザードは、IP アドレスの属性値を入力するように要求します。 これらの属性についての詳細は、表 10–2 を参照してください。147 ページの「IP アドレスの管理に必要な選択 (作業マップ)」では、さらに詳細な情報が説明されています。
4. 情報を入力し終わったら、画面ごとに右矢印ボタンをクリックし、最後の画面で「完了 **(Finish)**」をクリックします。
   「アドレス (Addresses)」 タブに新規アドレスが更新されます。

▼ アドレスを追加する方法 (pntadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # **pntadm -A** *ip-address options network-address*

   pntadm -A と共に使用できるオプションについては、pntadm のマニュアルページを参照してください。 また、表 10–2 に、オプションを指定する pntadm コマンドの例をいくつか示しています。

注 – pntadm を使用すると、複数のアドレスを追加するスクリプトを作成できます。例については、例 12–1を参照してください。

DHCP サービスでの IP アドレスの変更

IP アドレスを DHCP サービスに追加すると、DHCP マネージャまたは `pntadm -M` コマンドを使用して、表 10–2 に示す属性を変更することができます。`pntadm -M` についての詳細は、`pntadm` のマニュアルページを参照してください。

次に、IP アドレスの属性を変更するときに使用する「アドレス属性 (Address Properties)」ダイアログボックスを示します。

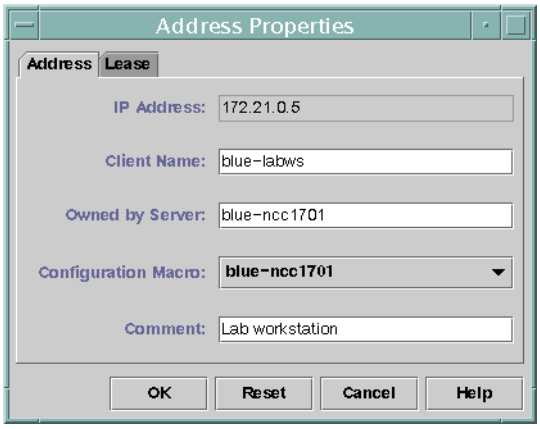


図 10–10 「アドレスの属性 (Address Properties)」ダイアログボックス

次に、複数の IP アドレスを変更するために使用する「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックスを示します。

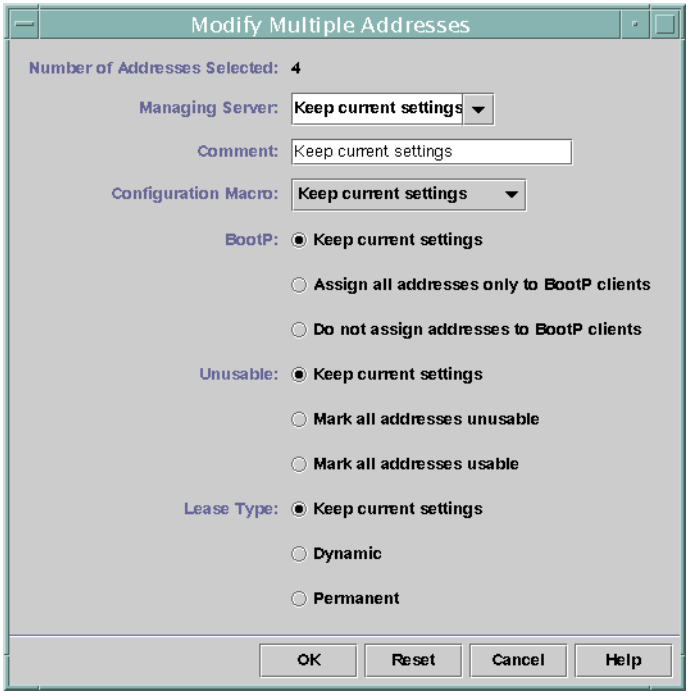


図 **10–11** 「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックス

▼ IP アドレスの属性を変更する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. その **IP** アドレスのネットワークを選択します。
3. 変更する **IP** アドレスを **1** つまたは複数選択します。
   複数のアドレスを変更する場合は、`Control` キーを押しながらマウスをクリックして、複数のアドレスを選択します。`Shift` キーを押しながらマウスをクリックして、一定範囲のアドレスを選択することもできます。
4. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「アドレスの変更 (Modify Addresses)」ダイアログボックスまたは「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックスが開きます。
5. 適切な属性を変更します。
   属性については、「ヘルプ (Help)」ボタンをクリックするか、表 10–2 を参照してください。

6. 「了解 **(OK)**」をクリックします。

▼ IP アドレスの属性を変更する方法 (pntadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # **pntadm -M** *ip-address options network-address*

   いくつかのオプションを pntadm コマンドと共に使用できます。これらのオプションについては、pntadm のマニュアルページを参照してください。

   表 10–2 に、オプションを指定する pntadm コマンドの例をいくつか示しています。

DHCP サービスからのアドレスの削除

特定の 1 つまたは複数のアドレスについて、DHCP サービスによる管理を停止したい場合があります。DHCP からアドレスを削除する方法は、その変更が一時的なものか永続的なものかによって異なります。

- アドレスを一時的に使用不可にするには、206 ページの「DHCP サービスで IP アドレスを使用不可にする」で説明しているように「アドレスの属性 (Address Properties)」ダイアログボックスでそれらのアドレスを使用不可に指定できます。
- 永続的に DHCP クライアントがアドレスを使用できないようにするには、207 ページの「DHCP サービスからの IP アドレスの削除」で説明しているように、DHCP ネットワークテーブルからそれらのアドレスを削除します。

DHCP サービスで IP アドレスを使用不可にする

-f UNUSABLE オプションを付けて pntadm -M コマンドを使用すると、アドレスを使用不可に指定できます。

DHCP マネージャでは、次の手順に示すとおり、図 10–10 の「アドレスの属性 (Address Properties)」ダイアログボックスを使用して各アドレスを指定でき、図 10–11 の「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックスを使用して複数のアドレスを指定できます。

▼ アドレスを使用不可に指定する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. その **IP** アドレスのネットワークを選択します。

3. 使用不可に指定したい **IP** アドレスを **1** つまたは複数選択します。
   複数のアドレスを使用不可に指定する場合は、`Control` キーを押しながらマウスをクリックして、複数のアドレスを選択します。`Shift` キーを押しながらマウスをクリックして、一定範囲のアドレスを選択することもできます。
4. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「アドレスの変更 (Modify Addresses)」ダイアログボックスまたは「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックスが開きます。
5. アドレスを **1** つ変更する場合は、「リース **(Lease)**」タブを選択します。
6. 「アドレスを使用しない **(Address is Unusable)**」を選択します。
   複数のアドレスを編集する場合は、「すべてのアドレスを使用しない (Mark All Addresses Unusable)」を選択します。
7. 「了解 **(OK)**」をクリックします。

▼ アドレスを使用不可に指定する方法 (pntadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # **pntadm -M** *ip-address* **-f UNUSABLE** *network-address*

   たとえば、アドレス `10.64.3.3` を使用不可に指定するには、次のように入力します。

```
pntadm -M 10.64.3.3 -f UNUSABLE 10.64.3.0
```

DHCP サービスからの IP アドレスの削除

IP アドレスを DHCP で管理したくない場合は、DHCP サービスデータベースからそのアドレスを削除する必要があります。`pntadm -D` コマンドまたは DHCP マネージャの「アドレスの削除 (Delete Address)」ダイアログボックスを使用できます。

次に、「アドレスの削除 (Delete Address)」ダイアログボックスを示します。

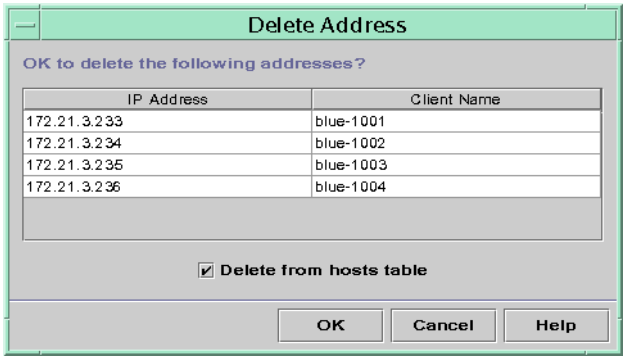


図 **10–12** 「アドレスの削除 (Delete Address)」ダイアログボックス

▼ DHCP サービスから IP アドレスを削除する方法 (DHCP マネージャ)

1. 「アドレス **(Addresses)**」タブを選択します。
2. その **IP** アドレスのネットワークを選択します。
3. 削除する **IP** アドレスを選択します。
   複数のアドレスを削除する場合は、`Control` キーを押しながらマウスをクリックして、複数のアドレスを選択します。`Shift` キーを押しながらマウスをクリックして、一定範囲のアドレスを選択することもできます。
4. 「編集 **(Edit)**」メニューから「削除 **(Delete)**」を選択します。
   「アドレスの削除 (Delete Address)」ダイアログボックスに、選択したアドレスがリストされるので、削除する内容を確認できます。
5. ホスト名をホストテーブルから削除したい場合、「ホストテーブルから削除 **(Delete From Hosts Table)**」を選択します。
   ホスト名が DHCP マネージャによって生成されたものである場合、ホストテーブルからその名前を削除できます。
6. 「了解 **(OK)**」をクリックします。

▼ DHCP サービスから IP アドレスを削除する方法 (`pntadm`)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。

2. 次の書式でコマンドを入力します。

   # **pntadm -D** *ip-address*

   -y オプションを指定した場合、ホスト名を保持しているネームサービスからホスト名が削除されます。

固定 IP アドレスを DHCP クライアントに設定する

Solaris DHCP サービスは、以前に DHCP を使用してアドレスを取得したクライアントに同じ IP アドレスを与えようとします。ただし、動的リースを使用している場合は除きます。

ネットワークにとって重要なルーター、NIS または NIS+、DNS サーバー、その他のホストは、IP アドレスの取得にあたってネットワークに依存するべきではないため、DHCP を使用するべきではありません。プリンタやファイルサーバーなどのクライアントも一定の IP アドレスを持つべきですが、DHCP を使用してネットワークの構成を受け取るように設定できます。

使用させたいアドレスにクライアントの ID を予約したり、手動で割り当てたりすると、クライアントがその構成を要求するたびに同じ IP アドレスを受け取るように設定できます。アドレスの使用を追跡しやすくするために、動的リースを使用するように予約済みアドレスを設定できます。あるいは、アドレスの使用を追跡する必要がない場合は、常時リースを使用するように設定できます。ただし、常時リースを取得すると、IP アドレスを解放したり DHCP リースネゴシエーションを再起動したりしない限り、クライアントはサーバーと連絡を取れず、更新された構成情報を取得できなくなるので、常時リースを使用したくない場合もあります。動的リースで予約済みアドレスを使用すべきクライアントの例としては、ディスクレスクライアントがあります。

`pntadm -M` コマンドまたは DHCP マネージャの「アドレスの属性 (Address Properties)」ダイアログボックスを使用することができます。

次に、リースを変更するために使用する「アドレスの属性 (Address Properties)」ダイアログボックスの「リース (Lease)」タブを示します。

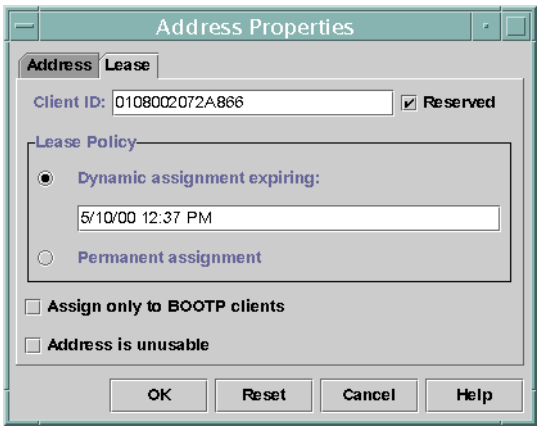


図 **10–13** 「アドレスの属性 (Address Properties)」の「リース (Lease)」タブ

▼ 固定 IP アドレスを DHCP クライアントに割り当てる方法 (DHCP マネージャ)

1. 固定 **IP** アドレスを割り当てたいクライアントのクライアント **ID** を決定します。
   クライアント ID を決定する方法については、表 10–2 の「クライアント ID」 の項目を参照してください。
2. **DHCP** マネージャの「アドレス **(Addresses)**」タブを選択します。
3. 適切なネットワークを選択します。
4. クライアントで使用したい **IP** アドレスをダブルクリックします。
   「アドレスの属性 (Address Properties)」ウィンドウが開きます。
5. 「リース **(Lease)**」タブを選択します。
6. 「クライアント **ID (Client ID)**」フィールドに、そのクライアントのハードウェアアドレスから決定したクライアント **ID** を入力します。
   詳細については、表 10–2 の「クライアント ID」の項目を参照してください。
7. 「予約 **(Reserved)**」オプションを選択して、その **IP** アドレスがサーバーによって返還を要求されないようにします。
8. 「アドレスの属性 **(Address Properties)**」ウィンドウの「リースポリシー **(Lease Policy)**」領域で、「動的 **(Dynamic)**」または「常時 **(Permanent)**」の割り当てを選択します。
   クライアントでリースを更新するネゴシエーションを行なって、アドレスが使用されている場合に追跡できるようにしたい場合は、「動的 (Dynamic)」を選択します。「予約 (Reserved)」を選択しているので、アドレスは動的リースを使用してい

ても再利用できません。このリースの有効期限は入力する必要がありません。DHCP サーバーがリース期間に基づいて有効期限を計算します。

「常時 (Permanent)」を選択した場合、トランザクションの記録を有効にしない限り、IP アドレスの使用を追跡できません。

▼ 固定 IP アドレスを DHCP クライアントに割り当てる方法 (pntadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # **pntadm -M** *ip-address* **-i** *client-id* **-f MANUAL+BOOTP** *network-address*

   クライアント ID を決定する方法については、表 10–2 の「クライアント ID」の項目を参照してください。

DHCP マクロを使用した作業 (作業マップ)

DHCP マクロは、DHCP オプションのコンテナです。Solaris DHCP サービスはマクロを使用して、クライアントに渡す必要があるオプションをまとめます。サーバーが構成されると、DHCP マネージャと `dhcpconfig` は、いくつかのマクロを自動的に作成します。マクロに関する背景情報については、129 ページの「マクロについて」を参照してください。デフォルトで作成されるマクロについての詳細は、第 9 章を参照してください。

ネットワークに変更が生じると、クライアントに渡す構成情報を変更しなければならない場合があります。この場合、DHCP マクロを使用して作業する必要があります。DHCP マクロは、表示、作成、変更、複製、削除することができます。

マクロを使用して作業するには、DHCP の標準オプションについて知っておく必要があります。DHCP の標準オプションについては、`dhcp_inittab` のマニュアルページを参照してください。

次の作業マップに、DHCP マクロを表示、作成、変更、および削除するときに役立つ作業のリストを示します。

作業	説明	参照先
DHCP マクロの表示	DHCP サーバーで定義されているすべてのマクロのリストを表示する	213 ページの「DHCP サーバー上で定義されたマクロを表示する方法 (DHCP マネージャ)」 214 ページの「DHCP サーバー上で定義されたマクロを表示する方法 (dhtadm)」
DHCP マクロの作成	DHCP クライアントをサポートする新しいマクロを追加する	219 ページの「DHCP マクロを作成する方法 (DHCP マネージャ)」 220 ページの「DHCP マクロを作成する方法 (dhtadm)」
DHCP クライアントに渡されるマクロ内の値の変更	既存のオプションの変更、マクロへのオプションの追加、マクロからのオプションの削除によって、マクロを変更する	215 ページの「DHCP マクロ内のオプションの値を変更する方法 (DHCP マネージャ)」 216 ページの「DHCP マクロ内のオプションの値を変更する方法 (dhtadm)」 216 ページの「DHCP マクロにオプションを追加する方法 (DHCP マネージャ)」 217 ページの「DHCP マクロにオプションを追加する方法 (dhtadm)」 217 ページの「DHCP マクロからオプションを削除する方法 (DHCP マネージャ)」 218 ページの「DHCP マクロからオプションを削除する方法 (dhtadm)」
DHCP マクロの削除	使用しない DHCP マクロを削除する	221 ページの「DHCP マクロを削除する方法 (DHCP マネージャ)」 221 ページの「DHCP マクロを削除する方法 (dhtadm)」

次に、DHCP マネージャウィンドウの「マクロ (Macros)」タブを示します。

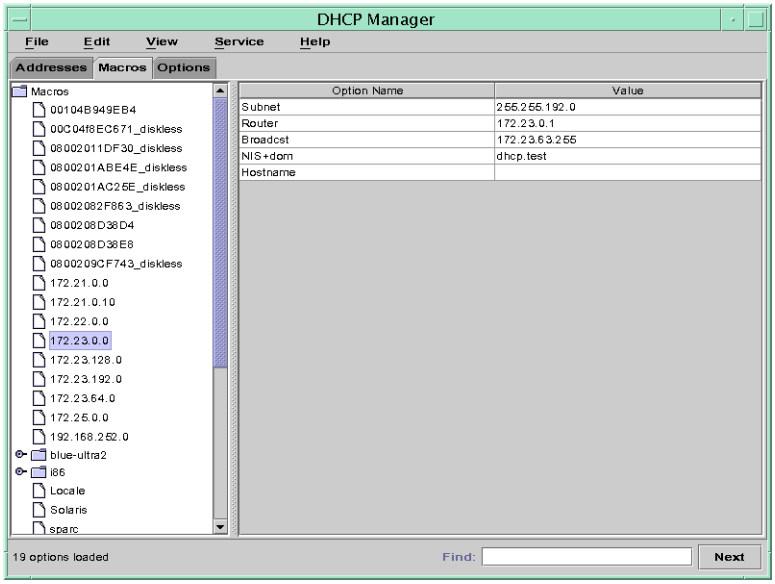


図 **10–14** DHCP マネージャの「マクロ (Macros)」タブ

▼ DHCP サーバー上で定義されたマクロを表示する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。

   ウィンドウ左側の「マクロ (Macros)」領域に、このサーバーで定義されたすべてのマクロがアルファベット順に表示されます。前にフォルダアイコンが付いたマクロには、他のマクロへの参照が含まれています。前にドキュメントアイコンが付いたマクロには、他のマクロへの参照が含まれていません。

2. マクロフォルダを開くには、フォルダアイコンの左にある開閉ウィジェットをクリックします。

   選択したマクロに含まれるマクロがリストされます。

3. マクロの内容を表示するには、マクロ名をクリックして、ウィンドウの右側の領域を確認します。

   オプションとそれらに割り当てられた値が表示されます。

▼ DHCP サーバー上で定義されたマクロを表示する方法 (dhtadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次のコマンドを入力します。

   ```
   # dhtadm -P
   ```

   このコマンドは、dhcptab の内容 (サーバー上で定義されたすべてのマクロとシンボルを含む) をフォーマットして標準出力に出力します。

DHCP マクロの変更

ネットワークの一部の設定が変更され、1 台または複数のクライアントにその変更を通知する必要がある場合、マクロを変更する必要があるかもしれません。たとえば、ルーターや NIS サーバーを追加したり、新しいサブネットを作成したり、リースポリシーの変更を決定したりした場合です。

マクロを変更する際には、変更、追加、または削除しようとしているパラメータに対応した DHCP オプションの名前を知っている必要があります。DHCP の標準オプションについては、DHCP マネージャのヘルプおよび dhcp_inittab のマニュアルページを参照してください。

dhtadm -M -m コマンドまたは DHCP マネージャを使用すると、マクロを変更することができます。dhtadm についての詳細は、dhtadm のマニュアルページを参照してください。

次に、DHCP マネージャの「マクロの属性 (Macro Properties)」ダイアログボックスを示します。

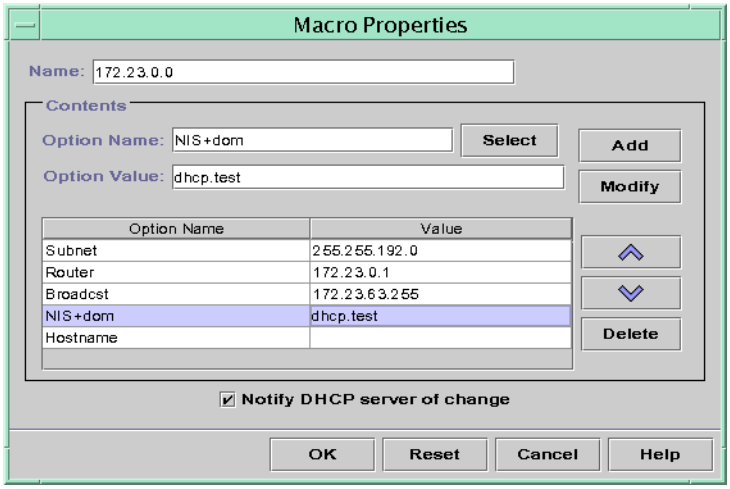


図 **10–15** 「マクロの属性 (Macro Properties)」ダイアログボックス

▼ DHCP マクロ内のオプションの値を変更する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。
2. 変更するマクロを選択します。
3. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「マクロの属性 (Macro Properties)」ダイアログボックスが開きます。
4. 「オプション **(Options)**」のテーブルで、変更するオプションを選択します。
   このオプションの名前と値は、「オプション名 (Option Name)」と「オプションの値 (Option Value)」のフィールドに表示されます。
5. 「オプションの値 **(Option Value)**」フィールドで、古い値を選択し、そのオプションの新しい値を入力します。
6. 「変更 **(Modify)**」をクリックします。
   新しい値がオプションテーブルに表示されます。
7. 「**DHCP** サーバーに変更を通知 **(Notify DHCP Server of Change)**」を選択します。
   この選択によって、DHCP サーバーは `dhcptab` を再読み込みし、「了解 (OK)」をクリックすると直ちに変更が適用されます。
8. 「了解 **(OK)**」をクリックします。

▼ DHCP マクロ内のオプションの値を変更する方法 (dhtadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # dhtadm -M -m *macroname* **-e 'option=value:option=value'**

   たとえば、マクロ `bluenote` 内のリース期間、および UTC との時間差を変更するには、次のコマンドを入力します。

   ```
   # dhtadm -M -m bluenote -e 'LeaseTim=43200:UTCOffst=28800'
   ```

▼ DHCP マクロにオプションを追加する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。
2. 変更するマクロを選択します。
3. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。

   「マクロの属性 (Macro Properties)」ダイアログボックスが開きます。
4. 「オプション名 **(Option Name)**」フィールドで、次のどちらかの方法を使用して、オプション名を指定します。

   a. 「オプション名 **(Option Name)**」フィールドの隣にある「選択 **(Select)**」ボタンをクリックして、マクロに追加したいオプションを選択します。

      「オプションの選択 (Select Option)」ダイアログボックスに、「標準 (Standard)」カテゴリのオプションの名前と説明がアルファベット順にリストされます。「標準 (Standard)」カテゴリ以外のオプションを追加したい場合は、「カテゴリ (Category)」リストを使用して、追加するカテゴリを選択してください。

      マクロカテゴリについての詳細は、129 ページの「マクロについて」を参照してください。

   b. 既存のマクロへの参照を新しいマクロに含めたい場合は、**`Include`** と入力してください。
5. 「オプションの値 **(Option Value)**」フィールドにオプションの値を入力します。

   オプション名を **`Include`** と入力した場合は、「オプションの値 (Option Value)」フィールドに既存のマクロの名前を指定する必要があります。
6. 「追加 **(Add)**」をクリックします。

   このオプションは、このマクロについて表示されたオプションのリストの一番下に追加されます。リスト内のオプションの位置を変更する場合は、そのオプションを

選択してリストの隣にある矢印キーをクリックし、オプションを上下に移動させます。

7. 「**DHCP** サーバーに変更を通知 **(Notify DHCP Server of Change)**」を選択します。
   この選択によって、DHCP サーバーは `dhcptab` を再読み込みし、「了解 (OK)」をクリックすると直ちに変更が適用されます。
8. 「了解 **(OK)**」をクリックします。

▼ DHCP マクロにオプションを追加する方法 (`dhtadm`)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   ```
   # dhtadm -M -m macroname -e 'option=value'
   ```

   たとえば、リースのネゴシエーションを行うオプションをマクロ `bluenote` に追加するには、次のコマンドを入力します。

   ```
   # dhtadm -M -m bluenote -e 'LeaseNeg=_NULL_VALUE'
   ```

   値を必要としないオプションの場合、オプションの値として `_NULL_VALUE` を使用してください。

▼ DHCP マクロからオプションを削除する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。
2. 変更するマクロを選択します。
3. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「マクロの属性 (Macro Properties)」ダイアログボックスが開きます。
4. マクロから削除するオプションを選択します。
5. 「削除 **(Delete)**」をクリックします。
   選択されたオプションが、このマクロに関するオプションのリストから削除されます。
6. 「**DHCP** サーバーに変更を通知 **(Notify DHCP Server of Change)**」を選択します。
   この選択によって、DHCP サーバーは `dhcptab` を再読み込みし、「了解 (OK)」をクリックすると直ちに変更が適用されます。

7. 「了解 **(OK)**」をクリックします。

▼ DHCP マクロからオプションを削除する方法 (dhtadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # dhtadm -M -m *macroname* **-e 'option='**

   たとえば、リースのネゴシエーションを行うオプションをマクロ `bluenote` から削除するには、次のコマンドを入力します。

   # dhtadm -M -m bluenote -e 'LeaseNeg='

   オプションに値を指定しなかった場合、オプションはマクロから削除されます。

DHCP マクロの作成

DHCP サービスに新しいマクロを追加して、特定の要求を持ったクライアントをサポートしたい場合があります。`dhtadm -A -m` コマンドまたは DHCP マネージャの「マクロの作成 (Create Macro)」ダイアログボックスを使用して、マクロを追加できます。`dhtadm` コマンドについての詳細は、`dhtadm` のマニュアルページを参照してください。

次に、DHCP マネージャの「マクロの作成 (Create Macro)」ダイアログボックスを示します。

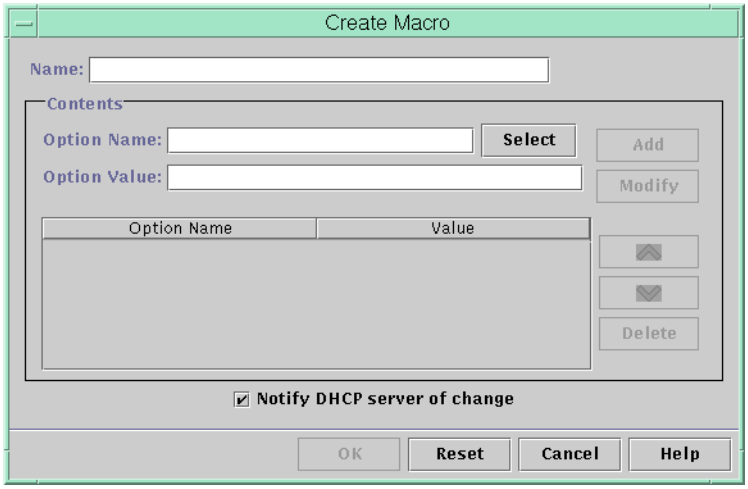


図 **10–16** 「マクロの作成 (Create Macro)」ダイアログボックス

▼ DHCP マクロを作成する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。
2. 「編集 **(Edit)**」メニューから「作成 **(Create)**」を選択します。
   「マクロの作成 (Create Macro)」ダイアログボックスが開きます。
3. そのマクロの名前 **(**固有の名前**)** を入力します。
   名前には 128 文字までの英数字を使用できます。 ベンダークラス識別子、ネットワークアドレス、またはクライアント ID に一致する名前を使用している場合は、そのマクロは適切なクライアントに対して自動的に処理されます。異なる名前を使用している場合は、そのマクロが特定の IP アドレスに割り当てられているか、または処理された別のマクロに含まれている場合のみ、そのマクロを処理することができます。詳細については、129 ページの「DHCP サーバーによるマクロ処理」を参照してください。
4. 「オプション名 **(Optioin Name)**」フィールドの隣にある「選択 **(Select)**」ボタンをクリックします。
   「オプションの選択 (Select Option)」ダイアログボックスに、「標準 (Standard)」カテゴリのオプションの名前と説明がアルファベット順にリストされます。
5. 「標準 **(Standard)**」カテゴリ以外のオプションを追加したい場合は、「カテゴリ **(Category)**」リストを使用して、追加するカテゴリを選択してください。
   オプションカテゴリについての詳細は、128 ページの「オプションについて」を参照してください。
6. マクロに追加したいオプションを選択して、「了解 **(OK)**」をクリックします。

「マクロの属性 (Macro Properties)」ダイアログボックスが、「オプション名 (Option Name)」フィールドに選択されたオプションを表示します。

7. 「オプションの値 **(Option Value)**」フィールドにオプションの値を入力します。

8. 「追加 **(Add)**」をクリックします。
   このオプションは、このマクロについて表示されたオプションのリストの一番下に追加されます。リスト内のオプションの位置を変更する場合は、そのオプションを選択してリストの隣にある矢印キーをクリックし、オプションを上下に移動させます。

9. マクロに追加するオプションごとに、手順 **6** から 手順 **8** までを繰り返します。

10. オプションの追加が終了したら、「**DHCP** サーバーに変更を通知 **(Notify DHCP Server of Change)**」を選択します。
    この選択によって、DHCP サーバーは `dhcptab` を再読み込みし、「了解 (OK)」をクリックすると直ちに変更が適用されます。

11. 「了解 **(OK)**」をクリックします。

▼ DHCP マクロを作成する方法 (`dhtadm`)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。

2. 次の書式でコマンドを入力します。

   ```
   # dhtadm -A -m macroname -d ':option=value:option=value:option=value:'
   ```

   `-d` への引数として指定するオプションと値のペアの数に制限はありません。引数はコロンで始まり、コロンで終了する必要があります。コロンはオプションと値の各ペアを区切ります。

   たとえば、マクロ `bluenote` を作成するには、次のコマンドを入力します。

   ```
   # dhtadm -A -m bluenote -d \
   ':Router=10.63.6.121:LeaseNeg=_NULL_VALUE:'DNSserv=10.63.28.12:'
   ```

   値を必要としないオプションの場合、オプションの値として `_NULL_VALUE` を使用してください。

DHCP マクロの削除

DHCP サービスからマクロを削除したい場合があります。たとえば、DHCP サービスからネットワークを削除する場合、関連するネットワークマクロも削除できます。

`dhtadm -D -m` コマンドまたは DHCP マネージャを使用して、マクロを削除することができます。

▼ DHCP マクロを削除する方法 (DHCP マネージャ)

1. 「マクロ **(Macros)**」タブを選択します。
2. 削除したいマクロを選択します。
   「マクロの削除 (Delete Macro)」ダイアログボックスは、指定したマクロの削除を確認するように求めます。
3. 「**DHCP** サーバーに変更を通知 **(Notify DHCP Server of Change)**」を選択します。
4. 「了解 **(OK)**」をクリックします。

▼ DHCP マクロを削除する方法 (dhtadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   ```
   # dhtadm -D -m macroname
   ```

   たとえば、マクロ `bluenote` を削除するには、次のコマンドを入力します。

   ```
   # dhtadm -D -m bluenote
   ```

DHCP オプションを使用した作業 (作業マップ)

オプションは、DHCP サーバーがクライアントに渡すネットワーク構成パラメータのキーワードです。Solaris DHCP サービスでは、作成、削除、または変更できるオプションは、Solaris DHCP サービスで標準オプションに指定されていないものだけです。そのため、初めて DHCP サービスを設定すると、サイト用のオプションを作成するまでは、DHCP マネージャの「オプション (Options)」タブは空です。

DHCP サーバー上でオプションを作成する場合、DHCP クライアント上でもそのオプションに関する情報を追加する必要があります。Solaris DHCP クライアントに対しては、`/etc/dhcp/inittab` ファイルを編集して、新しいオプションに関するエントリを追加する必要があります。このファイルについての詳細は、`dhcp_inittab` のマニュアルページを参照してください。

Solaris DHCP 以外のクライアントを使用している場合、新しいオプションまたはシンボルを追加する方法については、使用しているクライアント用のマニュアルを参照してください。Solaris DHCP でのオプションについての詳細は、128 ページの「オプションについて」を参照してください。

DHCP マネージャまたは `dhtadm` コマンドを使用して、オプションを作成、変更、削除できます。

注 – DHCP の文献では、オプションを「シンボル」と呼びます。`dhtadm` コマンドとマニュアルページでもオプションをシンボルと呼びます。

次の作業マップに、DHCP オプションを作成、変更、削除する際に必要な作業とその手順を示します。

作業	説明	参照先
DHCP オプションの作成	標準的な DHCP オプションで扱わない情報に関する新しいオプションを追加する	225 ページの「DHCP オプションを作成する方法 (DHCP マネージャ)」 225 ページの「DHCP オプションを作成する方法 (`dhtadm`)」 229 ページの「Solaris DHCP クライアントのオプション情報の変更」
DHCP オプションの変更	作成済みの DHCP オプションの属性を変更する	227 ページの「DHCP オプションの属性を変更する方法 (DHCP マネージャ)」 227 ページの「DHCP オプションの属性を変更する方法 (`dhtadm`)」
DHCP オプションの削除	作成済みの DHCP オプションを削除する	228 ページの「DHCP オプションを削除する方法 (DHCP マネージャ)」 228 ページの「DHCP オプションを削除する方法 （`dhtadm`)」

オプションを作成する前に、次の表に示すオプションの属性をよく理解しておく必要があります。

表 **10–3** DHCP オプションの属性

オプションの属性	説明
カテゴリ	オプションのカテゴリは、次のいずれかにする必要がある
	ベンダー – クライアントのベンダーのプラットフォームに固有のオプションであり、ハードウェアかソフトウェアになる
	サイト – サイトに固有のオプション
	拡張 – DHCP プロトコルに追加された比較的新しいオプションだが、まだ Solaris DHCP の標準オプションとして実装されていない
コード	コードは、オプションに割り当てる一意の番号。同じオプションカテゴリ内の他のオプションで、同じコードを使用することはできない。オプションカテゴリに対して適切なコードにする必要がある
	ベンダー – ベンダークラスごとに 1 から 254 のコード値
	サイト – 128 から 254 のコード値
	拡張 – 77 から 127 のコード値
データ型	データ型は、そのオプションの値として割り当てることができるデータの種類を指定する。有効なデータ型は次の通り
	ASCII – テキスト文字列値
	BOOLEAN – ブール型のデータ型に関連値はない。このオプションが存在すれば条件は真となり、存在しなければ偽となる。たとえば、標準オプションであり変更できない「Hostname」オプションはブール型。「Hostname」オプションがマクロに含まれている場合は、そのオプションは DHCP サーバーに、割り当てられたアドレスに関連するホスト名が存在するかどうかを調べるよう通知する
	IP – ドットで区切られた 10 進法形式 (*xxx.xxx.xxx.xxx*) の 1 つまたは複数のアドレス
	OCTET – 2 進データを翻訳されない 16 進 ASCII で表示したもの。たとえば、クライアント ID は、この 16 進形式のデータ型を使用する
	UNUMBER8, UNUMBER16, UNUMBER32, UNUMBER64, SNUMBER8, SNUMBER16, SNUMBER32, または SNUMBER64 の数値。 単語の先頭にある U または S は、数字が unsigned (符号なし) または signed (符号付き) であることを示す。単語の末尾にある数字 (8 から 64) は、数値のビット数を示す
データの単位数 (Granularity)	オプション値全体を表すために必要なデータ型の「インスタンス」の個数を指定する。たとえば、IP のデータ型でデータ単位数 2 の場合、オプション値には 2 つの IP アドレスが含まれる必要がある。
最大値	オプションについて指定可能な値の最大個数。前の例をもとにすると、最大値が 2、データ単位数が 2 で、データ型が IP の場合、オプション値には、最大 2 組の IP アドレスを含ことができる

表 **10–3** DHCP オプションの属性　(続き)

オプションの属性	説明
ベンダークライアントクラス	このオプションは、オプションカテゴリがベンダーの場合のみ利用できる。このオプションは、 オプションが関連するクライアントクラスを識別する。クラスはクライアントのマシンタイプやオペレーティングシステムを表す ASCII 文字列である (たとえば、`SUNW.Ultra5_10`)。このタイプのオプションを使用すると、あるクラスに属するすべてのクライアント (かつそのクラスに属するクライアント) だけに渡される構成パラメータを定義できる 複数のクライアントクラスを指定することができる。指定されたクライアントクラスと一致するクライアントクラス値の DHCP クライアントだけが、そのクラスに含まれるオプションを受け取る クライアントクラスは DHCP クライアントのベンダーによって決定される。Solaris クライアント以外の DHCP クライアントの場合、クライアントクラスについては、DHCP クライアントのベンダーのマニュアルを参照すること Solaris クライアントの場合、クライアント上で `uname -i` と入力して、 クライアントクラスを確認できる。ベンダークライアントクラスを指定するには、`uname` コマンドで返される文字列の中のすべてのカンマをピリオドに置き換える。たとえば、`uname -i` コマンドから文字列 `SUNW,Ultra5_10` が返される場合、ベンダークライアントクラスを `SUNW.Ultra5_10` として指定する

DHCP オプションの作成

渡す必要があるクライアント情報に対応するオプションが DHCP プロトコルにない場合は、オプションを作成できます。独自のオプションを作成するとき、Solaris DHCP で定義されているすべてのオプションのリストは、`dhcp_inittab` のマニュアルページに記載されています。

`dhtadm -A -s` コマンドまたは DHCP マネージャの「オプションの作成 (Create Option)」ダイアログボックスを使用すると、新しいオプションを作成することができます。

次に、DHCP マネージャの「オプションの作成 (Create Option)」ダイアログボックスを示します。

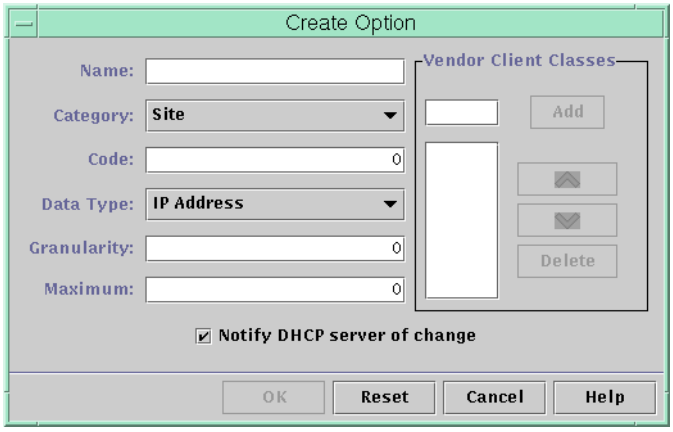


図 **10–17** 「オプションの作成 (Create Option)」ダイアログボックス

▼ DHCP オプションを作成する方法 (DHCP マネージャ)

1. 「オプション **(Options)**」タブを選択します。
2. 「編集 **(Edit)**」メニューから「作成 **(Create)**」を選択します。
   「オプションの作成 (Create Option)」ダイアログボックスが開きます。
3. 新しいオプションの略式記述名を入力します。
   この名前には、128 文字までの英数字 (空白文字を含む) を含めることができます。
4. ダイアログボックスの各設定について、値を入力または選択します。
   各設定についての詳細は、表 10–3 を参照してください。
5. オプションの作成が終わったら、「**DHCP** サーバーに変更を通知 **(Notify DHCP Server of Change)**」を選択します。
6. 「了解 **(OK)**」をクリックします。
   これでオプションをマクロに追加し、クライアントに渡すオプションに値を割り当てることができます。

▼ DHCP オプションを作成する方法 (`dhtadm`)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

```
# dhtadm -A -s option-name-d 'category,code,data-type,granularity,maximum'
```

次に、各引数について説明します。

option-name	128 文字以内の英数字文字列
category	`Site`、`Extend`、または `Vendor=`*list-of-classes*。*list-of-classes* は、オプションが適用されるベンダークライアントクラスの空白文字で区切られたリスト。ベンダークライアントクラスを決定する方法については、表 10–3 を参照のこと。
code	オプションカテゴリに適する数値 (表 10–3 を参照)
data-type	オプションと一緒に渡されるデータのタイプを示すキーワード (表 10–3 を参照)
granularity	負でない数値 (表 10–3 を参照)
maximum	負でない数値 (表 10–3 を参照)

次に例を 2 つ示します。

```
# dhtadm -A -s NewOpt -d 'Site,130,UNUMBER8,1,1'

# dhtadm -A -s NewServ -d 'Vendor=SUNW.Ultra-1 \
SUNW.SPARCstation10,200,IP,1,1'
```

DHCP オプションの変更

DHCP サービス用にオプションを独自に作成した場合、DHCP マネージャまたは `dhtadm` コマンドを使用すると、オプションの属性を変更できます。

`dhtadm -M -s` コマンドまたは DHCP マネージャの「オプションの属性 (Option Properties)」ダイアログボックスを使用して、オプションを変更できます。

Solaris DHCP クライアントのオプション情報を変更して、DHCP サービスに加えたのと同じ変更内容を反映する必要があります。229 ページの「Solaris DHCP クライアントのオプション情報の変更」を参照してください。

次に、DHCP マネージャの「オプションの属性 (Option Properties)」ダイアログボックスを示します。

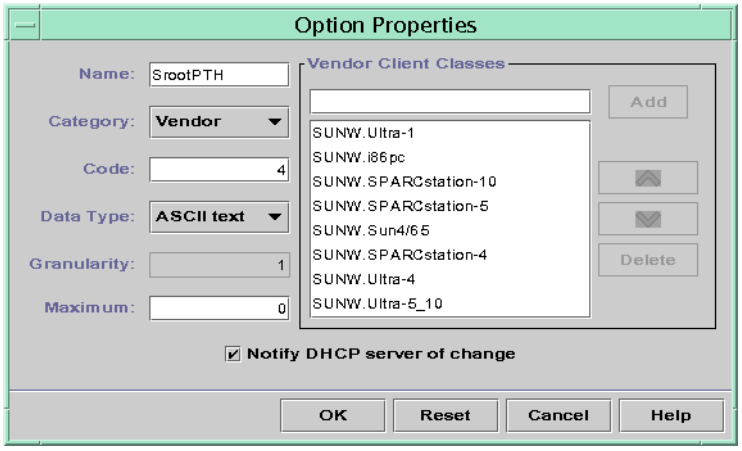


図 **10–18** 「オプションの属性 (Option Properties)」ダイアログボックス

▼ DHCP オプションの属性を変更する方法 (DHCP マネージャ)

1. 「オプション **(Options)**」タブを選択します。
2. 属性を変更するオプションを選択します。
3. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「オプションの属性 (Option Properties)」ダイアログボックスが開きます。
4. 必要に応じて属性を編集します。
   これらの属性についての詳細は、表 10–3 を参照してください。
5. オプションの変更が終わったら、「**DHCP** サーバーに変更を通知 **(Notify Server of Change)**」を選択します。
6. 「了解 **(OK)**」をクリックします。

▼ DHCP オプションの属性を変更する方法 (dhtadm)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

```
# dhtadm -M -s option-name-d 'category,code,data-type,granularity,maximum'
```

次に、各引数について説明します。

option-name	定義を変更するオプション名
category	`Site`、`Extend`、または `Vendor=`*list-of-classes*。*list-of-classes* は、オプションが適用されるベンダークライアントクラスの空白文字で区切られたリスト。たとえば、`SUNW.Ultra5_10 SUNW.Ultra-1 SUNWi86pc`
code	オプションカテゴリに適する数値 (表 10–3 を参照)
data-type	オプションと一緒に渡されるデータのタイプを示すキーワード (表 10–3 を参照)
granularity	負でない数値 (表 10–3 を参照)
maximum	負でない数値 (表 10–3 を参照)

変更する属性だけでなく、DHCP オプション属性すべてを `-d` スイッチで指定する必要があることに注意してください。

次に例を 2 つ示します。

```
# dhtadm -M -s NewOpt -d 'Site,135,UNUMBER8,1,1'
# dhtadm -M -s NewServ -d 'Vendor=SUNW.Ultra-1 \
SUNW.i86pc,200,IP,1,1'
```

DHCP オプションの削除

標準的な DHCP オプションは削除できません。しかし、独自のオプションを DHCP サービス用に定義した場合、DHCP マネージャまたは `dhtadm` コマンドを使用すると、それらのオプションを削除できます。

▼ DHCP オプションを削除する方法 (DHCP マネージャ)

1. 「オプション **(Options)**」タブを選択します。
2. 「編集 **(Edit)**」メニューから「削除 **(Delete)**」を選択します。
   「オプションの削除 (Delete Options)」ダイアログボックスが開きます。
3. 「了解 **(OK)**」をクリックして削除を確認します。

▼ DHCP オプションを削除する方法 （`dhtadm`)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。

2. 次の書式でコマンドを入力します。

```
# dhtadm -D -s option-name
```

Solaris DHCP クライアントのオプション情報の変更

新しい DHCP オプションを DHCP サーバーに追加する場合、各 DHCP クライアントのオプション情報に、補足エントリを追加する必要があります。Solaris DHCP クライアント以外の DHCP クライアントを使用している場合、オプションまたはシンボルを追加する方法については、そのクライアントのマニュアルを参照してください。

Solaris DHCP クライアントでは、`/etc/dhcp/inittab` ファイルを編集して、DHCP サーバーに追加するオプションごとにエントリを追加する必要があります。後にそのオプションをサーバー上で変更する場合、クライアントの `/etc/dhcp/inittab` ファイルのエントリも変更する必要があります。

`/etc/dhcp/inittab` ファイルの構文についての詳細は、`dhcp_inittab` のマニュアルページを参照してください。

注 – 以前のリリースの Solaris DHCP で `dhcptags` ファイルに DHCP オプションを追加していた場合、それらのオプションを `/etc/dhcp/inittab` ファイルに追加する必要があります。詳細については、280 ページの「DHCP のオプション」を参照してください。

DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)

DHCP を使用すると、ネットワーク上のクライアントシステムに Solaris オペレーティング環境をインストールできます。この機能を使用できるのは、Sun Enterprise Ultra システムと Solaris オペレーティング環境を実行するための要件を満たしている Intel システムだけです。

ディスクレスクライアントのサポートについては、236 ページの「リモートブートクライアントとディスクレスブートクライアントのサポート (作業マップ)」を参照してください。

次の作業マップに、クライアントが DHCP を使用してインストールパラメータを取得できるようにするために実行する必要がある作業を示します。

作業	説明	参照先
インストールサーバーの構成	Solaris サーバーを設定して、ネットワークから Solaris オペレーティング環境をインストールする必要があるクライアントをサポートする	『*Solaris 9* インストールガイド』の「ネットワークインストールの準備 (概要)」
DHCP を使用してネットワーク経由で Solaris をインストールできるようクライアントシステムを構成する	`add_install_client -d` を使用して、一定のマシンタイプのクライアントなど、任意のクラスのクライアントまたは特定のクライアント ID について、DHCP ネットワークインストールのサポートを追加する	Solaris DVD を使用する場合 『*Solaris 9* インストールガイド』の「ネットワークからインストールするシステムの追加」 Solaris CD を使用する場合 『*Solaris 9* インストールガイド』の「ネットワークからインストールするシステムの追加」 `add_install_client(1M)`
インストールパラメータについての DHCP オプションとそのオプションを含むマクロの作成	DHCP マネージャまたは `dhtadm` を使用して、DHCP サーバーがインストール情報をクライアントに渡すときに使用できる新しいベンダーオプションとマクロを作成する	230 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」

Solaris インストールパラメータ用の DHCP オプションとマクロの作成

インストールサーバー上で `add_install_client -d` スクリプトを使用してクライアントを追加するとき、そのスクリプトは DHCP 構成情報を標準出力にレポートします。この情報は、ネットワークインストール情報をクライアントに伝えるために必要なオプションとマクロを作成する際に使用できます。

ネットワークから Solaris のインストールが必要なクライアントをサポートするには、ベンダーカテゴリオプションを作成して、Solaris オペレーティング環境を適切にインストールするために必要な情報を渡す必要があります。次の表に、作成する必要があるオプションと、それらのオプションを作成するために必要な属性を示します。

表 **10–4** Solaris クライアント用にベンダーカテゴリオプションを作成するための値

名	コード	データ型	データ単位数 (Granularity)	最大個数 (Maximum)	ベンダークライアントクラス	説明
SrootOpt	1	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	クライアントのルートファイルシステム用の NFS マウントオプション
SrootIP4	2	IP アドレス	1	1	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	ルートサーバーの IP アドレス
SrootNM	3	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	ルートサーバーのホスト名
SrootPTH	4	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	ルートサーバーにあるクライアントのルートディレクトリへのパス
SswapIP4	5	IP アドレス	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	スワップサーバーの IP アドレス
SswapPTH	6	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	スワップサーバーにあるクライアントのスワップファイルへのパス
SbootFIL	7	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	クライアントのブートファイルへのパス
Stz	8	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	クライアントのタイムゾーン
SbootRS	9	NUMBER	2	1	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	カーネルを読み込む際にスタンドアロンの起動プログラムが使用する NFS 読み込みサイズ
SinstIP4	10	IP アドレス	1	1	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	JumpStart™ インストールサーバーの IP アドレス
SinstNM	11	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	インストールサーバーのホスト名

表 **10–4** Solaris クライアント用にベンダーカテゴリオプションを作成するための値　*(続き)*

名	コード	データ型	データ単位数 **(Granularity)**	最大個数 **(Maximum)**	ベンダークライアントクラス	説明
SinstPTH	12	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	インストールサーバーのインストールイメージへのパス
SsysidCF	13	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	*server:/path* という形式での、sysidcfg ファイルへのパス
SjumpsCF	14	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	*server:/path* という形式での、JumpStart 構成ファイルへのパス
Sterm	15	ASCII テキスト	1	0	SUNW.Ultra-1, SUNW.Ultra-30, SUNW.i86pc	端末タイプ

* ベンダークライアントクラスは、そのオプションを使用できるクライアントのクラスを決定します。ここに示されたベンダークライアントクラスは、提案に過ぎません。ネットワークからインストールする必要がある実際のクライアントについて、クライアントクラスを指定する必要があります。クライアントのベンダークライアントクラスを決定する方法については、表 10–3 を参照してください。

オプションが作成されている場合は、それらのオプションを含んだマクロを作成することができます。次に、クライアントについて Solaris のインストールをサポートするために作成することができる推奨マクロを示します。

表 **10–5** ネットワークインストールクライアントをサポートする推奨マクロ

マクロ名	含まれるオプションとマクロ
Solaris	SrootIP4、SrootNM、SinstIP4、SinstNM、Sterm
sparc	SrootPTH、SinstPTH
sun4u	Solaris マクロと sparc マクロ
i86pc	Solaris マクロ、SrootPTH、SinstPTH、SbootFIL
SUNW.i86pc*	i86pc マクロ
SUNW.Ultra-1*	sun4u マクロ、SbootFIL
SUNW.Ultra-30*	sun4u マクロ、SbootFIL マクロ
xxx.xxx.xxx.xxx (ネットワークアドレスマクロ)	BootSrvA オプションは既存のネットワークアドレスマクロに追加できます。BootSrvA の値は tftboot サーバーを示す必要があります。

表 **10–5** ネットワークインストールクライアントをサポートする推奨マクロ　(続き)

マクロ名	含まれるオプションとマクロ

* これらのマクロ名は、ネットワークからインストールするクライアントのベンダークライアントクラスと一致します。これらの名前は、ネットワーク上に持つことができるクライアントの例です。クライアントのベンダークライアントクラスの判定に関する情報については、表 10–3 を参照してください。

dhtadm コマンドまたは DHCP マネージャを使用して、これらのオプションとマクロを作成することができます。dhtadm を使用する場合は、dhtadm コマンドを繰り返し使用するスクリプトを作ってオプションとマクロを作成することをお勧めします。

233 ページの「dhtadm を使用してオプションとマクロを作成するスクリプトの作成」に、dhtadm コマンドを使用するスクリプトのサンプルを示します。DHCP マネージャを使用する場合は、235 ページの「DHCP マネージャを使用したインストールオプションとマクロの作成」を参照してください。

dhtadm を使用してオプションとマクロを作成するスクリプトの作成

例 10–1 の例を適用して、表 10–4 に示されたすべてのオプションといくつかの有用なマクロを作成する Korn シェルスクリプトを作成することができます。引用符に囲まれたすべての IP アドレスと値を、各ネットワークに関する適切な IP アドレス、サーバー名、パスに変更してください。また、Vendor= キーを編集して、使用するクライアントのクラスを示す必要もあります。add_install_client -d でレポートされる情報を使用して、スクリプトを各システムに適用するのに必要なデータを取得してください。

例 **10–1** ネットワークインストールをサポートするスクリプトの例

```
# Load the Solaris vendor specific options. We'll start out supporting
# the Ultra-1, Ultra-30, and i86 platforms. Changing -A to -M would replace
# the current values, rather than add them.
dhtadm -A -s SrootOpt -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,1,ASCII,1,0'
dhtadm -A -s SrootIP4 -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,2,IP,1,1'
dhtadm -A -s SrootNM -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,3,ASCII,1,0'
dhtadm -A -s SrootPTH -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,4,ASCII,1,0'
dhtadm -A -s SswapIP4 -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,5,IP,1,0'
dhtadm -A -s SswapPTH -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,6,ASCII,1,0'
dhtadm -A -s SbootFIL -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,7,ASCII,1,0'
dhtadm -A -s Stz -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,8,ASCII,1,0'
dhtadm -A -s SbootRS -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,9,NUMBER,2,1'
dhtadm -A -s SinstIP4 -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,10,IP,1,1'
dhtadm -A -s SinstNM -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,11,ASCII,1,0'
dhtadm -A -s SinstPTH -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,12,ASCII,1,0'
dhtadm -A -s SsysidCF -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,13,ASCII,1,0'
dhtadm -A -s SjumpsCF -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,14,ASCII,1,0'
dhtadm -A -s Sterm -d 'Vendor=SUNW.Ultra-1 SUNW.Ultra-30 SUNW.i86pc,15,ASCII,1,0'
# Load some useful Macro definitions
```

例 **10–1** ネットワークインストールをサポートするスクリプトの例　*(続き)*

```
# Define all Solaris-generic options under this macro named Solaris.
dhtadm -A -m Solaris -d ':SrootIP4=10.21.0.2:SrootNM="blue2":SinstIP4=10.21.0.2:\
SinstNM="red5":Sterm="xterm":'
# Define all sparc-platform specific options under this macro named sparc.
dhtadm -A -m sparc -d ':SrootPTH="/export/sparc/root":SinstPTH="/export/sparc/install":'
# Define all sun4u architecture-specific options under this macro named sun4u. (Includes
# Solaris and sparc macros.)
dhtadm -A -m sun4u -d ':Include=Solaris:Include=sparc:'
# Solaris on IA32-platform-specific parameters are under this macro named i86pc.
dhtadm -A -m i86pc -d \
':Include=Solaris:SrootPTH="/export/i86pc/root":SinstPTH="/export/i86pc/install"\
:SbootFIL="/platform/i86pc/kernel/unix":'
# Solaris on IA32 machines are identified by the "SUNW.i86pc" class. All
# clients identifying themselves as members of this class will see these
# parameters in the macro called SUNW.i86pc, which includes the i86pc macro.
dhtadm -A -m SUNW.i86pc -d ':Include=i86pc:'
# Ultra-1 platforms identify themselves as part of the "SUNW.Ultra-1" class.
# By default, we boot these machines in 32bit mode. All clients identifying
# themselves as members of this class will see these parameters.
dhtadm -A -m SUNW.Ultra-1 -d ':SbootFIL="/platform/sun4u/kernel/unix":Include=sun4u:'
# Ultra-30 platforms identify themselves as part of the "SUNW.Ultra-30" class.
# By default, we will boot these machines in 64bit mode. All clients
# identifying themselves as members of this class will see these parameters.
dhtadm -A -m SUNW.Ultra-30 -d ':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":\
Include=sun4u:'
# Add our boot server IP to each of the network macros for our topology served by our
# DHCP server. Our boot server happens to be the same machine running our DHCP server.
dhtadm -M -m 10.20.64.64 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.128 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.21.0.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.22.0.0    -e BootSrvA=10.21.0.2
# Make sure we return host names to our clients.
dhtadm -M -m DHCP-servername -e Hostname=_NULL_VALUE_
# The client with this MAC address is a diskless client. Override the root settings
# which at the network scope setup for Install with our client's root directory.
dhtadm -A -m 0800201AC25E -d \
':SrootIP4=10.23.128.2:SrootNM="orange-svr-2":SrootPTH="/export/root/10.23.128.12":'
```

スーパーユーザーとして、dhtadm をバッチモードで実行して、オプションとマクロを dhcptab に追加するスクリプトの名前を指定します。たとえば、スクリプトの名前が netinstalloptions の場合、次のコマンドを入力します。

dhtadm -B netinstalloptions

これが完了すると、Vendor= 文字列に示されたクライアントクラスを持つクライアントは、DHCP を使用して、ネットワークから Solaris をインストールするときに必要なパラメータを取得できます。

DHCP マネージャを使用したインストールオプションとマクロの作成

DHCP マネージャを使用すると、表 10–4 に示されたオプションと表 10–5 に示されたマクロを作成できます。

オプションとマクロの作成に使用するダイアログボックスについては、図 10–17 および図 10–16 を参照してください。

▼ Solaris のインストールをサポートするオプションを作成する方法 (DHCP マネージャ)

1. **DHCP** マネージャで「オプション **(Options)**」を選択します。
2. 「編集 **(Edit)**」メニューから「作成 **(Create)**」を選択します。
   「オプションの作成 (Create Option)」ダイアログボックスが開きます。
3. 最初のオプションのオプション名を入力し、そのオプションに値を入力します。
   表 10–4 を使用して、作成する必要があるオプションの名前と値を調べます。ベンダークライアントクラスは推奨値に過ぎないことに注意してください。DHCP サービスから Solaris インストールパラメータを取得する必要がある実際のクライアントのタイプを示すクラスを作成する必要があります。 クライアントのベンダークライアントクラスを決定する方法については、表 10–3 を参照してください。
4. すべての値を入力したら、「了解 **(OK)**」をクリックします。
5. 「オプション **(Options)**」タブで、今作成したオプションを選択します。
6. 「編集 **(Edit)**」メニューから「複製 **(Duplicate)**」を選択します。
   「オプションの複製 (Duplicate Option)」ダイアログボックスが開きます。
7. 新しく作成するオプションの名前を入力し、その他の値を適宜変更します。
   コード、データ型、データ単位数、最大個数は通常は変更する必要があります。これらの値については、表 10–4 を参照してください。
8. すべてのオプションを作成するまで、手順 **5** から 手順 **7** までを繰り返します。
   これで、次の手順の説明に従って、ネットワークインストールクライアントにオプションを渡すマクロを作成できます。

注 – これらのオプションはすでに Solaris クライアントの `/etc/dhcp/inittab` ファイルに含まれているので、わざわざ追加する必要はありません。

▼ Solaris のインストールをサポートするマクロを作成する方法 (DHCP マネージャ)

1. **DHCP** マネージャで「マクロ **(Macros)**」を選択します。
2. 「編集 **(Edit)**」メニューから「作成 **(Create)**」を選択します。
   「マクロの作成 (Create Macro)」ダイアログボックスが開きます。
3. マクロの名前を入力します。
   使用できるマクロ名については、表 10–5 を参照してください。
4. 「選択 **(Select)**」ボタンをクリックします。
   「オプションの選択 (Select Option)」ダイアログボックスが開きます。
5. 「カテゴリ **(Category)**」リストで「ベンダー **(Vendor)**」を選択します。
   作成したベンダーオプションがリストされます。
6. マクロに追加するオプションを選択して、「了解 **(OK)**」をクリックします。
7. オプションの値を入力します。
   オプションのデータ型については表 10–4 を参照してください。
   `add_install_client -d` がレポートする情報も参照してください。
8. すべてのオプションを追加するまで、手順 **6** から 手順 **7** までを繰り返します。
   別のマクロを追加するには、オプション名に **`Include`** と入力し、オプション値にそのマクロ名を入力します。
9. マクロが完成したら、「了解 **(OK)**」をクリックします。

リモートブートクライアントとディスクレスブートクライアントのサポート (作業マップ)

Solaris DHCP サービスは、オペレーティングシステムファイルを他のマシン (OS サーバー) からリモートでマウントする Solaris クライアントシステムをサポートしています。このようなクライアントを「ディスクレスクライアント」と呼びます。ディスクレスクライアントは、起動するたびにオペレーティングシステムファイルをホストするサーバーの名前と IP アドレスを取得して、これらのファイルからリモートで起動する必要があるので、永続的なリモートブートクライアントであると考えることができます。

各ディスクレスクライアントは、OS サーバー上に自分のルートパーティションを持っており、これらはクライアントのホスト名で共有されます。 つまり、DHCP サーバーは常に同じ IP アドレスをクライアントに返し、そのアドレスはネームサービス (DNS など) 内にある同じホスト名にマップされたままであることが必要です。このために、各ディスクレスクライアントには固定 IP アドレスが割り当てられる必要があります。

IP アドレスとホスト名に加えて、DHCP サーバーは、OS サーバー上のオペレーティングシステムファイルを見つけるために必要なすべての情報をディスクレスクライアントに提供できます。ただし、DHCP メッセージパケットで情報を渡すために使用できるオプションとマクロを作成する必要があります。

次の作業マップに、ディスクレスクライアント (あるいは、永続的なリモートブートクライアント) をサポートするために必要な作業を示します。

タスク	説明	参照先
Solaris サーバー上での OS サービスの設定	`smosservice` コマンドを使用して、クライアント用のオペレーティングシステムファイルを作成する	『*Solaris* のシステム管理 (基本編)』の「ディスクレスクライアントの管理 (手順)」 `smosservice` のマニュアルページも参照すること
ネットワークブートクライアントをサポートするための DHCP サービスの設定	DHCP マネージャまたは `dhtadm` を使用して、ブート情報をクライアントに渡すために DHCP サーバーが使用できる新しいベンダーオプションとマクロを作成する ネットワークインストールクライアント用のオプションをすでに作成している場合は、ディスクレスクライアントのベンダー クライアントタイプ用のマクロを作成するだけでよい	229 ページの「DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)」
ディスクレスクライアントへの予約済み IP アドレスの割り当て	DHCP マネージャまたは `pntadm` を使用して、アドレスがディスクレスクライアント用に予約されている (あるいは、手動で設定される) ことを指定する	209 ページの「固定 IP アドレスを DHCP クライアントに設定する」
OS サービス用のディスクレスクライアントの設定	`smdiskless` コマンドを使用して、クライアントごとにオペレーティングシステムサポートを OS サーバーに追加する。 クライアントごとに予約済みの IP アドレスを指定する	『*Solaris* のシステム管理 (基本編)』の「ディスクレスクライアントの管理 (手順)」 `smdiskless` のマニュアルページも参照すること

タスク	説明	参照先
ディスクレスクライアントへの予約済み IP アドレスの割り当て	DHCP マネージャまたは `pntadm` を使用して、アドレスがディスクレスクライアント用に予約されている (あるいは、手動で設定される) ことを指定する	209 ページの「固定 IP アドレスを DHCP クライアントに設定する」
OS サービス用のディスクレスクライアントの設定	`smdiskless` コマンドを使用して、クライアントごとにオペレーティングシステムサポートを OS サーバーに追加する。 クライアントごとに予約済みの IP アドレスを指定する	『*Solaris* のシステム管理 (基本編)』の「ディスクレスクライアントの管理 (手順)」 `smdiskless` のマニュアルページも参照すること

NIS+ クライアントとしての DHCP クライアントの設定

DHCP クライアントである Solaris システム上では NIS+ ネームサービスを使用できます。しかし、このためには NIS+ のセキュリティ拡張機能を部分的に犠牲にする (つまり、DES 資格を作成する) 必要があります。 DHCP を使用しない NIS+ クライアントを設定するときは、新しい NIS+ クライアントシステムごとに一意の DES 資格を NIS+ サーバー上にある `cred` テーブルに追加します。この方法はいくつかあります (`nisclient` スクリプトまたは `nisaddcred` コマンドを使用する方法など)

ただし、これらの方法では、資格を作成および保存するときに静的なホスト名を使用する必要があるので、DHCP を使用する NIS+ クライアントにこの方法は使用できません。NIS+ と DHCP を使用するクライアントを設定するときは、すべての DHCP クライアントのホスト名に使用できる同一の資格を作成する必要があります。この方法では、DHCP クライアントがどのような IP アドレス (および、関連するホスト名) を受け取っても、同じ DES 資格を使用できます。

注 – この作業を行う前に、NIS+ はセキュリティを考慮して設計されていること、ただしこの手順によってそのセキュリティが低下する (つまり、不特定の DHCP クライアントが NIS+ 資格を受け取ることができるようになる) ことを覚えておいてください。

次に、すべての DHCP ホスト名に使用できる同一の資格を作成する方法を示します。この手順を行うには、たとえば、ホスト名が DHCP サーバーによって生成されるときなどに、DHCP クライアントが使用するホスト名がわかる必要があります。

▼ NIS+ クライアントとして Solaris DHCP クライアントを設定する方法

NIS+ クライアントになる DHCP クライアントワークステーションは、NIS+ ドメイン内にある別の NIS+ クライアントワークステーションからコピーされた資格を使用する必要があります。この手順では、当該ワークステーションのみの資格が生成され、その資格は当該ワークステーションにログインしたスーパーユーザーだけに適用されます。当該ワークステーション (DHCP クライアント) にログインした他のユーザーは、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』の手順に従って作成された、独自の一意の資格を NIS+ サーバーに持つ必要があります。

1. **NIS+** サーバー上で次のコマンドを入力して、**NIS+** クライアント用の **cred** テーブルエントリを一時ファイルに書き込みます。

   ```
   # nisgrep nisplus-client-name cred.org_dir> /tmp/file
   ```

2. 一時ファイルの内容を見て、その資格をコピーし、そのコピーを使用して、**DHCP** クライアント用の資格を作成します。

   公開鍵と非公開鍵をコピーする必要があります。両者とも、コロンで区切られた数字と文字からなる長い文字列です。

3. 次のコマンドを入力して、**DHCP** クライアント用の資格を追加します。 一時ファイルから公開鍵と非公開鍵の情報をコピーします。

   ```
   # nistbladm -a cname=" dhcp-client-name@nisplus-domain" auth_type=DES \
   auth_name="unix.dhcp-client-name@nisplus-domain" \
   public_data=copied-public-data \
   private_data=copied-private-data
   ```

4. 各 **DHCP** クライアントシステム上で次のコマンドを入力して、**NIS+** クライアントファイルを **DHCP** クライアントシステムにリモートコピーします。

   ```
   # rcp nisplus-client-name:/var/nis/NIS_COLD_START /var/nis
   # rcp nisplus-client-name:/etc/.rootkey /etc
   # rcp nisplus-client-name:/etc/defaultdomain /etc
   ```

   「permission denied (アクセスが拒否された)」というメッセージを受信した場合、システムはリモートコピーを許可するように設定されていません。この場合はまず、一般ユーザーとして、中間地点にファイルをコピーします。次に、スーパーユーザーとして、DHCP クライアントシステム上の適切な場所にファイルをコピーします。

5. **DHCP** クライアントシステム上で次のコマンドを入力して、**NIS+** 用の正しいネームサービス切り替えファイルを使用します。

   ```
   # cp /etc/nsswitch.nisplus /etc/nsswitch.conf
   ```

6. **DHCP** クライアントシステムを再起動します。

   これで、DHCP クライアントシステムは NIS+ サービスを使用できます。

例 – NIS+ クライアントとしての Solaris DHCP クライアントの設定

次の例では、nisei というワークステーションが dev.example.net という NIS+ ドメイン内の NIS+ クライアントであり、dhow という DHCP クライアントを NIS+ クライアントにしようとしていると仮定します。

```
    (first log in as root on the NIS+ server)
# nisgrep nisei cred.org_dir> /tmp/nisei-cred
# cat /tmp/nisei-cred
nisei.dev.example.net.:DES:unix.nisei@dev.example.net:46199279911a84045b8e0
c76822179138173a20edbd8eab4:90f2e2bb6ffe7e3547346dda624ec4c7f0fe1d5f37e21cff63830
c05bc1c724b
# nistbladm -a cname="dhow@dev.example.net." \
auth_type=DES auth_name="unix.dhow@dev.example.net" \
public_data=46199279911a84045b8e0c76822179138173a20edbd8eab4 \
private_data=90f2e2bb6ffe7e3547346dda624ec4c7f0fe1d5f37e21cff63830\
c05bc1c724b
# rlogin dhow
    (log in as root on dhow)
# rcp nisei:/var/nis/NIS_COLD_START /var/nis
# rcp nisei:/etc/.rootkey /etc
# rcp nisei:/etc/defaultdomain /etc
# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
# reboot
```

これで、DHCP クライアントシステム dhow は NIS+ サービスを使用できます。

スクリプトによる資格の追加

多数の DHCP クライアントを NIS+ クライアントとして設定したい場合は、これらのエントリを cred テーブルに追加するためのスクリプトを作成します。次に、このようなスクリプトの例を示します。

例 **10–2** DHCP クライアントの資格を追加するスクリプトの例

```
#! /usr/bin/ksh
#
# Copyright (c) by Sun Microsystems, Inc. All rights reserved.
#
# Sample script for cloning a credential. Hosts file is already populated
# with entries of the form dhcp-[0-9][0-9][0-9]. The entry we're cloning
# is dhcp-001.
#
#
PUBLIC_DATA=6e72878d8dc095a8b5aea951733d6ea91b4ec59e136bd3b3
PRIVATE_DATA=3a86729b685e2b2320cd7e26d4f1519ee070a60620a93e48a8682c5031058df4
HOST="dhcp-"
DOMAIN="mydomain.example.com"

for
i in 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019
```

例 **10–2** DHCP クライアントの資格を追加するスクリプトの例　（続き）

```
do
     print - ${HOST}${i}
     #nistbladm -r [cname="${HOST}${i}.${DOMAIN}."]cred.org_dir
     nistbladm -a cname="${HOST}${i}.${DOMAIN}." \
         auth_type=DES auth_name="unix.${HOST}${i}@${DOMAIN}" \
         public_data=${PUBLIC_DATA} private_data=${PRIVATE_DTA} cred.org_Dir
done

exit 0
```

新しいデータストアへの変換

Solaris DHCP は、DHCP 構成データをあるデータストアから別のデータストアに変換するユーティリティを提供します。 新しいデータストアに変換する必要があるのは、たとえば、DHCP クライアントの数が増えて、より大きな性能や容量を DHCP サービスから受ける必要がある場合、あるいは DHCP サーバーの処理を複数のサーバーで分担したい場合などです。各タイプのデータストアの利点と欠点については、145 ページの「データストアの選択」を参照してください。

注 – DHCP サーバーシステム上で Solaris 8 7/01 以前の Solaris リリースからアップグレードした場合、Solaris のインストール後最初に Solaris DHCP 管理ツールを起動するときに、DHCP データテーブルを新しいデータストアに変換するように要求されます。Solaris 8 7/01 リリースでファイルと NIS+ の両方でデータストアのフォーマットが変更されているので、この変換は必須です。新しいデータストアに変換しない場合、DHCP サーバーは古いデータテーブルを読み込んで既存のクライアントのリース期間を延長します。古いデータテーブルを使用していると、新しい DHCP クライアントを登録したり、管理ツールを使用したりすることはできません。

変換ユーティリティは、Sun 提供のデータストアを Sun 以外のデータストアに変換する際にも便利です。 変換ユーティリティは、既存のデータストアのエントリを調べて、同じデータを持つ新しいエントリを新しいデータストアに追加します。 データストアのアクセスは各データストアごとに別々のモジュールに実装されています。したがって、変換ユーティリティは、各データストアが 1 つのモジュールを持っている場合、DHCP データをあるデータストアのフォーマットから別のデータストアのフォーマットに変換できます。 Sun 以外のデータストアをサポートするモジュールを作成する方法については、『*Solaris DHCP* サービス開発ガイド』を参照してください。

データストアの変換は、DHCP マネージャのデータストア変換ウィザードまたは `dhcpconfig -C` コマンドで実行できます。

次に、データストア変換ウィザードの初期ダイアログボックスを示します。

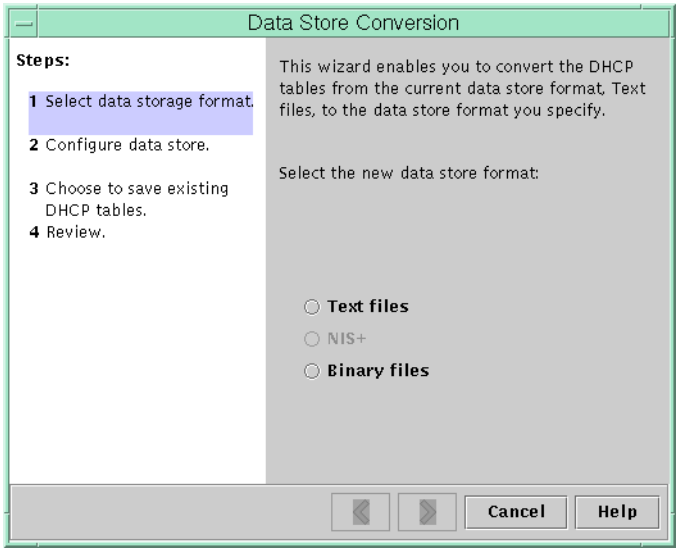


図 **10–19** データストア変換ウィザードのダイアログボックス

変換を開始する前、古いデータストアのテーブル (dhcptab テーブルとネットワークテーブル) を保存するかどうかを指定する必要があります。次に、変換ユーティリティは DHCP サーバーを停止し、データストアを変換し、変換が完了した後に、サーバーを再起動します。古いテーブルを保存すると指定しない場合、変換が完了した後、変換ユーティリティは古いテーブルを削除します。変換プロセスは時間がかかるので、進捗を示すメーターを表示してバックグラウンドで動作します。

▼ DHCP データストアを変換する方法 (DHCP マネージャ)

1. 「サービス **(Service)**」メニューから「データストアを変換 **(Convert Data Store)**」を選択します。

   データストア変換ウィザードが開きます。

2. ウィザードの質問に答えます。

   質問に対する回答がわからない場合は、「ヘルプ (Help)」をクリックすると、各ダイアログボックスについての詳細な情報を見ることができます。

▼ DHCP データストアを変換する方法 (dhcpconfig -C)

1. スーパーユーザーまたは **DHCP** 管理プロファイルに割り当てられたユーザーになります。
2. 次の書式でコマンドを入力します。

   # **/usr/sbin/dhcpconfig -C -r** *resource* **-p** *path*

   resource にはデータストア (SUNWbinfiles など) 、*path* にはデータへのパス (/var/dhcp など) を指定します。

   変換後も元のデータ (古いデータストア) を保存しておきたい場合は、-k オプションを指定してください。

DHCP サーバー間での構成データの移動 (作業マップ)

DHCP マネージャと dhcpconfig ユーティリティを使用すると、DHCP 構成データの一部またはすべてを、ある Solaris DHCP サーバーから別のサーバーに移動できます。ネットワーク全体と、アドレス、マクロ、および関連するオプションのすべてを移動することも、特定の IP アドレス、マクロ、およびオプションだけを移動することも可能です。また、データをサーバー上に保存しておくように指定すると、便利なマクロやオプションを (元のサーバーから削除せずに) コピーできます。

データを移動するのは、次のような場合です。

- サーバーを追加して DHCP の処理を分担させる
- DHCP サーバーのシステムを交換する
- データストアへのパスを変更する (同じデータストアを使用したままで)

次の作業マップに、DHCP 構成データを移動する場合に実行する必要がある手順を示します。

タスク	説明	参照先
1. 移動元のサーバーからデータをエクスポートする	移動先のサーバーに移動するデータを選択し、それをエクスポートしたデータのファイルを作成する	246 ページの「DHCP サーバーからデータをエクスポートする方法 (DHCP マネージャ)」 247 ページの「DHCP サーバーからデータをエクスポートする方法 (`dhcpconfig -x`)」
2. 移動先のサーバーにデータをインポートする	エクスポートしたデータを移動先の DHCP サーバーのデータストアにコピーする	246 ページの「DHCP サーバーにデータをインポートする方法 (DHCP マネージャ)」 248 ページの「DHCP サーバーにデータをインポートする方法 (`dhcpconfig -I`)」
3. インポートされたデータを新しいサーバー環境に合わせて変更する	サーバー固有の構成データを新しいサーバーの情報に一致するように変更する	246 ページの「インポートした DHCP データを変更する方法 (DHCP マネージャ)」 248 ページの「インポートした DHCP データを変更する方法 (`pntadm`、`dhtadm`)」

DHCP マネージャでは、「データをエクスポート(Export Data)」ウィザードと「データをインポート(Import Data)」ウィザードを使用して、データをあるサーバーから別のサーバーに移動し、「マクロ (Macros)」タブのマクロを変更します。次に各ウィザードの初期ダイアログボックスを示します。

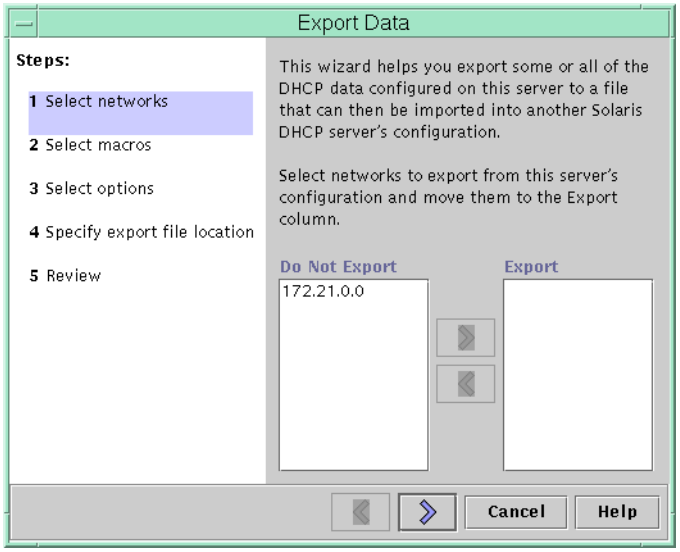


図 **10–20** 「データをエクスポート (Export Data)」ウィザードのダイアログボックス

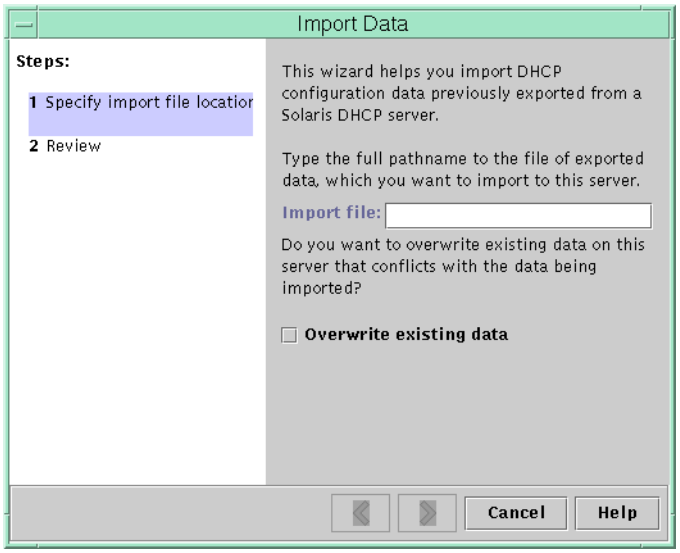


図 **10–21** 「データをインポート (Import Data)」ウィザードのダイアログボックス

▼ DHCP サーバーからデータをエクスポートする方法 (DHCP マネージャ)

1. データの移動 **(**またはコピー**)** 元のサーバー上で、スーパーユーザーになります。
2. 「サービス **(Service)**」メニューから「データをエクスポート **(Export Data)**」を選択します。

   図 10–20 に示すように「データをエクスポート (Export Data)」ウィザードが開きます。
3. ウィザードの質問に答えます。

   質問に対する回答がわからない場合は、「ヘルプ (Help)」をクリックすると、質問についての詳細な情報を見ることができます。
4. エクスポートするデータが入ったファイルを、データの移動先の **DHCP** サーバーがアクセス可能なファイルシステムに移動します。

   246 ページの「DHCP サーバーにデータをインポートする方法 (DHCP マネージャ)」に示す手順でデータをインポートしてください。

▼ DHCP サーバーにデータをインポートする方法 (DHCP マネージャ)

1. **DHCP** サーバーからエクスポートしたデータの移動先サーバー上で、スーパーユーザーになります。
2. **DHCP** マネージャで、「サービス **(Service)**」メニューから「データをインポート **(Import Data)**」を選択します。

   図 10–21 に示すように「データをインポート (Import Data)」ウィザードが開きます。
3. ウィザードの質問に答えます。

   質問に対する回答がわからない場合は、「ヘルプ (Help)」をクリックすると、質問についての詳細な情報を見ることができます。

▼ インポートした DHCP データを変更する方法 (DHCP マネージャ)

1. データをインポートしたサーバー上でスーパーユーザーになります。
2. インポートしたデータを調べて、変更する必要があるネットワーク固有情報を見つけます。

   ネットワークを移動した場合は、「アドレス (Addresses)」タブを開いて、移動 (インポート) したネットワーク内にあるアドレスの所有サーバーを変更する必要があります。また、「マクロ (Macros)」タブを開いて、マクロ内にある NIS、

NIS+、または DNS のドメイン名を変更する必要があります。

3. 「アドレス **(Addresses)**」タブを開いて、インポートしたネットワークを選択します。
4. すべてのアドレスを選択するには、最初のアドレスをクリックして、**Shift** キーを押したまま、最後のアドレスをクリックします。
5. 「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択します。
   「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックスが開きます。
6. 「管理サーバー **(Managing Server)**」プロンプトで、新しいサーバーの名前を選択します。
7. 「構成マクロ **(Configuration Macro)**」プロンプトで、当該ネットワーク上にあるすべてのクライアントに使用されるマクロを選択します。
8. 「了解 **(OK)**」をクリックします。
9. 「マクロ **(Macros)**」タブを開きます。
10. ウィンドウの下にある「検索 **(Find)**」機能を使用して、値を変更する必要があるオプションを見つけます。
    新しいサーバー上で変更する必要がありそうなオプションには、`DNSdmain`、`DNSserv`、`NISservs`、`NIS+serv`、`NISdmain` などがあります。
11. 変更する必要があるオプションを見つけたら、マクロ名を選択して、「編集 **(Edit)**」メニューから「属性 **(Properties)**」を選択し、その値を変更します。

▼ DHCP サーバーからデータをエクスポートする方法 (dhcpconfig -X)

1. データの移動 **(**またはコピー**)** 元のサーバー上で、スーパーユーザーになります。
2. 次の書式でコマンドを入力します。

   ```
   # /usr/sbin/dhcpconfig -X filename -a network-addresses -m macros -o options
   ```

   filename には、エクスポートするデータを圧縮して格納するための完全パス名を指定します。コマンドオプションにキーワード `ALL` を使用すると、すべてのネットワーク、マクロ、またはオプションをエクスポートできます。たとえば次のようにします。

   ```
   # /usr/sbin/dhcpconfig -X dhcp1065_data -a ALL -m ALL -o ALL
   ```

   あるいは、コンマで区切られたリストを使用して、特定のネットワークアドレス、マクロ、および構成オプションだけをエクスポートできます。たとえば次のようにします。

```
# /usr/sbin/dhcpconfig -X dhcp1065_data -a 10.63.0.0,10.62.0.0 \
-m 10.63.0.0,10.62.0.0,SUNW.Ultra-5_10 -o Sterm
```

dhcpconfig コマンドについての詳細は、dhcpconfig のマニュアルページを参照してください。

3. エクスポートするデータが入ったファイルを、データの移動先の **DHCP** サーバーがアクセス可能なファイルシステムに移動します。

248 ページの「DHCP サーバーにデータをインポートする方法 (dhcpconfig -I)」に示す手順でデータをインポートしてください。

▼ DHCP サーバーにデータをインポートする方法 (dhcpconfig -I)

1. データの移動先のサーバー上で、スーパーユーザーになります。
2. 次の書式でコマンドを入力します。

```
# /usr/sbin/dhcpconfig -I filename
```

filename には、エクスポートされたデータが入ったファイルの名前を指定します。

248 ページの「インポートした DHCP データを変更する方法 (pntadm、dhtadm)」に示す手順に従ってインポートしたデータを変更してください。

▼ インポートした DHCP データを変更する方法 (pntadm、dhtadm)

1. データをインポートしたサーバー上でスーパーユーザーになります。
2. ネットワークテーブルを調べて、変更する必要があるデータを見つけます。

   ネットワークを移動した場合は、pntadm -P *network_address* を使用して、移動したネットワークのネットワークテーブルを出力します。

3. **IP** アドレス情報は、**pntadm** コマンドを使用して変更します。

   インポートしたアドレスで使用される所有サーバーと構成マクロを変更する必要もあります。たとえば、アドレス 10.63.0.2 の所有サーバー (10.60.3.4) とマクロ (dhcpsrv-1060) を変更するには、次のコマンドを使用します。

```
pntadm -M 10.63.0.2 -s 10.60.3.4 -m dhcpsrv-1060 10.60.0.0
```

   アドレスが多数ある場合は、各アドレスを変更するコマンドが入ったスクリプトファイルを作成します。そして、そのスクリプトを pntadm -B コマンドで実行します。つまり、pntadm をバッチモードで実行します。 詳細については、pntadm のマニュアルページを参照してください。

4. **dhcptab** マクロを調べて、値を変更する必要があるオプションを見つけます。

   dhtadm -P を使用して、dhcptab 全体を出力します。そして、grep などのツールを使用して、変更する必要があるオプションまたは値を見つけます。

5. 必要に応じて、**dhtadm -M** コマンドを使用して、マクロで使われているオプションを変更します。

   たとえば、マクロ中の NIS、NIS+、または DNS のドメイン名やサーバー名を変更する必要もあります。たとえば、次のコマンドは、マクロ mymacro 内にある DNSdmain と DNSserv の値を変更します。

   dhtadm -M -m mymacro -e 'DNSserv=dnssrv2:DNSdmain=example.net'

第 11 章

DHCP の障害追跡 (リファレンス)

この章では、DHCP サーバーまたはクライアントを設定する際に検出される問題や、構成が完了した後に DHCP を使用する際の問題を解決する情報について説明します。

この章では、次の情報について説明します。

DHCP サーバーの問題の障害追跡

サーバーを構成する際に発生する問題は、次のカテゴリに分類されます。

- データとして NIS+ を使用している場合の NIS+
- IP アドレス割り当て

NIS+ の問題

DHCP データとして NIS+ を使用する場合に発生する問題は、次のカテゴリに分類されます。

- NIS+ をデータとして選択できない
- NIS+ が適切に構成されない
- 権限の不足と資格が原因の NIS+ アクセス問題

NIS+ をデータストアとして選択できない

NIS+ をデータストアとして選択しようとして、DHCP マネージャのデータストアの選択肢に NIS+ が含まれていなかったり、NIS+ のインストールと実行が確認できないというメッセージが `dhcpconfig` から返されたりすることがあります。これは、こ

のネットワークでは NIS+ が使用されている可能性はあるが、このサーバーには NIS+ が構成されていないことを意味します。NIS+ をデータとして選択するためには、サーバーマシンが NIS+ クライアントとして構成されている必要があります。

サーバーを NIS+ クライアントとして設定するためには、ドメインがすでに構成され、そのマスターサーバーが動作している必要があります。さらに、ドメインのテーブルのマスターサーバーがすでに作成され、ホストテーブルには新しいクライアントシステムのエントリ (DHCP サーバー) が存在している必要があります。『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』の「NIS+ クライアントマシンの設定」で、NIS+クライアントの構成についての詳細が説明されています。

NIS+ が適切に設定されない

DHCP で NIS+ が正常に使用できるようになっても、NIS+ を変更するとエラーになり、構成の問題が明らかになることがあります。表 11–1 を使用して、問題の原因を特定してください。

表 **11–1** NIS+ の設定問題

問題	情報の収集	解決方法
ルートオブジェクトが NIS+ ドメインに存在しない	次のコマンドを入力する。 `/usr/lib/nis/nisstat` ドメインの統計情報が表示される。ルートオブジェクトが存在しない場合は、統計情報は表示されない	『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を 参照して NIS+ ドメインを設定する
`passwd` と `publickey` の情報について NIS+ が使用されていない	次のコマンドを入力して、ネームサービススイッチ構成ファイルを表示する `cat /etc/nsswitch.conf` この「nisplus」キーワードに関する `passwd` と `publickey` の項目を確認する	ネームサービススイッチの構成について は、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照
ドメイン名が空である	次のコマンドを入力します。 `domainname` このコマンドによって空の文字列がリストされた場合は、このドメインについてドメイン名が設定されていない	データストアにローカルファイルを使用するか、あるいは、ネットワーク用に NIS+ ドメインを設定する。『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照
`NIS_COLD_START` ファイルが存在しない	サーバーシステムで次のコマンドを入力して、ファイルの存在を確認する `cat /var/nis/NIS_COLD_START`	データストアのローカルファイルを使用するか、あるいは、NIS+ クライアントを作成します。『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: FNS*、*NIS+* 編*)*』を参照

NIS+ アクセス問題

NIS+ のアクセス権に問題があると、DES 資格が適切でない、またはアクセス権が不十分なため NIS+ オブジェクトやテーブルを更新できないというエラーメッセージが表示されることがあります。表 11–2 を使用して、受け取った NIS+ エラーの原因を判定してください。

表 **11–2** NIS+ アクセス問題

問題	情報の収集	解決方法

NIS+ ドメインの `org_dir` オブジェクトに対する作成アクセス権が DHCP サーバーシステムにない	次のコマンドを入力します。
	`nisls -ld org_dir`
	アクセス権は `r---rmcdrmcdr---` といった形式でリストされる。これらのアクセス権はそれぞれ、未認証、所有者、グループ、その他に対応する。次にオブジェクトの所有者がリストされる
	通常、`org_dir` ディレクトリオブジェクトでは、所有者とグループにすべての権利 (読み取り、変更、作成、 削除) が与えられ、その他と未認証に読み取りアクセス権だけが与えられる
	DHCP サーバー名は、`org_dir` オブジェクトの所有者として、またはグループの主体として一覧表示される。このグループには作成アクセス権が必要。次のコマンドでグループをリストする
	`nisls -ldg org_dir`
		`nischmod` コマンドを使って `org_dir.` に対するアクセス権を変更する
		たとえば、グループに作成アクセス権を追加する場合は、次のコマンドを使用する
		`nischmod g+c org_dir`
		詳細は `nischmod(1)` のマニュアルページを参照

DHCP サーバーに、`org_dir` オブジェクトの下にテーブルを作成するアクセス権がない
通常は、サーバーシステムの主体名が `org_dir` オブジェクトの所有グループのメンバーでないか、所有グループが存在しないことを意味する
	次のコマンドを入力して所有グループ名を検索する
	`niscat -o org_dir`
	次のような行を探す
	`Group : "admin.example.com."`
	次のコマンドを使ってグループ内の主体名をリストする
	`nisgrpadm -l` *groupname*
		`nisgrpadm` コマンドを使ってサーバーシステムの名前をグループに追加する
		たとえば、サーバー名 `pacific` をグループ `admin.example.com` に追加するには、次のように入力する
		`nisgrpadm -a admin.example.com pacific.example.com`

表 **11–2** NIS+ アクセス問題　(続き)

問題	情報の収集	解決方法
	入力例 `nisgrpadm -l admin.example.com` サーバーシステムの名前がグループの明示的なメンバーとしてリストされるか、グループの暗黙的なメンバーとして含まれているはずである	詳細は、`nisgrpadm(1)` のマニュアルページを参照

DHCP サーバーが、NIS+ `cred` テーブルに有効なデータ暗号化規格 (DES) 資格を持っていない	これが問題である場合には、エラーメッセージは、ユーザーが NIS+ ネームサービスに DES 資格を持っていないことを示す	`nisaddcred` コマンドを使って DHCP サーバーシステムのセキュリティ資格を追加する
		次の例では、ドメイン `example.com` にあるシステム `mercury` についての DES 資格を追加する方法を示す
		`nisaddcred -`
		`punix.mercury@example.com \ -P`
		`mercury.example.com. DES`
		`example.com.`
		このコマンドは、暗号化された秘密鍵の生成に必要なスーパーユーザーのパスワードを要求する
		詳細は、`nisaddcred(1M)` のマニュアルページを参照

IP アドレス割り当てエラー

クライアントが IP アドレスを取得または確認しようとすると、次の問題が `syslog` やサーバーデバッグ出力に書き込まれることがあります。

表 **11–3** IP アドレスの割り当てとリースに関する問題

エラーメッセージ	説明	解決方法
`There is no` *n.n.n.n* `dhcp-network table for DHCP client's network.`	クライアントが特定の IP アドレスを要求しているか、現在の IP アドレスのリースを延長しようとしているが、DHCP サーバーはそのアドレスに対する DHCP ネットワークテーブルを見つけることができない	DHCP ネットワークテーブルが誤って削除されている場合がある。DHCP マネージャまたは `dhcpconfig` を使ってネットワークを再び追加すれば、ネットワークテーブルを再作成できる

表 **11–3** IP アドレスの割り当てとリースに関する問題　*(続き)*

エラーメッセージ	説明	解決方法
`ICMP ECHO reply to OFFER candidate: n.n.n.n, disabling`	DHCP クライアントに提供されようとしている IP アドレスがすでに使用されている。複数の DHCP サーバーがこのアドレスを所有しているか、DHCP ネットワーク以外のクライアント用にアドレスが手動で構成されていると、この状態になることがある	そのアドレスの適正な所有権を判定し、DHCP サーバーデータベースか、ホストのネットワーク設定を訂正する
`ICMP ECHO reply to OFFER candidate: n.n.n.n. No corresponding dhcp network record.`	DHCP クライアントに提供されようとしている IP アドレスのレコードがネットワークテーブルにない。IP アドレスが選択された後で、かつ重複アドレスチェックが完了する前に、その IP アドレスレコードが DHCP ネットワークテーブルから削除されると、この状態になることがある	DHCP マネージャまたは `pntadm` を使って DHCP ネットワークテーブルを表示する。IP アドレスのレコードがない場合は、DHCP マネージャ (「アドレス (Addresses)」タブの「編集 (Edit)」メニューから「作成 (Create)」を選択) または `pntadm` を使ってレコードを作成する
`DHCP network record for n.n.n.n is unavailable, ignoring request.`	要求された IP アドレスのレコードは DHCP ネットワークテーブルに存在しないので、サーバーが要求をドロップする	DHCP マネージャまたは `pntadm` を使って DHCP ネットワークテーブルを表示する。IP アドレスのレコードがない場合は、DHCP マネージャ (「アドレス (Addresses)」タブの「編集 (Edit)」メニューから「作成 (Create)」を選択) または `pntadm` を使ってレコードを作成する
n.n.n.n `currently marked as unusable.`	ネットワークテーブルで使用不可能に指定されているため、要求された IP アドレスを提供できない	DHCP マネージャまたは `pntadm` を使って、アドレスを使用可能にする
n.n.n.n `was manually allocated. No dynamic address will be allocated.`	クライアントの ID は、手動で割り当てられたアドレスに割り当てられている。そのアドレスは使用不可能に指定されている。そのため、サーバーがこのクライアントに別のアドレスを割り当てることができない	DHCP マネージャまたは `pntadm` を使用して、そのアドレスを使用できるようにするか、またはそのクライアントに別のアドレスを手動で割り当てる
`Manual allocation (`*n.n.n.n, client ID* `has` *n* `other records. Should have 0.`	指定されたクライアント ID を持つクライアントに、複数の IP アドレスが手動で割り当てられている。割り当てられる IP アドレスは 1 つでなければならない。サーバーは、ネットワークテーブルにある、最後に手動で割り当てられたアドレスを選択する	DHCP マネージャまたは `pntadm` を使って IP アドレスを修正し、余分な手動割り当てを取り除く

表 **11–3** IP アドレスの割り当てとリースに関する問題　(続き)

エラーメッセージ	説明	解決方法
`No more IP addresses on` *n.n.n.n* `network.`	指定されたネットワーク上で DHCP が現在管理しているすべての IP アドレスは、すでに割り当てられている	DHCP マネージャまたは `pntadm` を使って、このネットワーク用に新しい IP アドレスを作成する
`Client:` *clientid* `lease on` *n.n.n.n* `expired.`	リースがネゴシエーション可能ではなく、有効期限が切れている	クライアントは、プロトコルを自動的に再起動して新しいリースを取得すべきである
`Offer expired for client:` *n.n.n.n*	サーバーがクライアントに IP アドレスを提供したが、クライアントの応答に時間がかかり過ぎ、このオファーは期限切れとなった	クライアントは、新たな検索メッセージを自動的に発行すべきである。これも期限切れとなった場合は、DHCP サーバーのキャッシュオファータイムアウトを増加させる。DHCP マネージャでは、「サービス (Service)」メニューから「変更 (Modify)」を選択する
`Client:` *clientid* `REQUEST is missing requested IP option.`	クライアントの要求が、提供された IP アドレスを指定しなかったため、DHCP サーバーはこの要求を無視した。クライアントが新しい DHCP プロトコル RFC 2131 に準拠していないと、この状態になることがある	クライアントのソフトウェアを更新する
`Client:` *clientid* `is trying to renew` *n.n.n.n*`, an IP address it has not leased.`	DHCP ネットワークテーブルに記録されているこのクライアントの IP アドレスが、クライアントが更新要求で指定した IP アドレスと一致しない。DHCP サーバーはこのリースを更新しない	この問題は、クライアントがまだ IP アドレスを使用しているのに、クライアントのレコードを削除した場合に発生する DHCP マネージャまたは `pntadm` を使用してネットワークテーブルを調べ、必要に応じて訂正する。クライアントの ID は、指定された IP アドレスと結合されていなければならない。 結合されていない場合は、アドレスプロパティを編集してこのクライアント ID を追加する

表 **11–3** IP アドレスの割り当てとリースに関する問題　*(続き)*

エラーメッセージ	説明	解決方法

`Client:` *clientid* `is trying to verify unrecorded address:` *n.n.n.n* `, ignored.`	指定されたクライアントがこのアドレスに対して DHCP ネットワークテーブルに登録されていない。そのため、要求は DHCP サーバーに無視される	このネットワークの別の DHCP サーバーで、このクライアントにアドレスを割り当てられる
		
		ただし、クライアントがこの IP アドレスをまだ使用しているのにそのクライアントのレコードが削除されていることに原因がある場合もある
		
		DHCP マネージャまたは `pntadm` を使用して、このサーバーやネットワークの他の DHCP サーバーにあるネットワークテーブルを調べ、必要に応じて訂正する
		
		何もせずにリースが期限切れになるのを待つこともできる。そうすれば、期限切れの後に自動的にクライアントが新しいアドレスリースを要求する
		
		クライアントに新しいリースをすぐに取得させたい場合は、次のコマンドを使って、このクライアント上で DHCP プロトコルを再起動する
		
		`ifconfig` *interface* `dhcp release`
		
		`ifconfig` *interface* `dhcp start`

DHCP クライアント設定の障害追跡

DHCP クライアントで発生する可能性がある問題は、一般的に次のカテゴリに分類されます。

- 257 ページの「DHCP サーバーとの通信の問題」
- 266 ページの「不正確な DHCP 設定情報に伴う問題」

DHCP サーバーとの通信の問題

この節では、ネットワークに DHCP クライアントを追加する際に発生する可能性がある問題について説明します。

クライアントソフトウェアを使用可能にし、システムをリブートすると、クライアントはそのネットワーク構成を DHCP サーバーから取得しようとします。クライアントがサーバーと通信できない場合は、次のようなエラーメッセージが表示されます。

```
DHCP or BOOTP server not responding
```

問題を特定するには、クライアントとサーバーの両方から診断情報を収集して、その結果を分析する必要があります。情報を収集するために、次のことができます。

1. クライアントをデバッグモードで実行する
2. サーバーをデバッグモードで実行する
3. snoop を起動してネットワークのトラフィックを監視する

これらの方法を個別に、または同時に実行できます。

収集した情報は、問題の原因がクライアントにあるのか、サーバーにあるのか、リレーエージェントにあるのかを特定して解決策を見つける上で役立ちます。

▼ DHCP クライアントをデバッグモードで実行する方法

Solaris DHCP クライアント以外のクライアントをデバッグモードで実行する方法については、それぞれのマニュアルを参照してください。

Solaris DHCP クライアントをデバッグモードで実行するには、次のようにします。

1. クライアントシステムでスーパーユーザーになります。

2. 次のコマンドで **DHCP** クライアントデーモンをいったん停止してからデバッグモードで起動します。

```
# pkill -x dhcpagent
# /sbin/dhcpagent -d1 -f &
# ifconfig interface dhcp start
```

デバッグモードで実行すると、クライアントデーモンは画面に DHCP の要求を実行中であるというメッセージを表示します。クライアントデバッグ出力については、259 ページの「DHCP クライアントデバッグ出力」を参照してください。

▼ DHCP サーバーをデバッグモードで実行する方法

1. サーバーシステム上でスーパーユーザーになります。

2. 次のコマンドで **DHCP** デーモンをいったん停止してからデバッグモードで起動します。

```
# pkill -x in.dhcpd
# /usr/lib/inet/in.dhcpd -d -v
```

さらに、デーモンの実行で一般に使用する in.dhcpd のコマンド行オプションを指定します。たとえば、デーモンを BOOTP リレーエージェントとして実行する場合は、in.dhcpd -d -v コマンドに -r オプションを付けます。

デバッグモードで実行すると、デーモンによって画面に DHCP や BOOTP の要求を処理しているというメッセージが表示されます。サーバーデバッグ出力については、260 ページの「DHCP サーバーデバッグ出力」を参照してください。

▼ snoop を使用して DHCP ネットワークトラフィックを監視する方法

1. **DHCP** サーバーシステムでスーパーユーザーになります。
2. **snoop** を起動して、サーバーのネットワークインタフェース間のネットワークトラフィックの追跡を開始します。

```
# /usr/sbin/snoop -d interface -o snoop-output-filename udp port 67 or udp port 68
```

次に例を示します。

```
#  /usr/sbin/snoop -d le0 -o /tmp/snoop.output udp port 67 or udp port 68
```

必要な情報を入手した後 Control-C を押してsnoop を明示的に停止するまで、snoop はインタフェースを監視し続けることに注意してください。

3. クライアントシステムを起動するか、クライアントシステムで **dhcpagent** を再起動します。

   dhcpagent の再起動については、258 ページの「DHCP クライアントをデバッグモードで実行する方法」 を参照してください。

4. サーバーシステムで **snoop** を使用して、ネットワークパケットの内容を含む出力ファイルを表示させます。

   ```
   # /usr/sbin/snoop -i snoop-output-filename -x0 -v
   ```

   例 :

   ```
   # /usr/sbin/snoop -i /tmp/snoop.output -x0 -v
   ```

dhcpagent コマンドの -d スイッチは、クライアントを冗長性 1 のデバッグモードにします。-f スイッチは、出力を syslog ではなくコンソールに送信します。ifconfig コマンド行の *interface* は、le0 など、 クライアントのネットワークインタフェースの名前で置き換えてください。

出力の解釈については、263 ページの「DHCP snoop 出力」を参照してください。

DHCP クライアントデバッグ出力

例 11–1 では、DHCP クライアントが DHCP 要求を送信し、DHCP サーバーから構成情報を受信した場合の通常のデバッグ出力を示しています。

例 **11–1** DHCP クライアントの通常のデバッグ出力例

```
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x27fc8 (DHCP filter)
/sbin/dhcpagent: debug: init_ifs: initted interface le0
/sbin/dhcpagent: debug: insert_ifs: le0: sdumax 1500, optmax 1260, hwtype 1, hwlen 6
/sbin/dhcpagent: debug: insert_ifs: inserted interface le0
/sbin/dhcpagent: debug: register_acknak: registered acknak id 5
/sbin/dhcpagent: debug: unregister_acknak: unregistered acknak id 5
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x26018 (ARP reply filter)
/sbin/dhcpagent: info: setting IP netmask on le0 to 255.255.192.0
```

例 **11–1** DHCP クライアントの通常のデバッグ出力例　*(続き)*

```
/sbin/dhcpagent: info: setting IP address on le0 to 10.23.3.233
/sbin/dhcpagent: info: setting broadcast address on le0 to 10.23.63.255
/sbin/dhcpagent: info: added default router 10.23.0.1 on le0
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x28054 (blackhole filter)
/sbin/dhcpagent: debug: configure_if: bound ifsp->if_sock_ip_fd
/sbin/dhcpagent: info: le0 acquired lease, expires Tue Aug 10 16:18:33 1999
/sbin/dhcpagent: info: le0 begins renewal at Tue Aug 10 15:49:44 1999
/sbin/dhcpagent: info: le0 begins rebinding at Tue Aug 10 16:11:03 1999
```

クライアントが DHCP サーバーと通信できない場合は、例 11–2 のようなデバッグ出力が表示されます。

例 **11–2** DHCP クライアントのデバッグ出力例

```
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x27fc8 (DHCP filter)
/sbin/dhcpagent: debug: init_ifs: initted interface le0
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
```

このメッセージは、要求がサーバーに届いていないか、サーバーが回答をクライアントに送信できないことを意味します。259 ページの「snoop を使用して DHCP ネットワークトラフィックを監視する方法」 の説明に従って snoop コマンドをサーバーで実行し、クライアントのパケットがサーバーに届いているかどうかを確認します。

DHCP サーバーデバッグ出力

通常のサーバーデバッグ出力は、デーモンが起動したときに、サーバーの構成情報とそれに続く各ネットワークインタフェースの情報を表示します。デーモンが起動されると、デバッグ出力には、デーモンが処理している要求の情報が表示されます。例 11–3 は、DHCP サーバーのデバッグ出力例です。この DHCP サーバーは、起動された後にクライアントのリースを延長しています。ただし、このクライアントは、応答していない別の DHCP サーバーが所有しているアドレスを使用しています。

例 **11–3** DHCP サーバーのデバッグ出力例

```
Daemon Version: 3.1
Maximum relay hops: 4
Transaction logging to console enabled.
Run mode is: DHCP Server Mode.
Datastore: nisplus
Path: org_dir.dhcp.test..:dhcp.test..:$
DHCP offer TTL: 10
Ethers compatibility enabled.
BOOTP compatibility enabled.
ICMP validation timeout: 1000 milliseconds, Attempts: 2.
Monitor (0005/hme0) started...
```

例 **11–3** DHCP サーバーのデバッグ出力例　*(続き)*

```
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.21.255.255
Netmask: 255.255.0.0
Address: 10.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
Broadcast: 10.22.255.255
Netmask: 255.255.0.0
Address: 10.22.0.1
Monitor (0007/qe0) started...
Thread Id: 0007 - Monitoring Interface: qe0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.23.63.255
Netmask: 255.255.192.0
Address: 10.23.0.1
Read 33 entries from DHCP macro database on Tue Aug 10 15:10:27 1999
Datagram received on network device: qe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qe0
Client: 0800201DBA3A maps to IP: 10.23.3.233
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
DHCP EXTEND 0934312543 0934316143 10.23.3.233 10.21.0.2
          0800201DBA3A SUNW.SPARCstation-10 0800201DBA3A
```

例 11–4 は DHCP デーモンのデバッグ出力です。この DHCP デーモンは、BOOTP リレーエージェントとして起動された後、クライアントの要求を DHCP サーバーに、サーバーの応答をクライアントにそれぞれリレーしています。

例 **11–4** BOOTP リレーに関するデバッグ出力の例

```
Relay destination: 10.21.0.4 (blue-servr2)     network: 10.21.0.0
Daemon Version: 3.1
Maximum relay hops: 4
Transaction logging to console enabled.
Run mode is: Relay Agent Mode.
Monitor (0005/hme0) started...
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.21.255.255
Netmask: 255.255.0.0
Address: 10.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
```

例 **11–4** BOOTP リレーに関するデバッグ出力の例　(続き)

```
Broadcast: 10.22.255.255
Netmask: 255.255.0.0
Address: 10.22.0.1
Monitor (0007/qe0) started...
Thread Id: 0007 - Monitoring Interface: qe0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.23.63.255
Netmask: 255.255.192.0
Address: 10.23.0.1
Relaying request 0800201DBA3A to 10.21.0.4, server port.
BOOTP RELAY-SRVR 0934297685 0000000000 0.0.0.0 10.21.0.4 0800201DBA3A
N/A 0800201DBA3A
Packet received from relay agent: 10.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
BOOTP RELAY-CLNT 0934297688 0000000000 10.23.0.1 10.23.3.233 0800201DBA3A
N/A 0800201DBA3A
Relaying request 0800201DBA3A to 10.21.0.4, server port.
BOOTP RELAY-SRVR 0934297689 0000000000 0.0.0.0 10.21.0.4 0800201DBA3A
N/A 0800201DBA3A
Packet received from relay agent: 10.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
```

問題がある場合は、このデバッグ出力が警告またはエラーメッセージを表示します。表 11–4 からエラーメッセージと解決策を見つけてください。

表 **11–4** DHCP サーバーのエラーメッセージ

メッセージ	説明	解決方法
`ICMP ECHO reply to OFFER candidate:` *ip_address* `disabling`	DHCP サーバーは、IP アドレスをクライアントに提供する前に、ping コマンドを実行してそのアドレスが使用されていないことを確認する。クライアントが回答する場合、そのアドレスは使用されている	構成したアドレスが使用されていないことを確認する

表 **11–4** DHCP サーバーのエラーメッセージ　(続き)

メッセージ	説明	解決方法
`No more IP addresses on` *network_address* `network.`	クライアントのネットワークに対応する DHCP ネットワークテーブル中に、使用可能な IP アドレスがない	DHCP マネージャまたは `pntadm` を使って IP アドレスを追加する。DHCP デーモンが複数のサブネットを監視している場合は、クライアントが属するサブネットに IP アドレスを追加する
`No more IP addresses for` *network_address* `network (BOOTP` 互換モードで DHCP デーモンを実行時 (`-b` オプション))	BOOTP はリース期間を使用しないので、DHCP サーバーは、BOOTP クライアントに割り当てるために設定された BOOTP フラグを持つ空きアドレスを検索する	DHCP マネージャを使用して、BOOTP アドレスを割り当てる
`Request to access nonexistent per network database:` *database_name* `in datastore:` *datastore.*	DHCP サーバーの構成の際に、サブネットの DHCP ネットワークテーブルが作成されていない	DHCP マネージャまたは `pntadm` を使用して、DHCP ネットワークテーブルと新しい IP アドレスを作成する
`There is no` *table_name* `dhcp-network table for DHCP client's network.`	DHCP サーバーの構成の際に、サブネットの DHCP ネットワークテーブルが作成されていない	DHCP マネージャまたは `pntadm` を使用して、DHCP ネットワークテーブルと新しい IP アドレスを作成する
`Client using non_RFC1048 BOOTP cookie.`	ネットワーク上のデバイスが、BOOTP のサポートされていない実装にアクセスしようとした	このデバイスを構成する必要がない場合は、このメッセージを無視する

DHCP snoop 出力

下記の snoop 出力を見ると、DHCP クライアントシステムと DHCP サーバーシステムの間でパケットが交換されていることがわかります。個々のパケットには、各システムの IP アドレス と、リレーエージェントやルーターが中間にある場合はそれらが示されます。システムの間でパケットが交換されていない場合は、クライアントシステムからサーバーシステムにアクセスできないのかもしれません。その場合、問題はより下位レベルにあります。

snoop の出力を評価するためには、要求が BOOTP リレーエージェントを介して行われるべきであるかどうかなど、本来の動作がどのようなものであるかを知っている必要があります。さらに、関係するシステムの (および、複数のネットワークインタフェースがある場合は、それらの) MAC アドレスや IP アドレスを知っていないと、それらの値が正しいかどうかを判断できません。次の例は、DHCP 肯定応答メッセー

ジの通常の snoop 出力を示しています。このメッセージは、blue-servr2 上の DHCP サーバーから MAC アドレスが 8:0:20:8e:f3:7e のクライアントに送信されたものです。このメッセージを見ると、サーバーがクライアントに IP アドレスとして 172.168.252.6 を、ホスト名として white-6 を割り当てていることがわかります。さらに、このメッセージには、クライアントに対するいくつもの標準的なネットワークオプションといくつかのベンダー固有のオプションが含まれています。

例 **11–5** 1 つのパケットに関する snoop 出力の例

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 26 arrived at 14:43:19.14
ETHER:  Packet size = 540 bytes
ETHER:  Destination = 8:0:20:8e:f3:7e, Sun
ETHER:  Source      = 8:0:20:1e:31:c1, Sun
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 526 bytes
IP:   Identification = 64667
IP:   Flags = 0x4 IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 254 seconds/hops
IP:   Protocol = 17 (UDP)
IP:   Header checksum = 157a
IP:   Source address = 10.21.0.4, blue-servr2
IP:   Destination address = 192.168.252.6, white-6
IP:   No options
IP:   UDP:  ----- UDP Header -----
UDP:
UDP:  Source port = 67
UDP:  Destination port = 68 (BOOTPC)
UDP:  Length = 506
UDP:  Checksum = 5D4C
UDP:
DHCP: ----- Dynamic Host Configuration Protocol -----
DHCP:
DHCP: Hardware address type (htype) =  1 (Ethernet (10Mb))
DHCP: Hardware address length (hlen) = 6 octets
DHCP: Relay agent hops = 0
DHCP: Transaction ID = 0x2e210f17
DHCP: Time since boot = 0 seconds
DHCP: Flags = 0x0000
DHCP: Client address (ciaddr) = 0.0.0.0
DHCP: Your client address (yiaddr) = 192.168.252.6
```

例 **11–5** 1 つのパケットに関する snoop 出力の例　(続き)

```
DHCP: Next server address (siaddr) = 10.21.0.2
DHCP: Relay agent address (giaddr) = 0.0.0.0
DHCP: Client hardware address (chaddr) = 08:00:20:11:E0:1B
DHCP:
DHCP: ----- (Options) field options -----
DHCP:
DHCP: Message type = DHCPACK
DHCP: DHCP Server Identifier = 10.21.0.4
DHCP: Subnet Mask = 255.255.255.0
DHCP: Router at = 192.168.252.1
DHCP: Broadcast Address = 192.168.252.255
DHCP: NISPLUS Domainname = dhcp.test
DHCP: IP Address Lease Time = 3600 seconds
DHCP: UTC Time Offset = -14400 seconds
DHCP: RFC868 Time Servers at = 10.21.0.4
DHCP: DNS Domain Name = sem.example.com
DHCP: DNS Servers at = 10.21.0.1
DHCP: Client Hostname = white-6
DHCP: Vendor-specific Options (166 total octets):
DHCP:   (02) 04 octets  0x8194AE1B (unprintable)
DHCP:   (03) 08 octets  "pacific"
DHCP:   (10) 04 octets  0x8194AE1B (unprintable)
DHCP:   (11) 08 octets  "pacific"
DHCP:   (15) 05 octets  "xterm"
DHCP:   (04) 53 octets  "/export/s2/base.s2s/latest/Solaris_8/Tools/Boot"
DHCP:   (12) 32 octets  "/export/s2/base.s2s/latest"
DHCP:   (07) 27 octets  "/platform/sun4m/kernel/unix"
DHCP:   (08) 07 octets  "EST5EDT"
  0: 0800 208e f37e 0800 201e 31c1 0800 4500    .. .ó~.. .1...E.
 16: 020e fc9b 4000 fe11 157a ac15 0004 c0a8    ....@....z......
 32: fc06 0043 0044 01fa 5d4c 0201 0600 2e21    ...C.D..]L.....!
 48: 0f17 0000 0000 0000 0000 c0a8 fc06 ac15    ................
 64: 0002 0000 0000 0800 2011 e01b 0000 0000    ........ .......
 80: 0000 0000 0000 0000 0000 0000 0000 0000    ................
 96: 0000 0000 0000 0000 0000 0000 0000 0000    ................
112: 0000 0000 0000 0000 0000 0000 0000 0000    ................
128: 0000 0000 0000 0000 0000 0000 0000 0000    ................
144: 0000 0000 0000 0000 0000 0000 0000 0000    ................
160: 0000 0000 0000 0000 0000 0000 0000 0000    ................
176: 0000 0000 0000 0000 0000 0000 0000 0000    ................
192: 0000 0000 0000 0000 0000 0000 0000 0000    ................
208: 0000 0000 0000 0000 0000 0000 0000 0000    ................
224: 0000 0000 0000 0000 0000 0000 0000 0000    ................
240: 0000 0000 0000 0000 0000 0000 0000 0000    ................
256: 0000 0000 0000 0000 0000 0000 0000 0000    ................
272: 0000 0000 0000 6382 5363 3501 0536 04ac    ......c.Sc5..6..
288: 1500 0401 04ff ffff 0003 04c0 a8fc 011c    ................
304: 04c0 a8fc ff40 0964 6863 702e 7465 7374    .....@.dhcp.test
320: 3304 0000 0e10 0204 ffff c7c0 0404 ac15    3...............
336: 0004 0f10 736e 742e 6561 7374 2e73 756e    ....sem.example.
352: 2e63 6f6d 0604 ac15 0001 0c07 7768 6974    com.........whit
368: 652d 362b a602 0481 94ae 1b03 0861 746c    e-6+.........pac
384: 616e 7469 630a 0481 94ae 1b0b 0861 746c    ific.........pac
```

例 **11–5** 1 つのパケットに関する snoop 出力の例　*(続き)*

```
400: 616e 7469 630f 0578 7465 726d 0435 2f65    ific...xterm.5/e
416: 7870 6f72 742f 7332 382f 6261 7365 2e73    xport/sx2/bcvf.s
432: 3238 735f 776f 732f 6c61 7465 7374 2f53    2xs_btf/latest/S
448: 6f6c 6172 6973 5f38 2f54 6f6f 6c73 2f42    olaris_x/Tools/B
464: 6f6f 740c 202f 6578 706f 7274 2f73 3238    oot. /export/s2x
480: 2f62 6173 652e 7332 3873 5f77 6f73 2f6c    /bcvf.s2xs_btf/l
496: 6174 6573 7407 1b2f 706c 6174 666f 726d    atest../platform
512: 2f73 756e 346d 2f6b 6572 6e65 6c2f 756e    /sun4m/kernel/un
528: 6978 0807 4553 5435 4544 54ff              ix..EST5EDT.
```

不正確な DHCP 設定情報に伴う問題

DHCP が受信したネットワーク構成情報の中に、誤った NIS ドメイン名や不正確なルーター IP アドレスといった不正確な情報がある場合は、このクライアントの DHCP サーバーが処理したマクロの中に正しくないオプションの値がないかを調べる必要があります。

正しくない情報の原因がどこにあるのかを特定する際には、次の一般的な指針に従ってください。

- 213 ページの「DHCP サーバー上で定義されたマクロを表示する方法 (DHCP マネージャ)」で説明されている、サーバーで定義されたマクロを調べてください。さらに、130 ページの「マクロ処理の順序」を参照して、このクライアントに対しどのマクロが自動的に処理されるのかを確認します。
- ネットワークテーブルを調べて、クライアントの IP アドレスに構成マクロとして割り当てられたマクロ (ある場合) を確認します。詳細については、195 ページの「DHCP サービスで IP アドレスを使用して作業する (作業マップ)」を参照してください。
- 複数のマクロで発生するオプションに注意して、最後に処理されるマクロでオプションに入力したい値を設定されることを確認します。
- 適切なマクロを編集して、正確な値がクライアントに確実に渡されるようにします。214 ページの「DHCP マクロの変更」を参照してください。

クライアント指定のホスト名に関連する問題

ここでは、独自のホスト名を指定し、それを DNS に登録する必要がある DHCP クライアントの問題について説明します。

クライアントがホスト名を要求しない

クライアントが Solaris DHCP クライアントでない場合は、そのクライアントのマニュアルを参照して、ホスト名を要求するために必要なクライアントの構成方法を調べてください。Solaris DHCP クライアントの場合は、181 ページの「特定のホスト名に応答するように Solaris クライアントを有効にする方法」 を参照してください。

要求されたホスト名を DHCP クライアントが受け取らない

表 **11–5** DHCP クライアントのホスト名要求に伴う問題と解決策

起こりうる問題	情報の収集	解決方法
クライアントは DHCP サーバーからオファーを受け取るが、サーバーが DNS 更新を行わない	1. snoop またはネットワークパケットを獲得するその他のアプリケーションをクライアントで実行する。DHCP サーバー識別子を探し、サーバーの IP アドレスを得る。 2. DHCP サーバーにログインし、動的更新を行うように構成されているかを確認する。`/etc/inet/dhcpsvc.conf` ファイルで `UPDATE_TIMEOUT` のエントリを探す。 3. DNS サーバーの `/etc/named.conf` ファイルで、適切なドメインの `zone` セクションの `allow-update` キーワードに DHCP サーバーの IP アドレスが指定されているかを確認する	DHCP サーバーと DNS サーバーの構成方法については、179 ページの「DHCP サーバーによる動的 DNS 更新の有効化」を参照する クライアントから 2 つの DHCP サーバーにアクセスできる場合は、両方のサーバーが DNS 更新を行うように構成されている必要がある
クライアントが FQDN オプション (オプションコード 89) を使ってホスト名を指定している。FQDN オプションは DHCP プロトコルに正式には含まれていないため、現在、DHCP ではサポートされていない	snoop またはネットワークパケットを獲得するその他のアプリケーションをサーバーで実行し、クライアントからのパケットに FQDN オプションがないかを確認する	`Hostname` オプション (オプションコード 12) を使ってホスト名を指定するようにクライアントの構成を変更する。 詳細は、クライアントのマニュアルを参照
クライアントにアドレスを提供する DHCP サーバーがクライアントの DNS 名を知らない	DHCP サーバーで、`DNSdmain` オプションに有効な値が設定されているかを確認する	このクライアントに対して処理されるマクロの `DNSdmain` オプションに正しい DNS ドメイン名を設定する。`DNSdmain` は通常、ネットワークマクロに含まれている

表 **11–5** DHCP クライアントのホスト名要求に伴う問題と解決策　(続き)

起こりうる問題	情報の収集	解決方法
クライアントが要求したホスト名が DHCP サーバーが管理していない IP アドレスに対応している。Solaris DHCP は、自らが管理していない IP アドレスに対し DNS 更新を行わない	`syslog` を調べて、`There is no` *n.n.n.n* `dhcp-network table for DHCP client's network.` や `DHCP network record for` *n.n.n.n* `is unavailable, ignoring request.` のような DHCP サーバーのメッセージを見つける。	対応する IP アドレスがない名前か、DHCP サーバーが管理するアドレスに対応する名前を選択するようにクライアントを構成する
クライアントが要求したホスト名に対応する IP アドレスは、現在使用中であるか、リースされているか、別のクライアントに提案中である	`syslog` を調べて、次のような DHCP サーバーのメッセージを探す。`ICMP ECHO reply to OFFER candidate:` *n.n.n.n.*	異なる IP アドレスに対応する名前を選択するようにクライアントを構成する。あるいは、そのアドレスを使用するクライアントからアドレスを取り返す
DHCP サーバーからの更新を受け付けるように DNS サーバーが構成されていない	DNS サーバーの `/etc/named.conf` ファイルを調べ、DHCP サーバーのドメインに対する適切な zone セクションで `allow-update` キーワードを持つ DHCP サーバーの IP アドレスを探す	DNS サーバーの構成方法については、180 ページの「DHCP クライアント用に動的 DNS 更新を有効にする方法」 を参照する DHCP サーバーに複数のインタフェースがある場合は、DHCP サーバーのすべてのアドレスからの更新を受け付けるように DNS サーバーを構成する必要がある場合がある。DNS サーバーのデバッグ機能を有効にして、更新が DNS サーバーに届いているか確認する。届いている場合は、更新が行われない原因をデバッグ出力で調べる

表 11–5 DHCP クライアントのホスト名要求に伴う問題と解決策　(続き)

起こりうる問題	情報の収集	解決方法
DNS 更新が、割り当てられた時間内に行われていない可能性がある。設定された時間内に DNS 更新が完了しないと、DHCP サーバーはホスト名をクライアントに返さない。ただし、DNS 更新を完了する試みは続けられる	nslookup コマンドを使って、更新が正常に終わっているかを確認する。nslookup(1M) のマニュアルページを参照する たとえば、DNS ドメインが hills.example.org で、DNS サーバーの IP アドレスが 10.76.178.11、クライアントが登録したいホスト名が cathedral であるとする。次のコマンドを使用すれば、cathedral が DNS に登録されたかどうかを知ることができる。 nslookup cathedral.hills.example.org 10.76.178.11	更新は正常に行われたが、割り当てられた時間を超えている場合は、タイムアウト値を増やす必要があるDNS 更新を有効にする方法の 手順 5 を参照する

第 12 章

DHCP のファイルおよびコマンド (リファレンス)

この章では、ファイルとそれらのファイルを使用するコマンドとの関係について説明します 。コマンドの使用方法については説明していません。

この章では、次の内容について説明します。

DHCP のコマンド

次の表に、ネットワーク上で DHCP を管理する際に役立つコマンドを示します。

表 **12–1** DHCP で使用されるコマンド

コマンド名	説明
`dhtadm`	`dhcptab` 内のオプションやマクロを変更するときに使用する。 このコマンドは、DHCP 情報を自動的に変更するために作成するスクリプトで最も役立つ。`dhtadm` に `-P` オプションを指定し、その結果を `grep` コマンドに渡すと、`dhcptab` 内の特定のオプション値を素早く検索できる
`pntadm`	DHCP ネットワークテーブルを変更するときに使用する。このテーブルでは、クライアント ID と IP アドレスが対応付けられ、オプションとして構成情報と IP アドレスが関連付けられている
`dhcpconfig`	DHCP サーバーや BOOTP リレーエージェントの構成や構成解除を行なったり、データストアを別のデータストアへ変換したり、DHCP 構成データのインポートやエクスポートを行うときに使用する

表 **12–1** DHCP で使用されるコマンド　(続き)

コマンド名	説明
in.dhcpd	DHCP サーバーデーモン。システムスクリプトでは、このコマンドを使って DHCP サービスの起動や停止を行う。 デバッグ用の -d のようなデフォルトでないオプションを使用して in.dhcpd を起動できる
dhcpmgr	DHCP マネージャは、DHCP サービスの構成や管理を行うグラフィカルツールです。DHCP マネージャは、推奨される Solaris DHCP 管理ツールです。
ifconfig	システムの起動時に使用され、IP アドレスをネットワークインタフェースに割り当てたり、ネットワークインタフェースのパラメータを構成したりする (または、その両方)。Solaris DHCP クライアントでは、ifconfig は DHCP を起動し、IP アドレスなどの、ネットワークインタフェースの設定に必要なパラメータを取得する
dhcpinfo	クライアントシステムのシステム起動スクリプトの中で、DHCP クライアントデーモン (dhcpagent) からホスト名などの情報を取得するときに使用する。また、スクリプトやコマンド行で dhcpinfo を使用して、特定のパラメータ値を取得することもできる
snoop	ネットワークを介して送信されているパケットの内容を獲得および表示するときに使用する。snoop は、DHCP サービスに伴う問題を障害追跡する際に役立つ
dhcpagent	DHCP クライアントデーモン。DHCP プロトコルのクライアント側を実装している

スクリプトにおける DHCP コマンドの実行

dhcpconfig、dhtadm、pntadm コマンドは、スクリプト中での使用に適しています。 特に、pntadm コマンドは大量の IP アドレスエントリを DHCP ネットワークテーブルに作成するときに便利です。次のサンプルスクリプトでは、バッチモードで pntadm を使って、IP アドレスを作成しています。

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する

```
#! /usr/bin/ksh
#
# This script utilizes the pntadm batch facility to add client entries
# to a DHCP network table. It assumes that the user has the rights to
# run pntadm to add entries to DHCP network tables.

#
# Based on the nsswitch setting, query the netmasks table for a netmask.
# Accepts one argument, a dotted IP address.
#
get_netmask()
{
    MTMP=`getent netmasks ${1} | awk '{ print $2 }'`
    if [ ! -z "${MTMP}" ]
    then
        print - ${MTMP}
```

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する　*(続き)*

```
    fi
}

#
# Based on the network specification, determine whether or not network is
# subnetted or supernetted.
# Given a dotted IP network number, convert it to the default class
# network.(used to detect subnetting). Requires one argument, the
# network number. (e.g. 10.0.0.0) Echos the default network and default
# mask for success, null if error.
#
get_default_class()
{
    NN01=${1%%.*}
    tmp=${1#*.}
    NN02=${tmp%%.*}
    tmp=${tmp#*.}
    NN03=${tmp%%.*}
    tmp=${tmp#*.}
    NN04=${tmp%%.*}
    RETNET=""
    RETMASK=""

    typeset -i16 ONE=10#${1%%.*}
    typeset -i10 X=$((${ONE}&16#f0))
    if [ ${X} -eq 224 ]
    then
        # Multicast
        typeset -i10 TMP=$((${ONE}&16#f0))
        RETNET="${TMP}.0.0.0"
        RETMASK="240.0.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#80))
    if [ -z "${RETNET}" -a ${X} -eq 0 ]
    then
        # Class A
        RETNET="${NN01}.0.0.0"
        RETMASK="255.0.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#c0))
    if [ -z "${RETNET}" -a ${X} -eq 128 ]
    then
        # Class B
        RETNET="${NN01}.${NN02}.0.0"
        RETMASK="255.255.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#e0))
    if [ -z "${RETNET}" -a ${X} -eq 192 ]
    then
        # Class C
        RETNET="${NN01}.${NN02}.${NN03}.0"
        RETMASK="255.255.255.0"
    fi
```

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する　(続き)

```
    print - ${RETNET} ${RETMASK}
    unset NNO1 NNO2 NNO3 NNO4 RETNET RETMASK X ONE
}

#
# Given a dotted form of an IP address, convert it to its hex equivalent.
#
convert_dotted_to_hex()
{
    typeset -i10 one=${1%%.*}
    typeset -i16 one=${one}
    typeset -Z2 one=${one}
    tmp=${1#*.}

    typeset -i10 two=${tmp%%.*}
    typeset -i16 two=${two}
    typeset -Z2 two=${two}
    tmp=${tmp#*.}

    typeset -i10 three=${tmp%%.*}
    typeset -i16 three=${three}
    typeset -Z2 three=${three}
    tmp=${tmp#*.}

    typeset -i10 four=${tmp%%.*}
    typeset -i16 four=${four}
    typeset -Z2 four=${four}

     hex=`print - ${one}${two}${three}${four} | sed -e 's/#/0/g'`
     print - 16#${hex}
     unset one two three four tmp
}

#
# Generate an IP address given the network address, mask, increment.
#
get_addr()
{
    typeset -i16 net=`convert_dotted_to_hex ${1}`
    typeset -i16 mask=`convert_dotted_to_hex ${2}`
    typeset -i16 incr=10#${3}

    # Maximum legal value - invert the mask, add to net.
    typeset -i16 mhosts=~${mask}
    typeset -i16 maxnet=${net}+${mhosts}

    # Add the incr value.
    let net=${net}+${incr}

    if [ $((${net} < ${maxnet})) -eq 1 ]
    then
        typeset -i16 a=${net}\&16#ff000000
        typeset -i10 a="${a}>>24"
```

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する　*(続き)*

```
        typeset -i16 b=${net}\&16#ff0000
        typeset -i10 b="${b}>>16"

        typeset -i16 c=${net}\&16#ff00
        typeset -i10 c="${c}>>8"

        typeset -i10 d=${net}\&16#ff
        print - "${a}.${b}.${c}.${d}"
    fi
    unset net mask incr mhosts maxnet a b c d
}

# Given a network address and client address, return the index.
client_index()
{
    typeset -i NNO1=${1%%.*}
    tmp=${1#*.}
    typeset -i NNO2=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO3=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO4=${tmp%%.*}

    typeset -i16 NNF1
    let NNF1=${NNO1}
    typeset -i16 NNF2
    let NNF2=${NNO2}
    typeset -i16 NNF3
    let NNF3=${NNO3}
    typeset -i16 NNF4
    let NNF4=${NNO4}
    typeset +i16 NNF1
    typeset +i16 NNF2
    typeset +i16 NNF3
    typeset +i16 NNF4
    NNF1=${NNF1#16\#}
    NNF2=${NNF2#16\#}
    NNF3=${NNF3#16\#}
    NNF4=${NNF4#16\#}
    if [ ${#NNF1} -eq 1 ]
    then
        NNF1="0${NNF1}"
    fi
    if [ ${#NNF2} -eq 1 ]
    then
        NNF2="0${NNF2}"
    fi
    if [ ${#NNF3} -eq 1 ]
    then
        NNF3="0${NNF3}"
    fi
    if [ ${#NNF4} -eq 1 ]
```

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する　(続き)

```
    then
        NNF4="0${NNF4}"
    fi
    typeset -i16 NN
    let NN=16#${NNF1}${NNF2}${NNF3}${NNF4}
    unset NNF1 NNF2 NNF3 NNF4

    typeset -i NNO1=${2%%.*}
    tmp=${2#*.}
    typeset -i NNO2=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO3=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO4=${tmp%%.*}
    typeset -i16 NNF1
    let NNF1=${NNO1}
    typeset -i16 NNF2
    let NNF2=${NNO2}
    typeset -i16 NNF3
    let NNF3=${NNO3}
    typeset -i16 NNF4
    let NNF4=${NNO4}
    typeset +i16 NNF1
    typeset +i16 NNF2
    typeset +i16 NNF3
    typeset +i16 NNF4
    NNF1=${NNF1#16\#}
    NNF2=${NNF2#16\#}
    NNF3=${NNF3#16\#}
    NNF4=${NNF4#16\#}
    if [ ${#NNF1} -eq 1 ]
    then
        NNF1="0${NNF1}"
    fi
    if [ ${#NNF2} -eq 1 ]
    then
        NNF2="0${NNF2}"
    fi
    if [ ${#NNF3} -eq 1 ]
    then
        NNF3="0${NNF3}"
    fi
    if [ ${#NNF4} -eq 1 ]
    then
        NNF4="0${NNF4}"
    fi
    typeset -i16 NC
    let NC=16#${NNF1}${NNF2}${NNF3}${NNF4}
    typeset -i10 ANS
    let ANS=${NC}-${NN}
    print - $ANS
}
```

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する　*(続き)*

```
#
# Check usage.
#
if [ "$#" != 3 ]
then
    print "This script is used to add client entries to a DHCP network"
    print "table by utilizing the pntadm batch facilty.\n"
    print "usage: $0 network start_ip entries\n"
    print "where: network is the IP address of the network"
        print "       start_ip is the starting IP address \n"
        print "       entries is the number of the entries to add\n"
    print "example: $0 10.148.174.0 10.148.174.1 254\n"
    return
fi

#
# Use input arguments to set script variables.
#
NETWORK=$1
START_IP=$2
typeset -i STRTNUM=`client_index ${NETWORK} ${START_IP}`
let ENDNUM=${STRTNUM}+$3
let ENTRYNUM=${STRTNUM}
BATCHFILE=/tmp/batchfile.$$
MACRO=`uname -n`

#
# Check if mask in netmasks table. First try
# for network address as given, in case VLSM
# is in use.
#
NETMASK=`get_netmask ${NETWORK}`
if [ -z "${NETMASK}" ]
then
    get_default_class ${NETWORK} | read DEFNET DEFMASK
    # use the default.
    if [ "${DEFNET}" != "${NETWORK}" ]
    then
        # likely subnetted/supernetted.
        print - "\n\n###\tWarning\t###\n"
        print - "Network ${NETWORK} is netmasked, but no entry was found  \n
              in the 'netmasks' table; please update the 'netmasks'  \n
              table in the appropriate nameservice before continuing. \n
              (See /etc/nsswitch.conf.) \n">&2
        return 1
    else
        # use the default.
        NETMASK="${DEFMASK}"
    fi
fi

#
# Create a batch file.
```

例 **12–1** addclient.ksh スクリプトで pntadm コマンドを使用する　(続き)

```
#
print -n "Creating batch file "
while [ ${ENTRYNUM} -lt ${ENDNUM} ]
do
    if [ $((${ENTRYNUM}-${STRTNUM}))%50 -eq 0 ]
    then
        print -n "."
    fi

    CLIENTIP=`get_addr ${NETWORK} ${NETMASK} ${ENTRYNUM}`
    print "pntadm -A ${CLIENTIP} -m ${MACRO} ${NETWORK}">> ${BATCHFILE}
    let ENTRYNUM=${ENTRYNUM}+1
done
print " done.\n"

#
# Run pntadm in batch mode and redirect output to a temporary file.
# Progress can be monitored by using the output file.
#
print "Batch processing output redirected to ${BATCHFILE}"
print "Batch processing started."

pntadm -B ${BATCHFILE} -v> /tmp/batch.out 2>&1

print "Batch processing completed."
```

DHCP のファイル

表 12–2 に、Solaris DHCP に関連するファイルを示します。

表 **12–2** DHCP デーモンや DHCP コマンドで使用されるファイル

ファイル/テーブル	説明
dhcptab	オプションとその値の組み合わせからなる DHCP 構成情報のテーブルを表す総称的な用語。構成情報はさらにマクロとしてグループ化される。 dhcptab テーブルの名前と位置は、DHCP 情報の格納にどのデータストアを使用するかによって異なる
DHCP ネットワークテーブル	IP アドレスをクライアント ID と構成オプションに割り当てる。DHCP ネットワークテーブルの名前は、10.21.32.0 など、ネットワークの IP アドレスに基づいて付けられる。dhcp_network というファイルはありません。DHCP ネットワークテーブルの名前と位置は、DHCP 情報の格納にどのデータストアを使用するかによって異なる

表 **12–2** DHCP デーモンや DHCP コマンドで使用されるファイル　(続き)

ファイル/テーブル	説明
`dhcpsvc.conf`	DHCP デーモンの起動オプションと、`dhcptab` テーブルおよびネットワークテーブルのデータストアリソースおよび場所を格納している。 このファイルは `/etc/inet` ディレクトリにある
`nsswitch.conf`	ネームサービスデータベースの場所と、それらのデータベースをどのような順序で検索してさまざまな情報を入手するかを指定する。`nsswitch.conf` ファイルは、DHCP サーバーを構成する際に正確な構成情報を入手するために使用される。このファイルは、`/etc` ディレクトリに存在する
`resolv.conf`	DNS リゾルバによって使用される情報が入っている。DHCP サーバーの構成中に、このファイルで、DNS ドメインと DNS サーバーに関する情報が調べられる。このファイルは、`/etc` ディレクトリに存在する
`dhcp.`*interface*	ファイル名 (`dhcp.qe0` など) で指定されたクライアントのネットワークインタフェースで DHCP が使用されることを示す。`dhcp.`*interface* ファイルには、そのクライアント上で DHCP を起動するための `ifconfig` *interface* `dhcp start` *option* コマンドにオプションとして渡されるコマンドが含まれていることがある。このファイルは、Solaris DHCP クライアントシステムの `/etc` ディレクトリにある
interface`.dhc`	DHCP から得られた特定のネットワークインタフェースの構成パラメータが入っている。インタフェースの IP アドレスのリースがドロップされると、このクライアントは、`/etc/dhcp/`*interface*`.dhc` にある現在の設定情報をキャッシュする。DHCP が次にこのインタフェースで起動するときに、リースの有効期限内であれば、このクライアントはキャッシュされた情報を使用するように要求する。DHCP サーバーがこの要求を拒否すると、クライアントは標準 DHCP リースネゴシエーション手順を開始する
`dhcpagent`	`dhcpagent` クライアントデーモンのパラメータ値を設定する。このファイルへのパスは `/etc/default/dhcpagent`。 パラメータの詳細については、このファイル自体、または `dhcpagent(1M)` のマニュアルページを参照
DHCP `inittab`	データタイプなど、DHCP オプションコードのさまざまな要素を定義するとともに、ニーモニックラベルを割り当てる。ファイル構文については、`dhcp_inittab` のマニュアルページを参照する クライアント側では、`dhcpinfo` が `/etc/dhcp/inittab` ファイル中の情報を人が判読可能な情報として提供する。このファイルは、`/etc/dhcp/dhcptags` ファイルに代わる。この変更については、280 ページの「DHCP のオプション」を参照する。DHCP サーバーシステムでは、DHCP デーモンと管理ツールがこのファイルから DHCP オプション情報を入手する

DHCP のオプション

従来、DHCP のオプション情報は、サーバーの dhcptab テーブルやクライアントの dhcptags ファイル、それに in.dhcpd や snoop、dhcpinfo、dhcpmgr の内部テーブルなど、Solaris DHCP の複数の場所に格納されてきました。 そのため、Solaris 8 から DHCP 製品には、オプション情報を統合するために /etc/dhcp/inittab ファイルが導入されています。このファイルについての詳細は、dhcp_inittab のマニュアルページを参照してください。

Solaris DHCP クライアントでは、dhcptags ファイルの代わりに DHCP inittab ファイルを使って、DHCP パケットで受信するオプションコードの情報を入手します。DHCP サーバーの in.dhcpd、snoop、dhcpmgr プログラムでもこの inittab ファイルを使用します。

注 – Solaris DHCP を使用するほとんどのサイトでは、この変更による影響はありません。この影響があるのは、Solaris 8 以降へのアップグレードを計画している場合で、かつ新しい DHCP オプションを以前に作成し、/etc/dhcp/dhcptags ファイルを変更済みで、この変更を保持したい場合だけです。アップグレードを行うと、dhcptags ファイルが変更されたために DHCP inittab ファイルを変更する必要があることを示すメッセージがアップグレードログに書き込まれます。

---

dhcptags と inittab の違い

inittab ファイルには dhcptags ファイルよりも多くの情報が含まれ、その構文も異なります。

dhcptags のエントリの例は次の通りです。

```
33 StaticRt - IPList Static_Routes
```

33 は DHCP パケットで渡される数値コードです。StaticRt はオプション名であり、IPList は期待されるデータが IP アドレスのリストであることを示しています。Static_Routes はこのオプションを説明する名前です。

inittab ファイルは、これらのオプションを 1 行で表した複数のレコードから構成されています。形式は、dhcptab のシンボルを定義する形式と似ています。 次の表に、inittab の構文について説明します。

表 **12–3** DHCP `inittab` ファイルの構文

オプション	説明
option-name	オプションの名前。オプション名は、そのオプションのカテゴリ内部で一意である必要がある。また、Standard、Site、Vendor のカテゴリにある、他のオプション名と重複できない。たとえば、同じ名前を持つ Site オプションを 2 つ持つことはできず、Standard のオプションと同じ名前の Site のオプションは作成できない
category	オプションが所属する名前空間を特定する。Standard、Site、Vendor、Field、または Internal の 1 つにする必要がある
code	オプションがネットワーク経由で送信されたときにそのオプションを特定する。多くの場合、カテゴリがなくてもコードはオプションを一意に特定する。ただし、Field や Internal のような内部カテゴリの場合は、コードが他の目的のために使用されていることがあるため、広域的に一意ではないことがある。コードは、オプションのカテゴリ内部では一意であることが必要で、Standard と Site のフィールドにあるコードと重複することはできない
type	このオプションと関連するデータを記述する。有効なタイプには、IP, Ascii, Octet, Boolean, Unumber8, Unumber16, Unumber32, Unumber64, Snumber8, Snumber16, Snumber32, Snumber64 がある。 数値の場合、最初の U または S はその数値が符号付きか符合なしかを表し、終わりの数字はその数値のビット数を表す。タイプには、大文字小文字の区別はない
granularity	このオプションの値を構成するデータ単位数を記述する
maximum	このオプションに指定可能な値の個数を記述する。0 は、無限大の数を表す

consumers	この情報を使用できるプログラムを記述する。これには次の `sdmi` を指定する。
	`s – snoop`
	`d – in.dhcpd`
	`m – dhcpmgr`
	`i – dhcpinfo`

`inittab` のエントリの例は、次の通りです。

```
StaticRt Standard, 33, IP, 2, 0, sdmi
```

このエントリは、`StaticRt` という名前のオプションを記述しています。このオプションは、Standard カテゴリにあり、オプションコード 33 です。データ型が `IP`、データ単位数が 2 個、指定可能な値の数が無限大 (0) であるため、無限個の IP アドレスの組を指定できることになります。このオプションを利用するのは `sdmi`: `snoop`、`in.dhcpd`、 `dhcpmgr`、`dhcpinfo` です。

dhcptags エントリの inittab エントリへの変換

以前にエントリを `dhcptags` ファイルに追加している場合は、新しい `inittab` ファイルに対応するエントリを追加する必要があります。次の例では、`dhcptags` エントリの例を `inittab` フォーマットで表す方法を示しています。

ネットワークに接続されたファックスについて、次の `dhcptags` エントリを追加したと想定してください。

```
128 FaxMchn - IP Fax_Machine
```

コード 128 は、サイトカテゴリになければならないことを意味しており、オプション名は FaxMchn、データタイプは IP です。

対応する inittab エントリは次の通りです。

```
FaxMchn SITE, 128, IP, 1, 1, sdmi
```

データ単位数が 1、指定可能な値の数が 1 なので、このオプションには 1 つの IP アドレスを指定することを表しています。

第 13 章

IPv6 (トピック)

第 14 章	IPv6 の概要
第 15 章	IPv6 関連作業の手順
第 16 章	Solaris での IPv6 の実装
第 17 章	IPv6 への移行計画とそのメカニズム

第 14 章

IPv6 (概要)

Internet Protocol、バージョン 6 (IPv6) は、現在の IPv4 から飛躍的な進歩を図った Internet Protocol (IP) の新バージョンです。IPv4 から IPv6 には無理なく移行することができます。規定されている移行メカニズムを使用することにより、現在の運用に混乱を生じることなく IPv6 ネットワークを展開できます。IPv6 ではアドレス空間が増加しています。また、シンプルになったヘッダーフォーマット、認証とプライバシのサポート、アドレス割り当ての自動設定を採用し、サービス品質を一新してインターネット機能を強化しました。

この章では、以下の内容について説明します。

IPv6 の機能

IPv4 からIPv6 への変更内容は、次のように大きく分類できます。

- 拡張されたルーティングとアドレス指定機能 – IPv6 では IP アドレスサイズを 32 ビットから 128 ビットに拡大して、サポートするアドレス指定階層を広げています。また、アドレス可能なノード数を増やし、アドレスの自動設定を容易にしています。

  スコープフィールドの追加により、マルチキャストアドレスに対するマルチキャストルーティングのスケーラビリティを強化しました。

任意キャストアドレスという新しいタイプのアドレスを定義しました。任意キャストアドレスは、ノードセットを識別します。任意キャストアドレスに送信されたパケットは ノードの 1 つに配信されます。IPv6 ソースルートでは任意キャストアドレスを使用して、ノードでトラフィックフローのパスを制御できます。

- ヘッダーフォーマットの簡略化 – IPv4 ヘッダーフィールドが一部削除されたり、オプションになったりしました。この変更によってパケット処理の共通部分の処理コストが削減されます。また、アドレスのサイズは増えましたが、IPv6 ヘッダーの帯域幅コストは可能な限り少なくなりました。IPv6 アドレスの長さは、IPv4 アドレスの 4 倍ですが、IPv6 ヘッダーのサイズは IPv4 の 2 倍に抑えられています。
- オプションサポートの強化 – IP ヘッダーオプションのコード化の方法を変更したため、転送効率が改善されました。また、オプションの長さに関する制限が緩和されています。さらに、将来新しいオプションを導入する際の柔軟性が高くなりました。
- サービス品質の機能 – 新しい機能が追加されて、送信側が特別な処理を必要とする特定のトラフィックフローに属するパケットのラベル指定が可能になりました。たとえば、デフォルト以外の品質サービスやリアルタイムサービスなどです。
- 認証機能と機密機能 – IPv6 には認証、データの完全性、機密性をサポートする拡張機能の定義が組み込まれています。

IPv6 のヘッダーと拡張機能

IPv6 プロトコルは、基本 IPv6 ヘッダー、IPv6 拡張ヘッダーを含むヘッダーセットを定義します。

ヘッダーフォーマット

図 14–1 は、IPv6 ヘッダーに使用される要素とその順序を示します。

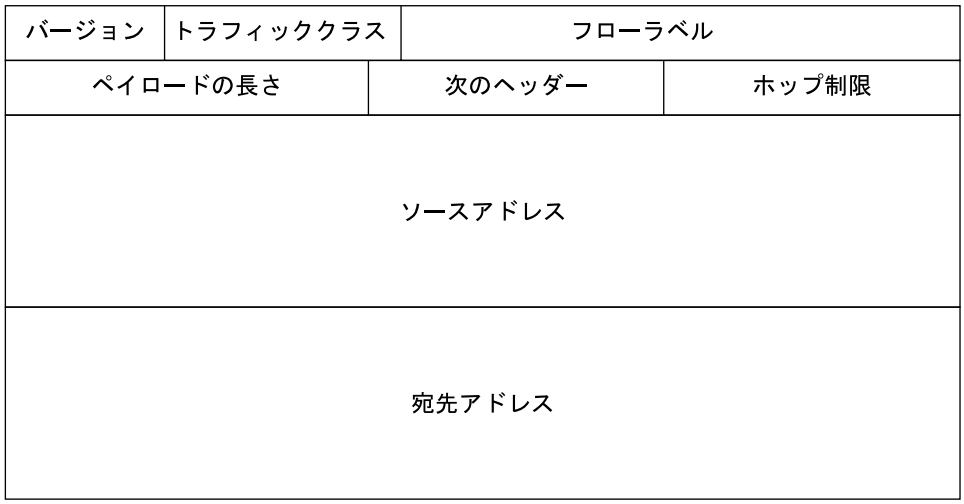


図 **14–1** IPv6 ヘッダーフォーマット

次に各ヘッダーフィールドの機能について説明します。

- バージョン – 4 ビットインターネットプロトコルバージョン番号。IPv6 では 6
- トラフィッククラス – 8 ビットトラフィッククラスフィールドの値 (305 ページの「トラフィッククラス」を参照)
- フローラベル – 20 ビットフィールド (304 ページの「IPv6 サービス品質 (QoS) 機能」を参照)
- ペイロードの長さ – オクテット単位で表す 16 ビット符号なし整数。IPv6 ヘッダーに続くパケットの残り
- 次のヘッダー – 8 ビットセレクタ。IPv6 ヘッダーのすぐ後ろに続くヘッダーのタイプを識別する。IPv4 プロトコルフィールドと同じ値を使用する (287 ページの「拡張ヘッダー」を参照)
- ホップ制限 – 8 ビット符号なし整数。パケットを送信するノードごとに値が 1 ずつ減る。ホップ制限がゼロになるとパケットが廃棄される
- ソースアドレス – 128 ビット。パケットの初期送信側のアドレス (288 ページの「IPv6 アドレス指定」を参照)
- 宛先アドレス – 128 ビット。パケットの予定受信側のアドレス。オプションのルーティングヘッダーがある場合、必ずしも受信側とは限らない

拡張ヘッダー

IPv6 には、IPv4 から強化されたオプション機能があります。IPv6 オプションは、IPv6 ヘッダーとトランスポート層の間の独立した拡張ヘッダーにあります。パケットが最終的な宛先に到着するまで、その配送パスに存在するルーターは、ほとんどの場

合 IPv6 拡張ヘッダを確認または処理しません。そのため、オプションがあるパケットを処理するルーターの性能が大幅に改善されました。IPv4 では、オプションがある場合、ルーターですべてのオプションを調べる必要がありました。

その他の改良点としては、IPv4 オプションと異なり、IPv6 拡張ヘッダーは長さを任意に設定できます。またパケットに組み込むことのできるオプションの合計数が 40 バイト以内に限定されない点があります。この機能とその処理方法によって、IPv4 では非現実的であった機能を IPv6 オプションが使用できるようになりました。その良い例が IPv6 認証オプションとセキュリティカプセル化オプションです。

後続のオプションヘッダー (およびそのあとのトランスポートプロトコル) を処理する際の性能を強化するため、IPv6 オプションは常に 8 オクテットの整数倍の長さです。これにより、後続ヘッダーのバイト境界が維持されています。

次の IPv6 拡張ヘッダーが現在、定義されています。

- ルーティング – 拡張ルーティング (IPv4 ルーズソースルートにあたる)
- 断片化 – 断片化および再結合
- 認証 – 整合性および認証、セキュリティ
- カプセル化 – 機密性
- ホップバイホップオプション – ホップごとの処理が必要な特別なオプション
- 宛先オプション – 宛先ノードが判断するオプション情報

IPv6 アドレス指定

IPv6 アドレスは 128 ビット長の識別子であり、個々のインタフェースや、インタフェースセットを識別します。すべてのタイプの IPv6 アドレスは、インタフェースに割り当てられ、ノード (ホストやルーター) には割り当てられません。各インタフェースの所属先は 1 つのノードだけなので、ノードのインタフェースのユニキャストアドレスは、そのノードの識別子として使用できます。1 つのインタフェースには、任意のタイプの複数の IPv6 アドレスを割り当てることができます。

IPv6 アドレスには、次の 3 種類のタイプがあります。ユニキャスト、任意キャスト、およびマルチキャスト。

- ユニキャストアドレスは、1 つのインタフェースを識別する
- 任意キャストアドレスは、インタフェースのセットを識別する。任意キャストアドレスに送信されたパケットは そのセットのメンバーの 1 つに配信される
- マルチキャストアドレスは、インタフェースのグループを識別する。マルチキャストアドレスに送信されるパケットは、グループにあるすべてのインタフェースに配信される。

IPv6 では、ブロードキャストアドレスの代わりにマルチキャストアドレスが使われます。

IPv6 は IPv4 アドレスの 4 倍のビット数のアドレス (128 対 32) をサポートします。したがって、計算上はそのアドレス空間は IPv4 のアドレス空間の 40 億 x 40 億倍の大きさになります。実際にはアドレスの割り当てとルーティングでは階層を作成する必要があり、アドレス領域の利用効率が減少するため、結果として、利用できるアドレス数は減少します。ただし当面は、IPv6 で提供するアドレス空間で十分です。

アドレスの先頭ビットでは IPv6 アドレスのタイプを指定します。この先頭ビットがある可変長フィールドをフォーマットプレフィックス (FP) といいます。 次の表は、これらのプレフィックス (接頭辞) の初期割り当てです。

表 **14–1** フォーマットプレフィックスの割り当て

割り当て	プレフィックス (バイナリ)	アドレス領域の端数
予約済み	0000 0000	1/256
割り当てなし	0000 0001	1/256
NSAP 割り当てに予約	0000 001	1/128
IPX 割り当てに予約	0000 010	1/128
割り当てなし	0000 011	1/128
割り当てなし	0000 1	1/32
割り当てなし	0001	1/16
集約グローバルユニキャストアドレス	001	1/8
割り当てなし	010	1/8
割り当てなし	011	1/8
ニュートラル相互接続ベースユニキャストアドレスに予約	100	1/8
割り当てなし	101	1/8
割り当てなし	110	1/8
割り当てなし	1110	1/16
割り当てなし	1111 0	1/32
割り当てなし	1111 10	1/64
割り当てなし	1111 110	1/128
割り当てなし	1111 1110 0	1/512
リンクローカル用アドレス	1111 1110 10	1/1024
サイトローカル用アドレス	1111 1110 11	1/1024
マルチキャストアドレス	1111 1111	1/256

割り当てには、集約グローバルユニキャストアドレス、ローカル用アドレス、マルチキャストアドレスの直接割り当てがサポートされています。NSAP (ネットワークサービスアクセスポイント) アドレス、IPX (相互ネットワークパケット交換プロトコル) アドレス、ニュートラル相互接続アドレスには空間が予約されています。残りのアドレス空間は将来用に割り当てなしになっています。この残ったアドレス領域は、既存の領域の拡張部分 (集約グローバルユニキャストアドレスへの追加など) または、新しい用途 (独立したロケータや識別子) に利用できます。なお、任意キャストアドレスはユニキャストアドレス空間の範囲外に割り当てられるため、ここには示していません。

初期設定で、アドレス空間の約 15 パーセントが割り当てられます。残りの 85 パーセントは将来用に予約されています。

ユニキャストアドレス

IPv6 ユニキャストアドレスの割り当て形式は、次のとおりです。

- 集約グローバルユニキャストアドレス
- ニュートラル相互接続ユニキャストアドレス
- NSAP アドレス
- IPX 階層アドレス
- サイトローカル用アドレス
- リンクローカル用アドレス
- IPv4 対応ホストアドレス

その他のアドレスタイプは、あとから定義できます。

集約グローバルユニキャストアドレス

集約グローバルユニキャストアドレスは、グローバル通信に使用するアドレスです。CIDR (クラスレス相互ドメインルーティング) における IPv4 アドレスに機能的に似ています。表 14–2 に、そのフォーマットをまとめます。

表 14–2 集約グローバルユニキャストアドレスのフォーマット

3 ビット	13 ビット	8 ビット	24 ビット	16 ビット	64 ビット
FP	TLA ID	RES	NLA ID	SLA ID	Interface ID

FP	フォーマットプレフィックス (001)
TLA ID	最上位集約識別子
RES	将来用に予約
NLA ID	次レベル集約識別子

SLA ID	サイトレベル集約識別子
INTERFACE ID	インタフェース識別子

最初の 48 ビットはパブリックトポロジを表します。次の 16 ビットは各サイトのトポロジを表します。

最初の 3 ビットは集約グローバルユニキャストアドレスとしてアドレスを識別します。次のフィールドである TLA ID はルーティング階層の最上位レベルです。その次の 8 ビットは将来用に予約されています。NLA ID フィールドは TLA ID を割り当てられた組織が、アドレス指定階層の作成と、サイトの識別に使用します。

SLA ID フィールドは、組織で各ローカルアドレス指定階層の作成とサブネットを識別するときに使用します。SLA ID フィールドの使い方は IPv4 のサブネットと似ていますが、組織別に割り当てることのできるサブネット数がはるかに多いところが異なります。16 ビット SLA ID フィールドがサポートするサブネットの数は 65,535 です。Interface ID は、リンク上のインタフェースを識別するために使用します。Interface ID はそのリンク上で一意である必要があります。また、より広い範囲で一意とすることができます。通常、インタフェース識別子はインタフェースのリンク層のアドレスと同じか、そこから派生した値です。

ローカルアドレス

ローカル用アドレスは、ローカルにルーティング可能な範囲のみを対象とするユニキャストアドレスです。使用できるのは、サブネット内または加入者ネットワーク内に限定されます。ローカル用アドレスは、プラグアンドプレイのローカル通信と、グローバルアドレスを使用するためのブートストラップ操作を行うために、サイト内で使用します。

ローカル用のユニキャストアドレスには、リンクローカルとサイトローカルの 2 種類があります。リンクローカル用は単一リンクで使用します。サイトローカル用は単一サイトで使用します。次の表は、リンクローカル用アドレスフォーマットを示したものです。

表 14–3 リンクローカル用アドレスフォーマット

10 ビット	54 ビット	64 ビット
1111111010	0	Interface ID

リンクローカル用アドレスは自動アドレス設定などの目的で 1 つのリンク上のアドレス指定に使用します。

表 14–4 は、サイトローカル用アドレスフォーマットです。

表 **14–4** サイトローカル用アドレス

10 ビット	38 ビット	16 ビット	64 ビット
1111111011	0	Subnet ID	Interface ID

どちらのタイプのローカルアドレスでも、インタフェース ID はそれを使用するドメインで一意な識別子である必要があります。通常は、識別子としてノードの IEEE-802 48 ビットアドレスを使用します。Subnet ID は、サイト内の特定のサブネットを識別します。Subnet ID と Interface ID を組み合わせてローカルアドレスを作成します。これで大規模なプライベートインターネットを構築することができ、その他のアドレス割り当てを行う必要はありません。

現在グローバルインターネットに接続していない組織はローカルアドレスを使用できます。ローカルアドレスを使うだけでグローバルインターネットアドレス空間からのアドレスプレフィックスを要求する必要はありません。この組織が将来インターネットに接続する場合、Subnet ID と Interface ID をグローバルプレフィックスと組み合わせてグローバルアドレスを作成することができます。たとえば、Registry ID、Provider ID、Subscriber ID の組み合わせでグローバルアドレスを作成できます。この拡張機能は IPv4 に対する大幅な改善点です。IPv4 では、プライベート (非グローバル) な IPv4 アドレスを使うサイトは、インターネットに接続する場合に手動で番号を指定し直す必要があります。IPv6 の場合、番号は自動的に指定し直されます。

組み込み IPv4 アドレスを伴った IPv6 アドレス

IPv6 移行機能では、ホストとルーターが IPv4 ルーティングインフラストラクチャのもとで IPv6 パケットを動的にトンネル処理できる方式を採用しています。この方式を利用した IPv6 ノードには、下位 32 ビットに IPv4 アドレスを保存した特別な IPv6 ユニキャストアドレスが割り当てられます。このタイプのアドレスを *IPv4* 互換 *IPv6* アドレスといいます。 次の表にそのフォーマットを示します。

表 **14–5** IPv4 互換 IPv6 アドレスフォーマット

80 ビット	16 ビット	32 ビット
0000.....0000	0000	IPv4 アドレス

組み込み IPv4 アドレスを保存する第 2 のタイプの IPv6 アドレスも定義されています。このアドレスは IPv6 アドレス領域内の IPv4 アドレスを表すときに使用します。このアドレスは主に、アプリケーション、API、オペレーティングシステムの実装内で使用します。このタイプのアドレスを *IPv4* マップ *IPv6* アドレスといいます。 次の表にそのフォーマットを示します。

表 **14–6** IPv4 マップ IPv6 アドレスフォーマット

80 ビット	16 ビット	32 ビット
0000.....0000	FFFF	IPv4 アドレス

任意キャストアドレス

IPv6 任意キャストアドレスは複数のインタフェースに割り当てるアドレスです。通常は、任意キャストアドレスは異なるノードに所属しています。任意キャストアドレスに送信されたパケットは、ルーティングプロトコルの測定距離に基づいて同じアドレスで最も近くにあるインタフェースにルーティングされます。

任意キャストアドレスはルートシーケンスの一部に使用できます。 したがって、ノードはトラフィックを搬送するインターネットサービスプロバイダを選択できます。この機能をソース選択ポリシーと呼ぶこともあります。この機能を実装するには、インターネットサービスプロバイダに所属するルーターセットを識別するように任意キャストアドレスを構成します。たとえば、インターネットサービスプロバイダごとに 1 つの任意キャストを構成します。任意キャストを、IPv6 ルーティングヘッダーで中間アドレスとして使用できます。これにより、特定のプロバイダまたは一連のプロバイダによりパケットが配信されます。また、任意キャストアドレスは、特定のサブネットに接続されたルーターセットや、特定のルーティングドメインへのエントリを提供するルーターセットの識別にも使用できます。

定義済みのユニキャストアドレスフォーマットを利用すれば、ユニキャストアドレス領域から任意キャストを指定できます。そのため、任意キャストアドレスは、構文的にはユニキャストアドレスと区別がつきません。複数のインタフェースにユニキャストアドレスを割り当てる場合は、ユニキャストアドレスを任意キャストアドレスに変換します。ただし、そのアドレスが任意キャストアドレスであることがわかるように、アドレスを割り当てるノードを明示的に構成する必要があります。

マルチキャストアドレス

IPv6 マルチキャストアドレスは、インタフェースグループの識別子です。1 つのインタフェースが所属できるマルチキャストグループは複数設定できます。表 14–7 は、マルチキャストアドレスフォーマットを示します。

表 **14–7** マルチキャストアドレスフォーマット

8 ビット	4 ビット	4 ビット	112 ビット
11111111	FLGS	SCOP	グループ ID

アドレスの先頭の 11111111 は、アドレスがマルチキャストアドレスであることを表します。FLGS は、4 つのフラグ (0、0、0、T) のセットです。

上位 3 つのフラグは、予約されており、0 に初期化されます。

- **T=0** – 固定的に割り当てられた (既知の) マルチキャストアドレスを識別する。グローバルインターネット番号指定機関が割り当てる
- **T=1** – 非固定的に割り当てられた (一時的な) マルチキャストアドレスを識別します。

SCOP は、4 ビットのマルチキャストスコープの値であり、マルチキャストグループの有効範囲を表します。表 14–8 は、SCOP の値です。

表 **14–8** SCOP の値

0	予約済み	8	組織ローカルスコープ
1	ノードローカルスコープ	9	(割り当てなし)
2	リンクローカルスコープ	A	(割り当てなし)
3	(割り当てなし)	B	(割り当てなし)
4	(割り当てなし)	C	(割り当てなし)
5	サイトローカルスコープ	D	(割り当てなし)
6	(割り当てなし)	E	グローバルスコープ
7	(割り当てなし)	F	予約済み

グループ ID は、指定スコープ内で、固定または一時的のどちらかのマルチキャストグループを識別します。

IPv6 のルーティング

IPv6 のルーティングは、CIDR における IPv4 のルーティングとほぼ同じです。唯一の違いは、IPv4 では 32 ビットアドレスを使用しますが、IPv6 では 128 ビットアドレスを使用することです。非常に簡単な拡張で、IPv4 のルーティングアルゴリズム (OSPF、RIP、IDRP、IS-IS など) をすべて IPv6 のルーティングに使用できます。

IPv6 には、新たに強力なルーティング機能をサポートした簡単なルーティング拡張機能も組み込まれました。次のリストに、新しいルーティング機能を示します。

- プロバイダ選択 (ポリシー、性能、コストなどを基準に)
- ホストの移動性 (現在の場所までのルート)
- アドレスの自動的な再指定 (新しいアドレスへのルート)

新しいルーティング機能を利用するには、IPv6 ルーティングオプションを使用する IPv6 アドレスのシーケンスを作成します。IPv6 の送信元は、ルーティングオプションでを使用して、パケットが宛先に至るまでに経由する複数の中間ノード (またはトポロジカルグループ) をリストします。この中間ノードは、パケットの宛先の途中に通過します。この機能は、IPv4 での緩やかな経路制御と記録オプションによく似ています。

アドレスシーケンスを一般的に使用する場合、通常は、ホストが受信したパケットのルートを逆戻りする必要があります。このパケットは、IPv6 認証ヘッダーを使用して正常に認証される必要があります。パケットを発信者に戻すには、アドレスシーケンスがパケット内に格納されている必要があります。IPv6 ホストの実装では、この方式により始点経路の処理と逆引きをサポートしています。始点経路の処理と逆引きは、プロバイダが新機能を実装するホストを使用するためのポイントです。新機能には、プロバイダの選択や拡張アドレスが含まれます。

IPv6 の近傍検索

IPv6 では、同じリンクに接続されたノード間の対話に関連した問題をまとめて解決しました。そのため、次のような問題を個々に解決する仕組みを定義しています。

- ルーター検索 – 接続されたリンクにあるルーターをホストが探索する
- プレフィックス探索 – どの宛先がリンクに接続されているかを定義するアドレスプレフィックスのセットをホストが探索する (オンリンクということもある)。リンクにある宛先と、ルーターからだけアクセスできる宛先を、ノードではプレフィックスで区別します。
- パラメータ探索 – ノードは、リンク MTU (最大伝送単位) などのリンクパラメータを調べる。また、出力パケットに設定するホップ限界数などのインターネットパラメータを調べる
- アドレス自動設定 – インタフェースのアドレスをノードが自動的に設定する
- アドレス解決 – 宛先の IP アドレスだけを使用してノードが近傍のリンク層アドレスを判定する (オンリンク宛先)
- 次のホップの決定 – 宛先に向かうトラフィックの送信先である近傍の IP アドレスへの IP 宛先アドレスのマッピングをアルゴリズムで判別する。次のホップはルーターまたは宛先になる
- 不到達検出 – 近傍に到達不可能であることをノードが判定する。ルーターに使用される近傍の場合、代替デフォルトルーターを試行できる。ルーターとホストの場合、アドレス解決を再試行できる
- 重複アドレス検出 – あるノードがアドレスを要求したところ、別のノードがそのアドレスを使用していないかを判別する
- リダイレクト – 特定の宛先へのアクセス手段として、最適な最初のホップノードをルーターからホストに知らせる

近傍検索では、次の 5 種類の ICMP (インターネット制御メッセージプロトコル) パケットタイプを定義します。ルーター要請メッセージとルーター通知メッセージのペア、近傍要請メッセージと近傍通知メッセージのペア、およびリダイレクトメッセージ 。これらのメッセージの目的は、次のとおりです。

- ルーター要請 – インタフェースが使用可能になると、ホストはルーター要請を送信できる。この要請は、次に予定されている時刻ではなく、ただちにルーター通知メッセージを送信するようにルーターに要求する
- ルーター通知 – ルーターはさまざまなリンクパラメータやインターネットパラメータとともにその存在を通知する。ルーターは定期的に、あるいはルーター要請メッセージに応じて通知する。ルーター通知には、オンリンク判別またはアドレス設定、あるいはホップ限界数の選択肢などに使用するプレフィックスが含まれる
- 近傍要請 – 近傍のリンク層アドレスを判定するため、および、近傍がキャッシュリンク層アドレスで到達可能かどうかを確認するためにノードによって送信される。近傍要請は重複アドレス検出にも使用する
- 近傍通知 – 近傍要請メッセージに対する応答として、ノードでは未要請の近傍通知も送信してリンク層アドレスの変更を伝える
- リダイレクト – 宛先までの最適な最初のホップ、または宛先がオンリンクであることをルーターからホストに知らせる

ルーター通知

マルチキャスト対応リンクとポイントツーポイントリンクでは、ルーターは定期的にルーター通知パケットをマルチキャストして利用できることを知らせます。ホストはすべてのルーターからルーター通知を受け取り、デフォルトルーターのリストを作成します。利用できるルーターをホストが短時間 (2 、3 分以内) に知ることができるように、ルーターは頻繁にルーター通知を生成します。ただし、通知がないからといってルーターエラーであると判断できるほどの頻度ではありません。エラー検出には、近傍到達不能性を判別する別の検出アルゴリズムを利用します。

ルーター通知プレフィックス

ルーター通知には、オンリンク判別に使用するプレフィックスリストが含まれます。このプレフィックスリストは、自動アドレス設定にも使用されます。プレフィックスに付属するフラグは特定のプレフィックスの使用目的を表します。ホストは、通知されたオンリンクプレフィックスからリストを作成し管理します。リストは、パケットの宛先がいつオンリンクになっているか、あるいはルーターを離れているかを知るために使用します。通知されたオンリンクプレフィックスになくても宛先がオンリンクの場合があります。その場合、ルーターからリダイレクトを送信して宛先が近傍であることを送信者に知らせることができます。

ルーター通知 (およびプレフィックス別のフラグ) では、ルーターからホストにアドレスの自動設定の方法を伝えることができます。たとえば、ステートフル (DHCPv6) か自動 (ステートレス) のどちらのアドレス設定を使用するかなどがあります。

ルーター通知メッセージ

ルーター通知メッセージには、ホストが出力パケットで使用する必要があるホップ限界数などのインターネットパラメータも組み込むことができます。また、オプションでリンク MTU などのリンクパラメータも組み込むことができます。この機能により、重要なパラメータの集中管理が可能になります。パラメータは、ルーターに設定され、関連付けられたすべてのホストに自動的に伝達されます。

ノードでは、宛先ノードに対してそのリンク層アドレスを戻すよう要求する近傍要請をマルチキャストしてアドレス解決を行います。近傍要請メッセージは、宛先アドレスの要請先のノードマルチキャストアドレスにマルチキャストされます。宛先は、そのリンク層アドレスをユニキャスト近傍通知メッセージで戻します。発信元と宛先の両方に対して 1 つの要求応答パケットペアで互いのリンク層アドレスを処理できます。発信元は、近傍要請に発信元のリンク層アドレスを組み込みます。

近傍要請と不到達

近傍要請メッセージでは、複数のノードに同じユニキャストアドレスが割り当てられているかを確認することもできます。

近傍不到達検出では、近傍エラーや近傍への送信パスのエラーを検出します。近傍不到達検出では、近傍に送信されるパケットがその近傍に実際にアクセスして、その IP 層で正しく処理されたかどうかを確認する肯定確認が必要です。近傍不到達検出では、2 つのソースの確認を使用します。可能な場合、上位層のプロトコルでは、接続が送信を処理中であるという肯定確認を戻します。すなわち、先に送信されたデータは正しく配信されたということが通知されます。たとえば、最も新しい TCP 肯定を受信したことが通知されます。肯定応答が得られない場合、ノードはユニキャスト近傍要請メッセージを送信します。 このメッセージは、次のホップからの到達可能確認として近傍通知を要請します。不要なネットワークトラフィックを避けるため、ノードからアクティブにパケットが送信されている近傍にだけ探査メッセージが送信されます。

上記の一般的な問題を解決する以外に、近傍検索では次のような状況にも対応します。

- リンク層アドレスの変更 – リンク層アドレスの変更を認識したノードは、少数の (非要請) 近傍通知パケットをマルチキャストできる。ノードはすべてのノードにマルチキャストして、無効になったキャッシュリンク層アドレスを更新できる。非要請通知の送信は、性能強化が目的。近傍不到達検出アルゴリズムにより、すべてのノードが確実に新しいアドレスを探索できるが、遅延が多少伸びる可能性がある
- 入力負荷均衡 – インタフェースを複製したノードでは、同じリンク上の複数のネットワークインタフェース間の入力パケットの受信の負荷均衡ができる。このようなノード間では、同じインタフェースに複数のリンク層アドレスが割り当てられる。たとえば、1 つのネットワークドライバで、複数のネットワークインタフェースカードを、複数のリンク層アドレスを持つ 1 つの論理インタフェースとして表現できる

負荷均衡は、ルーターがソースリンク層アドレスをルーター通知パケットから省略することを可能にすることで処理する。この場合、近傍では近傍要請メッセージを使用してルーターのリンク層アドレスを確認する。近傍通知メッセージの戻りには、要請元によって異なるリンク層アドレスが組み込まれる

- 任意キャストアドレス – 任意キャストアドレスは、等価サービスを提供するノードセットの1 つを識別する。同じリンクの複数のノードは同じ任意を 認識するように設定できる。近傍検索では、ノードが同じ宛先に対する複数の近傍通知を受信するようにノードを設定して任意キャストを処理する。任意キャストアドレスの通知にはすべて、取り消しできない通知としてのタグが設定される。取り消しできない通知により、複数存在する可能性がある通知の中でどれを使用するかを判定する特定の規則が呼び出される
- プロキシ通知 – 近傍要請に応答できない宛先アドレスのかわりにパケットを受信するルーターは、取り消し無効の近傍通知を発行できる。現在はプロキシの使用方法は指定されていないが、オフリンクになった移動ノードをプロキシ通知で処理できる可能性がある。ただし、プロキシは、このプロトコルを実装していないノードを処理する一般的な機構として使用されることはない

IPv4 との比較

IPv6 近傍検索プロトコルは、IPv4 プロトコル ARP (アドレス解決プロトコル)、ICMP ルーター検索、ICMP リダイレクトを組み合わせたようなものです。IPv4 には近傍不到達検出に全般的に対応できるプロトコルや機構はありませんでした。ただし、ホスト条件ではデッドゲートウェイ検出に対応できるアルゴリズムがいくつか指定されています。デッドゲートウェイ検出は、近傍不到達検出の一部です。

近傍検索プロトコルでは、IPv4 プロトコルセットに対するさまざまな強化措置が施されています。

- ルーター検索はベースプロトコルセットの一部であり、ホストがルーティングプロトコルを *snoop* する必要はない
- ルーター通知ではリンク層アドレスが伝達される。ルーターのリンク層アドレスの解決に、これ以外のパケット交換は不要
- ルーター通知ではリンクのプレフィックスが伝達される。ネットマスクを設定する独立した機構は不要
- ルーター通知では、アドレス自動設定が使用可能になる
- ルーターは、ホストがリンクで使用するMTU を通知できる。したがって、MTU が定義されていないすべてのノードはリンク上の同じ MTU 値を使用する
- アドレス解決マルチキャストは、40 億 (2^32) マルチキャストアドレスに展開され、宛先以外のノードに対するアドレス解決関係の割り込みを大幅に削減した。さらに、IPv6 以外のマシンの割り込みをなくした
- リダイレクトには、新しい最初のホップのリンク層アドレスを保存する。独立したアドレス解決がなくてもリダイレクトを受信できる

- 同じリンクに複数のプレフィックスを関連付けられる。デフォルトで、ホストはルーター通知からすべてのオンリンクプレフィックスを受け取る。ただし、ルーター通知にあるプレフィックスをすべて、あるいは一部省略するようにルーターを設定できる。その場合、ホストは宛先がオフリンクであるとみなす。その結果、ホストはルーターにトラフィックを送信する。ルーターは適宜リダイレクトを発行する
- IPv4 と異なり、IPv6 リダイレクトの受信者は新しい次のホップがオンリンクであるとみなす。IPv4 では、ホストはリダイレクトを無視し、リンクのネットワークマスクに基づいて、リンクにない次ホップを指定する。IPv6 リダイレクト機構は XRedirect 機能に似ている。リダイレクト機構は、非ブロードキャストおよび共有メディアリンクで有効。これらのリンク上では、ノードがオンリンク宛先のすべてのプレフィックスをチェックすることは望ましくない、あるいは不可能である
- 近傍不能性検出により、障害ルーターがある場合のパケット伝送能力が改善される。また、この機能により、部分的に障害があるリンクやパーティション化されたリンクを経由するパケット伝送、あるいはリンク層アドレスが変更されたノードを経由するパケット伝送が改善される。たとえば、移動ノードは、頻繁に更新される ARP キャッシュのおかげでオフリンクになっても接続が切れない
- ARP と異なり、近傍検索では、ハーフリンクエラー (近傍不能性検出を利用) を検出し、双方向接続がない近傍にトラフィックが送信されるのを防ぐ
- IPv4 ルーター検索と異なり、ルーター通知メッセージにはユーザー定義フィールドはない。安定性の異なるルーターの操作にユーザー定義フィールドは不要。近傍不能性検出で、デッドルーターを検出し、アクティブルーターに切り替えることができる
- リンクローカルアドレスでルーターを一意に識別しておけば、ホストでルーター関連付けを維持できる。ルーターを識別する機能は、ルーター通知とリダイレクトメッセージで必要とされる。サイトが新しいグローバルプレフィックスを使用しても、ホストはルーター関連付けを維持する必要がある
- 近傍検索メッセージのホップ制限は受信時に 255 なので、プロトコルがオフリンクノードによるスプーフエラーの被害を受けることがない。これに対し、IPv4 オフリンクノードではICMP (インターネット制御メッセージプロトコル) リダイレクトとルーター通知メッセージの両方を送信できる
- ICMP 層にアドレス解決を配置すると、プロトコルが ARP よりも媒体に依存しなくなる。その結果、標準 IP 認証とセキュリティ機構が使用できるようになる

IPv6 ステートレスアドレス自動設定

ホストでは、IPv6 のインタフェースの自動設定を数ステップかけて実行します。自動設定プロセスでは、リンクローカルアドレスの作成、リンク上の一意性の検査、どのような情報を自動設定するか (アドレス、その他の情報、または両方)、 アドレスをス

テートフル機構またはステートフル機構、あるいはその両方で取得するかの決定が行われます。ここでは、リンクローカルアドレスの生成手順、ステートレスアドレス自動設定によるサイトローカルアドレスとグローバルアドレスの生成手順、そして重複アドレス検出手順について説明します。

ステートレス自動設定の条件

IPv6 では、ステートフルとステートレスのアドレス自動設定機構を定義しています。ステートレス自動設定では、手動によるホストの設定は不要です。ルーターは最小限の設定 (あれば) ですみ、サーバーの追加も不要です。ステートレス機構では、ローカルに取得できる情報とルーターが通知する情報を利用してホストがそれぞれのアドレスを生成できます。ルーターはリンクに関連付けられたサブネットを識別するプレフィックスを通知します。ホストはサブネット上で一意にインタフェースを識別するインタフェース識別子を生成します。アドレスはこれらのプレフィックスとインタフェース識別子を組み合わせて作ります。ルーターがない場合、ホストはリンクローカルアドレスだけを生成します。ただし、同じリンクに接続されたノード間の通信では、リンクローカルアドレスで十分です。

ステートフル自動設定モデル

ステートフル自動設定モデルでは、ホストはインタフェースアドレスや設定情報とパラメータをサーバーから取り込みます。サーバーでは、どのホストにどのアドレスが割り当てられたかを保存したデータベースを管理します。ホストは、ステートフル自動設定プロトコルを利用してアドレスやその他の設定情報をサーバーから取り込むことができます。ステートレス自動設定とステートフル自動設定は互いに補完し合います。たとえば、ホストでは、ステートレス自動設定でアドレスを設定し、ステートレス自動設定でその他の情報を取り込みます。

ステートレス方式とステートフル方式をいつ使用するか

ホストが使用するアドレスを厳密に知る必要はない場合に、ステートレス方式を使用します。ただし、アドレスが一意で正しくルートできる必要があります。正確なアドレス割り当てに対してサイトでさらに厳しく管理する必要がある場合に、ステートフル方式を使用します。ステートフルとステートレスのどちらのアドレス自動設定も同時に使用できます。サイト管理者は、ルーター通知メッセージのフィールドの設定を通じて、どの方式の自動設定を使用するかを指定します。

IPv6 アドレスは、一定の時間 (場合によっては無限に) インタフェースにリースされます。各アドレスには、アドレスがどれだけの時間、インタフェースに割り当てられるかを示す寿命があります。寿命が尽きると、結合 (とアドレス) が無効になり、そのアドレスを別のインタフェースに割り当てることができます。アドレスの割り当ての終

了を正常に行うため、アドレスはインタフェースに割り当てられた状態で 2 つの別々のフェーズを経ます。最初、アドレスには優先権が与えられ、任意に通信ができます。次に、アドレスの現在のインタフェース割り当てが無効になるという前提から、優先順位が下がります。優先順位が低い状態で、アドレスを使用するのは避けるべきですが、使用できないわけではありません。新しい通信 (たとえば、新しい TCP 接続の開始など) ではできるだけ優先順位の高いアドレスを使用します。優先順位の低いアドレスを使用できるのは、そのアドレスを使用中のアプリケーションだけにする必要があります。サービスを打ち切らないと別のアドレスに切り替えるのが困難なアプリケーションは、優先順位の低いアドレスを使用できます。

重複アドレスの検出アルゴリズム

特定のリンク上ですべての設定済みアドレスが一意であることを保証するため、ノードは重複アドレスの検出アルゴリズムを実行します。この実行は、インタフェースにアドレスを割り当てる前に行われる必要があります。重複アドレスの検出アルゴリズムはすべてのアドレスを対象として実行されます。

このマニュアルで指定する自動設定プロセスは、ホストにだけ適用し、ルーターには適用しません。ホストの自動設定では、ルーターが通知した情報を使用するため、ルーターは別の手段で設定する必要があります。ただし、このマニュアルで説明した機構を使用して、ルーターによってリンクローカルアドレスが生成される場合があります。また、インタフェースに割り当てられる前に、すべてのアドレスにおいてルーターによる重複アドレスの検出処理が正常終了していることが望まれます。

IPv6 プロトコルの概要

ここでは、自動設定中にインタフェースが実行する通常の手順について概要を説明します。自動設定が行われるのはマルチキャスト対応リンクだけです。たとえばシステム起動時など、マルチキャスト対応インタフェースが使用可能な状態で開始します。ノード (ホストとルーターの両方) では、そのインタフェースのリンクローカルアドレスを生成して自動設定プロセスを開始します。リンクローカルアドレスは、インタフェースの識別子を既知のリンクローカルプレフィックスに追加して作成します。

ノードは、この仮リンクローカルアドレスがリンク上の別のノードで使用されていないことを確認する必要があります。この確認が終わったら、リンクローカルアドレスをインタフェースに割り当てることができます。特に、ノードは宛先が仮アドレスになっている近傍要請メッセージを送信します。別のノードがそのアドレスを使用中の場合、そのノードはそのことを伝える内容を含む近傍要請を返信します。別のノードがそのアドレスを使用しようと試みている場合、そのノードもその宛先に近傍要請を送信します。近傍要請送信や再送の数と、連続した要請間の遅延 はリンクによって異なります。これらのパラメータは、システム管理で設定できます。

ノードにおいて、仮リンクローカルアドレスが一意でないことがわかると自動設定が打ち切られるため、手動でインタフェースを設定する必要があります。この状態からの回復を簡単にするには、管理者が代替インタフェース識別子を提供してデフォルト

識別子を無効にします。これにより、新しい (一意であると考えられる) インタフェース識別子を利用して自動設定機構を実行できます。そうでなければ、リンクローカルアドレスとその他のアドレスは手動で設定します。

この仮リンクローカルアドレスが一意であると判断されると、ノードはインタフェースにそのアドレスを割り当てます。このとき、ノードは近傍ノードと IP レベルで接続されます。自動設定手順の残りは、ホストだけで実行されます。

ルーター通知の受信

自動設定の次の手順では、ルーター通知を受信するか、ルーターが存在しないことを確認します。ルーターがあれば、ホストが実行すべき自動設定の種類を指定したルーター通知が送信されます。ルーターがない場合、ステートフル自動設定が呼び出されます。

ルーターはルーター通知を定期的に送信します。ただし、連続した送信と送信の間の遅延は、自動設定を実行するホスト側の待機時間より通常は長くなります。通知を迅速に受信するため、すべてのルーターマルチキャストグループに 1 つまたは複数のルーター要請を送信します。ルーター通知には 2 つのフラグがあり、どのようなステートフル自動設定 (あれば) を実行すべきかを表します。管理アドレス設定フラグは、アドレスの取得時にホストがステートフル自動設定を使用するかどうかを表します。もう 1 つのステートフル設定フラグは、その他の情報 (アドレスを除く) の取得時にホストがステートフル自動設定を使用するかどうかを表します。

プレフィックス情報

ルーター通知にプレフィックス情報オプションがある場合、これらのオプションにはステートレスアドレス自動設定におけるサイトローカルアドレスとグローバルアドレスの生成に必要な情報を保存します。ルーター通知のステートレスアドレス自動設定フィールドとステートフルアドレス自動設定フィールドは別々に処理されます。ホストでは、ステートフルアドレス自動設定とステートレスアドレス自動設定を同時に使用できます。プレフィックス情報オプションフィールドの1 つである自動アドレス設定フラグは、オプションがステートレス自動設定にも適用されるかどうかを表します。適用される場合、補助オプションフィールドにサブネットプレフィッスと寿命値が保存されます。これらの値は、プレフィックスから作成されたアドレスがどれだけの時間優先権を持ち有効であるかを表します。

ルーターではルーター通知が定期的に生成されるので、ホストでは常に新しい通知を受信します。ホストは各通知に組み込まれた情報を上記の手順で処理し、情報を追加します。また、ホストは前の通知で受け取った情報を更新します。

アドレスの一意性

安全性確保のため、すべてのアドレスについて、インタフェースに対する割り当て前に一意かどうかが確認されます。ただし、ステートレス自動設定で作成したアドレスの場合は状況が異なります。アドレスの一意性は、インタフェース識別子から生成されるアドレスの一部で主に決まります。そのため、ノードにおいてリンクローカルア

ドレスの一意性が確認されると、同じインタフェース識別子から生成される他のアドレスの個別の確認が不要になります。ただし、手動またはステートフルアドレス自動設定で得られたアドレスはすべて、個別に一意であることを確認する必要があります。一部のサイトでは、重複アドレスの検出を実行するためのオーバーヘッドが大きく、それを実行することで得られる利益が帳消しになる場合があります。そのようなサイトでは、インタフェース別設定フラグの設定で重複アドレスの検出の使用を無効にできます。

自動設定処理を短時間で終了するために、ルーター通知の待機とリンクローカルアドレスの生成 (およびその一意性の確認) をホストで並列して実行できます。ルーターでは、ルーター要請に対する応答が数秒遅れる可能性があります。そのため、上記 2 つの手順を 1 つずつ実行すると、自動設定を完了するために必要な合計時間が大幅に長くなる可能性があります。

IPv6 モビリティ (移動性) サポート

ルーティングは、パケットの宛先 IP アドレスのサブネットプレフィックスに基づいて行われます。そのため、モバイルノード、ホストまたはルーターを宛先とするパケットは、ホームリンクに関連付けられていないノードには到達できません。ホームリンクは、ノードの IPv6 サブネットプレフィックスが存在するリンクです。ノードの移動に関係なく通信を継続するために、モバイルノードは新しいリンクに移動するたびにその IP アドレスを変更できます。ただし、モバイルノードの位置を変更すると、移動ノードではトランスポート層とその上位層の接続が失われます。以上のことから、将来、インターネットに接続するモバイルコンピュータが増加することを考えると、IPv6 モビリティサポートが大きな意味を持つことになります。

上記の問題に IPv6 モビリティサポートが対応します。IPv6 モビリティでは、モバイルノードがリンク間を移動してもその IP アドレスは変更されません。モバイルノードに対する IP アドレスの割り当ては、そのノードのホームリンク上のホームサブネットプレフィックスの範囲内で行われます。これをノードのホームアドレスといいます。

これにより、モバイルノードのホームアドレスにルートされたパケットは、モバイルノードが現在インターネットのどこに接続していても宛先にアクセスできます。モバイルノードが新しいリンクに移動しても他のノード (固定またはモバイル) との通信は途切れません。

ホームを離れたモバイルノードと送受信するパケットを透過的にルーティングする問題は IPv6 移動サポートで解決できます。しかし、モバイルコンピュータや無線ネットワークの使用に伴うすべての問題が解決されるわけではありません。特に次の問題には対処できません。

- 通常の無線ネットワークのようにアクセスできるときとできないときがあるリンクの処理。ただし、移動検出手順でいくつかの問題は処理できる
- モバイルノードが接続しているリンクのアクセス制御

IPv6 サービス品質 (QoS) 機能

ホストは 、IPv6 ヘッダーのフローラベルフィールドとトラフィッククラスフィールドを使用できます。ホストは、これらのフィールドを使用して、IPv6 ルーターによる特別処理を要求するパケットを識別します。特別処理の例としては、デフォルト以外のサービス品質やリアルタイムサービスがあります。この機能により、ある程度一貫したスループット、遅延、ジッターが必要なアプリケーションをサポートできます。この種のアプリケーションには、マルチメディアアプリケーションまたはリアルタイムアプリケーションがあります。

フローラベル

発信元では、IPv6 ヘッダーの 20 ビットのフローラベルフィールドを使用できます。送信元は、IPv6 ルーターによる特別処理を要求するパケットに、このフィールドを使用してラベルを付けます。特別処理の例としては、デフォルト以外のサービス品質やリアルタイムサービスがあります。この IPv6 の機能はまだ実験段階であり、インターネットのフローサポートの条件が確定すると変更される可能性があります。一部のホストまたはルーターではフローラベルフィールドの機能をサポートしていません。このようなホストまたはルーターでは、パケットの生成時にフローラベルフィールドをゼロに設定する必要があります。パケットが転送される場合は、フローラベルフィールドは変更されないまま転送されます。パケットを受信したホストやルーターはフローラベルフィールドを無視します。

フローとは

フローは特定の送信元から特定の (ユニキャストまたはマルチキャスト) 宛先に送信されるパケットのシーケンスです。ソースは、ルーターによる特別処理を必要とします。特別処理の特性は、制御プロトコルによってルーターに伝達される場合があります。制御プロトコルとして、リソース予約プロトコルを使用できます。 また、ホップバイホップオプションなど、フローのパケット内の情報によって伝達される場合もあります。

ソースから宛先までのアクティブフローは複数のフローであることもあれば、どのフローにも関連付けられていないトラフィックを含む場合もあります。フローの一意の識別はソースアドレスとゼロ以外のフローラベルの組み合わせによって行います。フローに所属しないパケットは、ゼロに設定されたフローラベルを運びます。

フローのソースノードでは、フローにフローラベルを割り当てます。新しいフローラベルは 16 進数で 1 から FFFFF の範囲からランダム (疑似的な) かつ均等に選択します。ランダムに割り当てることにより、ルーターはフローラベルフィールド内の任意のビットセットをハッシュキーとして利用できます。ルーターは、ハッシュキーを使ってフローに関連付けられた状態を調べることができます。

同じフローに所属するパケット

同じフローに所属するパケットは、同じソースアドレス、同じ宛先アドレス、同じゼロ以外のフローラベルで送信します。これらのパケットのどれかにホップバイホップオプションヘッダーが含まれる場合、すべてのパケットを同じホップバイホップオプションヘッダーの内容で生成する必要があります。ただし、ホップバイホップオプションヘッダーの次のヘッダーフィールドは除かれます。これらのパケットのどれかにルーティングヘッダーが含まれる場合、すべてのパケットの拡張ヘッダーを同じ内容で生成する必要があります。この同じ内容には、ルーティングヘッダーより前のすべての拡張ヘッダーと、ルーティングヘッダーが含まれます。ただし、ルーティングヘッダーの次のヘッダーフィールドは除かれます。ルーターや宛先では、場合によってはこれらの条件が満たされているかを確認できます。違反を検出した場合、そのことを送信元に報告する必要があります。違反を報告するには、ICMP パラメータ問題メッセージ、コード 0 を使用します。違反は、フローラベルフィールドの上位オクテットで表されます。この上位オクテットは、IPv6 パケット内のオフセット 1 オクテットです。

ルーターは、任意のフローのフロー処理状態を自由にセットアップできます。この場合ルーターは、制御プロトコル、ホップバイホップオプション、その他の手段による、明示的なフロー確立情報を必要としません。たとえば、未知のゼロ以外に設定されたフローラベルを持つパケットを特定のソースから受信した場合、ルーターではその IPv6 ヘッダーを処理できます。ルーターは、フローラベルがゼロに設定されている拡張ヘッダーを処理する場合と同じ方法で、必要な拡張ヘッダーを処理できます。ルーターは、次中継点のインタフェースの判別を行います。場合によってはホップバイホップオプションの更新、ルーティングヘッダーのポインタとアドレスの加算、あるいはパケットのキューイングの方法の決定なども行います。パケットのキューイングの方法の決定は、パケットのトラフィッククラスフィールドに基づいて行われます。ルーターは、これらの処理手順の結果を記憶することを選択できます。そして、記憶した後でその情報をキャッシュに保存できます。始点アドレスとフローラベルがキャッシュキーとして使用されます。同じ始点アドレスとフローラベルを持つ後続のパケット については、キャッシュされた情報を参照することにより処理できます。これらのパケットの始点アドレスとフローラベルをすべて調べる必要はありません。ルーターは、フローの最初のパケットは確認しますが、その後はフィールドの内容は変更されないと仮定することができます。

トラフィッククラス

パケットを生成したノードは、IPv6 パケットの異なるクラスまたは優先順位を識別する必要があります。その場合、IPv6 ヘッダーのトラフィッククラスフィールドが使用されます。パケットを転送するルーターも同じ目的でトラフィッククラスフィールドを使用します。

トラフィッククラスフィールドには、以下の一般的な要件が適用されます。

- 1 つのノード内の IPv6 サービスへのサービスインタフェースは、上位層プロトコルに対して、トラフィッククラスビットの値を提供する必要があります。上位層プロトコルで生成されたパケットにはトラフィッククラスビットが必要です。デフォルト値は、8 ビットすべてが 0 です。

- 一部またはすべてのトラフィッククラスビットをサポートするノードは、ビットの値を変更することができます。変更できるのは、サポートする特定の使用方法に従ってそのノードが生成、転送、または受信するパケット内のビットの値です。ノードは、特定の使用方法をサポートしないすべてのトラフィッククラスフィールド内のビットを無視し、変更しないようにしなければなりません。
- 受信パケット内のトラフィッククラスビットは、そのパケットの発信元が送信した値とは異なる値である可能性があります。したがって、上位層プロトコルは、トラフィッククラスビットの値が同じであると仮定することはできません。

IPv6 セキュリティの強化

現在のインターネットには多くのセキュリティ問題があります。インターネットでは、アプリケーション層より下の層には有効な機密機構や認証機構がありません。この欠点に対し、IPv6 では、セキュリティサービスを提供する 2 つの統合オプションを設けて対応しています。この 2 つのオプションは、別々に、あるいはまとめて使用してさまざまなユーザーにさまざまなセキュリティレベルを提供できます。ユーザー通信が異なれば、セキュリティのニーズも異なります。

最初のオプションは IPv6 認証ヘッダー (AH) と呼ばれる拡張ヘッダーです。この拡張ヘッダーは、IPv6 データグラムに機密性を持たない認証と完全性を提供します。この拡張機能はアルゴリズムに依存せず、さまざまな認証方式をサポートします。認証ヘッダーは、ワールドワイドなインターネット内の相互運用性の保証を支援するために、それを使用することが提案されています。認証ヘッダーを使用することにより、ホストなりすまし攻撃など、主なネットワーク侵害を回避できます。IPv6 でソースルーティングを使用する場合、IP ソースルーティングに明らかな危険性があるので IPv6 認証ヘッダーが重要になります。上位層プロトコルおよび上位層サービスには、現在有効な保護策はありません。しかし、インターネット層に認証ヘッダーを使用することで、ホスト発信元認証を提供できます。

2 番めのオプションである、IPv6 カプセル化セキュリティペイロード (ESP) と呼ばれる拡張ヘッダーは、IPv6 データグラムに完全性と機密性を提供します。同種のセキュリティプロトコル (SP3D、ISO NLSP) に比べて単純ですが、柔軟性があり、アルゴリズムに依存しません。

IPv6 認証ヘッダーと IPv6 カプセル化セキュリティヘッダーは、新しいインターネットプロトコルセキュリティ (IPsec) の機能です。IPsec の概要については、第 19 章を参照してください。IPsec の実装方法については、第 20 章を参照してください。

第 15 章

IPv6 の管理 (手順)

この章では、IPv6 や IPv6 ルーターを有効にする方法、IPv6 アドレスを DNS、NIS、NIS+ 用に設定する方法、ルーター間のトンネルの作成方法、診断情報を表示する IPv6 の追加コマンドを実行する方法、IPv6 ネームサービス情報の表示方法について説明します。

この章では、以下の内容について説明します。

トピック	インフォメーション
IPv6 の概要	第 14 章
IPv4 から IPv6 への移行	第 17 章
この章で説明する手順に関する概念情報	第 16 章

IPv6 ノードを有効にする

この節では、IPv6 ノードをネットワークで設定するときに必要な手順について説明します。

注 – この節で「ノード」という用語は、Solaris サーバーまたはクライアントワークステーションを指します。

IPv6 ノードを有効にする (作業マップ)

表 **15–1** IPv6 ノードを有効にする (作業マップ)

タスク	説明	参照先
ノード上の IPv6 を有効にする	`hostname6.`*interface* ファイルの操作、アドレスの表示、`/etc/inet/ipnodes` ファイルへのアドレスの入力(「注」参照)	308 ページの「ノード上の IPv6 を有効にする方法」
Solaris IPv6 ルーターの設定	`indp.conf` ファイルへのエントリの追加	309 ページの「Solaris IPv6 ルーターの設定方法」
IPv6 アドレスを NIS と NIS+ に追加	`/etc/ipnodes` ファイルへのエントリ追加	310 ページの「NIS と NIS+ に対する IPv6 アドレスの追加方法」
IPv6 アドレスを DNS に追加	DNS ゾーンと逆ゾーンファイルに対する AAAA レコードの追加	311 ページの「DNS に対する IPv6 アドレスの追加方法」

注 – IPv6 は、Solaris ソフトウェアをインストールするときにシステムで有効にできます。インストールプロセスで `yes` と応答して IPv6 を有効にすると、あとの IPv6 を有効にする手順を省略できます。

▼ ノード上の IPv6 を有効にする方法

1. **IPv6** を有効にしたいシステム上でスーパーユーザーになります。
2. コマンド行で、各インタフェースに対して次のように入力します。

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

   interface `le0`、`le1` などのインタフェース名

3. リブートします。

注 – リブートすると、ルーター検索パケットが送信されます。ルーターがプレフィックスを応答することにより、ノードが IP アドレスでインタフェースを設定できるようになります。リブートすると、主なネットワークデーモンも IPv6 モードで再起動します。

4. コマンド行で次のコマンドを入力して **IPv6** アドレスを表示します。

```
# ifconfig -a
```

5. 適切なネームサービスに、**IPv6** アドレスを次のように追加します。
   - NIS と NIS+ については、310 ページの「NIS と NIS+ に対する IPv6 アドレスの追加方法」を参照してください。
   - DNS については、311 ページの「DNS に対する IPv6 アドレスの追加方法」を参照してください。

▼ Solaris IPv6 ルーターの設定方法

1. ルーターとして機能するシステム上で、スーパーユーザーになります。
2. **/etc/inet/ndpd.conf** ファイルを編集して、サブネットプレフィックスを使用して次のエントリを **1** つまたは複数追加します。

   変数と使用できる値のリストについては、`in.ndpd(1M)` のマニュアルページを参照してください。`ndpd.conf` ファイルについては、`ndpd.conf(4)` のマニュアルページを参照してください。

   a. すべてのインタフェースについて、ルーター動作を指定するエントリを追加します。

      `ifdefault` *variable value*

   b. プレフィックス通知のデフォルト動作を指定するエントリを追加します。

      `prefixdefault` *variable value*

   c. インタフェースパラメータごとのセットエントリを追加します。

      `if` *interface variable value*

   d. インタフェースプレフィックス情報ごとの通知エントリを追加します。

      `prefix` *prefix/length interface variable value*

3. リブートします。

注 – ホストのサブネットアドレスプレフィックスが、近傍検索 (in.ndpd) からホストにリレーされます。また、次世代 RIP ルーティングプロトコル (in.ripngd) が自動的に実行されます。

例 – ndpd.conf ルーター設定ファイル

```
# Send router advertisements out all NICs
ifdefault AdvSendAdvertisements on
# Advertise a global prefix and a
# site local prefix on three interfaces.
# 0x9255 = 146.85
prefix 2:0:0:9255::0/64          hme0
prefix fec0:0:0:9255::0/64      hme0
# 0x9256 = 146.86
prefix 2:0:0:9256::0/64      hme1
prefix fec0:0:0:9256::0/64     hme1
# 0x9259 = 146.89
prefix 2:0:0:9259::0/64         hme2
prefix fec0:0:0:9259::0/64     hme2
```

▼ NIS と NIS+ に対する IPv6 アドレスの追加方法

NIS+ 用に ipnodes.org_dir という新しいテーブルが追加されました。このテーブルには、ホスト用の IPv4 アドレスと IPv6 アドレスの両方が保存されています。既存の hosts.org_dir テーブルは IPv4 情報だけを保存していますが、既存のアプリケーションが動作するように変更されていません。hosts.org_dir テーブルと ipnodes.org_dir テーブルはどちらも IPv4 アドレスと整合させておく必要があります。概要については、338 ページの「Solaris ネームサービスに対する IPv6 拡張機能」を参照してください。

新しい ipnodes.org_dir テーブルの管理方法は、hosts.org_dir の管理方法と似ています。従来の NIS+ テーブルの管理に使用したのと同じツール、ユーティリティが ipnodes.org_dir にも有効です。NIS+ テーブルの操作についての詳細は、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

次の手順では、/etc/inet/ipnodes のエントリを ipnodes.org_dir テーブルに (詳細モードで) マージします。NIS+ テーブルは、nistbladm(1)、nissetup (1M)、または nisserver(1M) のどれかで作成されたものとします。

- コマンド行で、次のコマンドを入力します。

```
% nisaddent -mv -f /etc/inet/ipnodes ipnodes
```

ipnodes.org_dir テーブルを表示するには、次のように操作します。

- コマンド行で、次のコマンドを入力します。

```
% nisaddent -d ipnodes
```

NIS 用に、2 つの新しいマップが追加されました。ipnodes.byname と ipnodes.byaddr です。これらのマップは、いずれも IPv4 と IPv6 のホスト名とアドレスの関連付けを保存しています。hosts.byname マップと hosts.byaddr マップは、IPv4 のホスト名とアドレスの関連情報だけを保存していますが、既存のアプリケーションが動作できるように変更されていません。新しいマップの管理は、以前の hosts.byname マップと hosts.byaddr マップの管理方法と同様です。*hosts* マップを IPv4 アドレスで更新すると、新しい ipnode マップも同じ情報で更新されることに注意してください。

注 – IPv6 対応ツールは、新しい NIS マップと新しい NIS+ テーブルを使用します。

▼ DNS に対する IPv6 アドレスの追加方法

1. **DNS** があるシステム上でスーパーユーザーになります。
2. **DNS** ゾーンファイルに、**IPv6** 有効化ホストの **AAAA** レコードを次のフォーマットで追加して編集します。

   host-name IN AAAA *host-address*

3. **DNS** 逆ゾーンファイルを編集し、次のフォーマットで **PTR** レコードを追加します。

   host-address IN PTR *host-name*

AAAA レコードと PTR レコードの詳細については、RFC 1886 を参照してください。

例 – DNS ゾーンファイル

```
vallejo        IN        AAAA   2::9256:a00:20ff:fe12
IN        AAAA   fec0::9256:a00:20ff:fe12:528
```

例 – DNS 逆ゾーンファイル

```
$ORIGIN    ip6.int.
8.2.5.0.2.1.e.f.f.f.9.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0 \
    IN        PTR        vallejo.Eng.apex.COM.
8.2.5.0.2.1.e.f.f.f.9.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.c.e.f \
    IN        PTR        vallejo.Eng.apex.COM.
```

IPv6 の監視

次のコマンドは IPv6 の Solaris 実装に対応するように変更されています。

- `ifconfig(1M)`
- `netstat(1M)`
- `snoop(1M)`
- `ping(1M)`
- `traceroute(1M)`

追加コマンドを使用すると診断を実行できます。これらのコマンドの考え方については、327 ページの「`ifconfig` ユーティリティに対する IPv6 拡張機能」と 334 ページの「既存のユーティリティに対する IPv6 拡張機能」を参照してください。

IPv6 の監視 (作業マップ)

表 **15–2** IPv6 の監視 (作業マップ)

タスク	説明	参照先
インタフェースアドレス割り当ての表示	`ifconfig` コマンドで、すべてのアドレス割り当て、または IPv4 か IPv6 アドレス割り当てだけを表示	313 ページの「インタフェースアドレス割り当ての表示方法」
ネットワーク状態の表示	すべてのソケットとルーティングテーブルエントリ、IPv4 用の inet アドレスファミリー、IPv6 用の inet6 アドレスファミリー、`netstat` コマンドによるインタフェースの IPv6 または ICMPv6 カウンタの統計を表示	314 ページの「ネットワーク状態の表示方法」
IPv6 関連コマンドの出力表示の制御	`ping` コマンド、`netstat` コマンド、 `ifconfig` コマンド、`traceroute` コマンドの出力の制御。`inet_type` という名前のファイルの作成と、そのファイル内の DEFAULT_IP 変数の設定	317 ページの「IPv6 関連コマンドの出力表示の制御方法」
IPv6 ネットワークトラフィックだけの監視	`snoop` コマンドによるすべての IPv6 パケットの表示	318 ページの「IPv6 ネットワークトラフィックの監視方法」
すべてのマルチホームホストアドレスの探査	`ping` コマンドによるすべてのアドレスの確認	319 ページの「すべてのマルチホームホストアドレスの探査方法」
すべてのルートのトレース	`traceroute` コマンドの使用	319 ページの「すべてのルーターのトレース方法」

▼ インタフェースアドレス割り当ての表示方法

IPv4 や IPv6 のアドレス割り当ての場合だけでなく、すべてのアドレス割り当てを表示する場合も `ifconfig` コマンドを使用します。

- コマンド行で次のコマンドを入力します。

```
% ifconfig [option]
```

`ifconfig` コマンドの詳細については、`ifconfig(1M)` のマニュアルページを参照してください。

例 – すべてのインタフェースについてアドレス指定情報を表示

```
% ifconfig -a
lo0: flags=1000849 mtu 8232 index 1
        inet 120.10.0.1 netmask ff000000
le0: flags=1000843 mtu 1500 index 2
        inet 120.46.86.54 netmask ffffff00 broadcast 120.146.86.255
        ether 8:0:73:56:a8
lo0: flags=2000849 mtu 8252 index 1
        inet6 ::1/128
le0: flags=2000841 mtu 1500 index 2
        ether 8:0:20:56:a8
        inet6 fe80::a00:fe73:56a8/10
le0:1: flags=2080841 mtu 1500 index 2
        inet6 fec0::56:20ff:fe73:56a8/64
le0:2: flags=2080841 mtu 1500 index 2
        inet6 2::56:a00:fe73:56a8/64
```

例 – すべての IPv4 インタフェースについてアドレス指定情報を表示

```
% ifconfig -a4
lo0: flags=1000849 mtu 8232 index 1
        inet 120.10.0.1 netmask ff000000
le0: flags=1000843 mtu 1500 index 2
        inet 120.46.86.54 netmask ffffff00 broadcast 120.46.86.255
        ether 8:0:20:56:a8
```

例 – すべての IPv6 インタフェースについてアドレス指定情報を表示

```
% ifconfig -a6
lo0: flags=2000849 mtu 8252 index 1
        inet6 ::1/128
```

```
le0: flags=2000841 mtu 1500 index 2
        ether 8:0:20:56:a8
        inet6 fe80::a00:fe73:56a8/10
le0:1: flags=2080841 mtu 1500 index 2
        inet6 fec0::56:20ff:fe73:56a8/64
le0:2: flags=2080841 mtu 1500 index 2
        inet6 2::56:a00:fe73:56a8/64
```

▼ ネットワーク状態の表示方法

次の手順では、 `netstat` コマンドで、次に示すネットワークデータ構造フォーマットを表示できます。

- すべてのソケットとルーティングテーブルのエントリ
- IPv4 用の inet アドレスファミリ
- IPv6 用の inet6 アドレスファミリ
- インタフェース別統計 – IPv6/ICMPv6 カウンタ

● コマンド行で次のコマンドを入力します。

```
% netstat [option]
```

`netstat` コマンドの詳細については、`netstat(1M)` のマニュアルページを参照してください。

例 – すべてのソケットとルーティングテーブルエントリの表示

```
% netstat -a
UDP: IPv4
   Local Address         Remote Address     State
-------------------- -------------------- -------
      *.*                                  Unbound
      *.apexrpc                             Idle
      *.*                                  Unbound
            .
            .
UDP: IPv6
   Local Address                     Remote Address                   State
If
--------------------------------- ---------------------------------- -------
      *.*                                                           Unbound
      *.time                                                        Idle
      *.echo                                                        Idle
      *.discard                                                     Idle
      *.daytime                                                     Idle
      *.chargen                                                     Idle

TCP: IPv4
```

```
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*                0      0     0      0 IDLE
      *.apexrpc            *.*                0      0     0      0 LISTEN
      *.*                  *.*                0      0     0      0 IDLE
      *.ftp                *.*                0      0     0      0 LISTEN
localhost.427              *.*                0      0     0      0 LISTEN
      *.telnet             *.*                0      0     0      0 LISTEN
tn.apex.COM.telnet is.Eng.apex.COM         8760      0   8760     0 ESTABLISHED
tn.apex.COM.33528 np.apex.COM.46637        8760      0   8760     0 TIME_WAIT
tn.apex.COM.33529 np.apex.COM.apexrpc      8760      0   8760     0 TIME_WAIT
TCP: IPv6
   Local Address     Remote Address   Swind Send-Q  Rwind Recv-Q   State   If
----------------- -----------------   ----- ------  ----- ------   -----
      *.*                  *.*             0      0      0      0   IDLE
      *.ftp                *.*             0      0      0      0   LISTEN
      *.telnet             *.*             0      0      0      0   LISTEN
      *.shell              *.*             0      0      0      0   LISTEN
      *.smtp               *.*             0      0      0      0   LISTEN
        .
        .
 2::56:8.login     something.1023    8640      0   8640      0 ESTABLISHED
 fe80::a:a8.echo   fe80::a:89        8640      0   8640      0 ESTABLISHED
 fe80::a:a8.ftp    fe80::a:90        8640      0   8640      0 ESTABLISHED
```

例 – IPv4 用の inet アドレスファミリを表示

```
% netstat -f inet
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
tn.apex.COM.telnet    is.apex.COM.35388    8760      0  8760   0  ESTABLISHED
tn.apex.COM.1022      alive-v4.nfsd        8760      0  8760   0  ESTABLISHED
tn.apex.COM.1021      sl.apex.COM.nfsd     8760      0  8760   0  ESTABLISHED
.
.
tn.apex.COM.33539     np.apex.COM.apexrpc  8760      0  8760   0  TIME_WAIT
```

例 – IPv6 用の inet6 アドレスファミリを表示

```
% netstat -f inet6
TCP: IPv6
   Local Address        Remote Address     Swind Send-Q Rwind Recv-Q   State      If
-------------------- -------------------- ----- ------ ----- ------ ----------- -----
2::56:a8.login        something.1023       8640      0  8640      0 ESTABLISHED
fe80::a0:a8.echo      fe80::a0:de.35389    8640      0  8640      0 ESTABLISHED
.
.
fe80::a0:a8.ftp-data  fe80::a0:de.35394   25920      0 25920      0 TIME_WAIT
```

例 – インタフェース別統計を表示 - IPv6 / ICMPv6 カウンタ

```
% netstat -sa
RAWIP
        rawipInDatagrams    =  1407     rawipInErrors       =     0
        rawipInCksumErrs    =     0     rawipOutDatagrams   =     5
        rawipOutErrors      =     0

UDP
        udpInDatagrams      =  7900     udpInErrors         =     0
        udpOutDatagrams     =  7725     udpOutErrors        =     0

TCP     tcpRtoAlgorithm     =     4     tcpRtoMin           =   200
        tcpRtoMax           = 60000     tcpMaxConn          =    -1
                .
                .
IPv4    ipForwarding        =     2     ipDefaultTTL        =   255
        ipInReceives        =406345     ipInHdrErrors       =     0
        ipInAddrErrors      =     0     ipInCksumErrs       =     0
                .
                .
IPv6 for lo0
        ipv6Forwarding      =     2     ipv6DefaultHopLimit =     0
        ipv6InReceives      =     0     ipv6InHdrErrors     =     0
                .
                .
IPv6 for le0
        ipv6Forwarding      =     2     ipv6DefaultHopLimit =   255
        ipv6InReceives      =   885     ipv6InHdrErrors     =     0
                .
                .
IPv6    ipv6Forwarding      =     2     ipv6DefaultHopLimit =   255
        ipv6InReceives      =   885     ipv6InHdrErrors     =     0
                .
                .
ICMPv4  icmpInMsgs          =   618     icmpInErrors        =     0
        icmpInCksumErrs     =     0     icmpInUnknowns      =     0
        icmpInDestUnreachs  =     5     icmpInTimeExcds     =     0
                .
                .
ICMPv6 for lo0
        icmp6InMsgs         =     0     icmp6InErrors       =     0
        icmp6InDestUnreachs =     0     icmp6InAdminProhibs =     0
                .
                .
ICMPv6 for le0
        icmp6InMsgs         =   796     icmp6InErrors       =     0
        icmp6InDestUnreachs =     0     icmp6InAdminProhibs =     0
        icmp6InTimeExcds    =     0     icmp6InParmProblems =     0
                .
                .
ICMPv6  icmp6InMsgs         =   796     icmp6InErrors       =     0
        icmp6InDestUnreachs =     0     icmp6InAdminProhibs =     0
```

```
                .
                .
IGMP:
       2542 messages received
          0 messages received with too few bytes
          0 messages received with bad checksum
       2542 membership queries received
                .
                .
```

▼ IPv6 関連コマンドの出力表示の制御方法

netstat コマンドと ifconfig コマンドの出力は制御できます。まず、/etc/default ディレクトリで inet_type という名のファイルを作成します。次に、DEFAULT_IP 変数の値を指定します。inet_type の詳細については、inet_type(4) のマニュアルページを参照してください。

1. **/etc/default/inet_type** ファイルを作成します。
2. 必要に応じて、次のいずれかのエントリを作成します。
   - IPv4 情報だけを表示するには、次のように入力します。

     ```
     DEFAULT_IP=IP_VERSION4
     ```
   - IPv4 情報と IPv6 情報を表示するには、次のいずれかを入力します。

     ```
     DEFAULT_IP=BOTH
     ```

     または、

     ```
     DEFAULT_IP=IP_VERSION6
     ```

注 – ifconfig の -4 フラグと -6 フラグの設定は、inet_type ファイルに設定された値より優先します。また、netstat の -f フラグの設定も、inet_type ファイルに設定された値より優先します。

例 – IPv4 情報と IPv6 情報を選択する出力の制御

- DEFAULT_IP=BOTH または DEFAULT_IP=IP_VERSION6 変数を inet_type ファイルで設定する場合、次の結果が得られます。

```
% ifconfig -a
lo0: flags=1000849 mtu 8232 index 1
        inet 120.10.0.1 netmask ff000000
le0: flags=1000843 mtu 1500 index 2
        inet 120.46.86.54 netmask ffffff00 broadcast 120.46.86.255
        ether 8:0:20:56:a8
lo0: flags=2000849 mtu 8252 index 1
```

```
        inet6 ::1/128
le0: flags=2000841 mtu 1500 index 2
        ether 8:0:20:56:a8
        inet6 fe80::a00:fe73:56a8/10
le0:1: flags=2080841 mtu 1500 index 2
        inet6 fec0::56:a00:fe73:56a8/64
le0:2: flags=2080841 mtu 1500 index 2
        inet6 2::56:a00:fe73:56a8/64
```

- DEFAULT_IP=IP_VERSION4 変数を inet_type ファイルで設定する場合、次の結果が得られます。

```
% ifconfig -a
lo0: flags=849 mtu 8232
        inet 120.10.0.1 netmask ff000000
le0: flags=843 mtu 1500
        inet 120.46.86.54 netmask ffffff00 broadcast 120.46.86.255
        ether 8:0:20:56:a8
```

▼ IPv6 ネットワークトラフィックの監視方法

すべての IPv6 パケットを表示するためには、次のように snoop コマンドを実行します。

1. スーパーユーザーになります。
2. コマンド行で次のコマンドを入力します。

```
# snoop ip6
```

snoop コマンドの詳細については、snoop(1M) のマニュアルページを参照してください。

例 – IPv6 ネットワークトラフィックだけの表示

```
# snoop ip6
Using device /dev/le (promiscuous mode)
fe80::a0:a1 -> ff02::9     IPv6  S=fe80::a0:a1 D=ff02::9 LEN=892
fe80::a0:de -> fe80::a0:a8 IPv6  S=fe80::a0:de D=fe80::a0:a8 LEN=104
fe80::a0:a8 -> fe80::a0:de IPv6  S=fe80::a0:a8 D=fe80::a0:de LEN=104
fe80::a0:a1 -> ff02::9     IPv6  S=fe80::a0:a1 D=ff02::9 LEN=892
fe80::a0:de -> fe80::a0:a8 IPv6  S=fe80::a0:de D=fe80::a0:a8 LEN=104
fe80::a0:a8 -> fe80::a0:de IPv6  S=fe80::a0:a8 D=fe80::a0:de LEN=152
fe80::a0:a1 -> ff02::9     IPv6  S=fe80::a0:a1 D=ff02::9 LEN=892
fe80::a0:de -> fe80::a0:a8 IPv6  S=fe80::a0:de D=fe80::a0:a8 LEN=72
fe80::a0:a8 -> fe80::a0:de IPv6  S=fe80::a0:a8 D=fe80::a0:de LEN=72
fe80::a0:a8 -> fe80::a0:de IPv6  S=fe80::a0:a8 D=fe80::a0:de LEN=72
fe80::a0:de -> fe80::a0:a8 IPv6  S=fe80::a0:de D=fe80::a0:a8 LEN=72
```

▼ すべてのマルチホームホストアドレスの探査方法

この操作では、ping コマンドですべてのアドレスを調べます。

- コマンド行で次のコマンドを入力します。

```
% ping -a ipng11
ipng11 (2::102:a00:fe79:19b0) is alive
ipng11 (fec0::102:a00:fe79:19b0) is alive
ipng11 (190.68.10.75) is alive
```

ping コマンドについての詳細は、ping(1M) のマニュアルページを参照してください。

▼ すべてのルーターのトレース方法

この操作では、traceroute コマンドですべてのルーターを調べます。

- コマンド行で次のコマンドを入力します。

```
% traceroute -a <hostname>
```

traceroute コマンドの詳細については、traceroute(1M) のマニュアルページを参照してください。

例 – すべてのルーターのトレース

```
% traceroute -a ipng11
traceroute: Warning: Multiple interfaces found; using 2::56:a0:a8 @ le0:2
traceroute to ipng11 (2::102:a00:fe79:19b0),30 hops max, 60 byte packets
 1  ipng-rout86 (2::56:a00:fe1f:59a1)  35.534 ms  56.998 ms *
 2  2::255:0:c0a8:717  32.659 ms  39.444 ms *
 3  ipng61.Eng.apex.COM (2::103:a00:fe9a:ce7b)  401.518 ms  7.143 ms *
 4  ipng12-00 (2::100:a00:fe7c:cf35)  113.034 ms  7.949 ms *
 5  ipng11 (2::102:a00:fe79:19b0)  66.111 ms *  36.965 ms

traceroute: Warning: Multiple interfaces found; using fec0::56:a8 @ le0:1
traceroute to ipng11 (fec0::10:b0), 30 hops max, 60 byte packets
 1  ipng-rout86 (fec0::56:a00:fe1f:59a1) 96.342 ms  78.282 ms 88.327 ms
 2  ipng8-tun1 (fec0::25:0:0:c0a8:717)  268.614 ms  508.416 ms  438.774 ms
 3  ipng61.Eng.apex.COM (fec0::103:a00:fe9a:ce7b)  6.356 ms * 713.166 ms
 4  ipng12-00 (fec0::100:a00:fe7c:cf35)  7.409 ms * 122.094 ms
 5  ipng11 (fec0::102:a00:fe79:19b0)  10.620 ms * *

traceroute to ipng11.eng.apex.com (190.68.10.75),30 hops max,40 byte packets
 1  rmpj17c-086.Eng.apex.COM (120.46.86.1)  4.360 ms  3.452 ms  3.479 ms
 2  flrmpj17u.Eng.apex.COM (120.46.17.131)  4.062 ms  3.848 ms  3.505 ms
 3  ipng8.Eng.apex.COM (120.68.7.23)  4.773 ms *  4.294 ms
 4  ipng61.Eng.apex.COM (120.68.10.104)  5.128 ms  5.362 ms *
 5  ipng12-20.Eng.apex.COM (120.68.10.62)  7.298 ms  5.444 ms *
```

```
6  ipng11.Eng.apex.COM (120.68.10.75)  8.053 ms  6.394 ms *
```

IPv4 トンネルによる IPv6 の設定

ここでは、IPv4 トンネル経由で IPv6 を設定する方法について説明します。

トンネルの概念については、337 ページの「IPv6 の Solaris トンネルインタフェース」と 346 ページの「トンネル機構」を参照してください。

▼ IPv4 トンネルによる IPv6 の設定方法

1. スーパーユーザーになります。
2. **`/etc/hostname6.ip.tun`***n* ファイルを作成します。*n* には **0**、**1**、**2** などの値を使用します。次に、以下の手順に従って、エントリを追加します。

   a. トンネルソースアドレスとトンネル宛先アドレスを追加します。

      `tsrc` *IPv4-source-addr* `tdst` *IPv4-destination-addr* `up`

   b. **(**省略可能**)** ソース **IPv6** アドレスと宛先 **IPv6** アドレスの論理インタフェースを追加します。

      `addif` *IPv6-source-address IPv6-destination-address* `up`

      このインタフェースに対してアドレスを自動設定したい場合は、この手順を省きます。各トンネルに対するリンクローカルアドレスを設定する必要はありません。リンクローカルアドレスは自動的に設定されます。

トンネルを設定したあと、リブートしてください。

注 **–** 双方向通信を実現するには、トンネルのもう一方の端についても同じ手順を行う必要があります。

使用するシステムをルーターとして設定する場合、リブートする前にトンネルインタフェースに通知するようにルーターを設定する必要もあります。321 ページの「トンネルインタフェースで通知するためのルーターの設定方法」 を参照してください。

例 — IPv6 アドレスを自動設定するための IPv6 設定ファイルのエントリ

次に、すべての IPv6 アドレスが自動設定されるトンネルの例を示します。

```
tsrc 129.146.86.138 tdst 192.168.7.19 up
```

例 — 手動で設定されたアドレスの IPv6 設定ファイルのエントリ

次に、グローバルソースアドレスとグローバル宛先アドレスが手動で設定されるトンネルの例を示します。サイトローカルソースアドレスとサイトローカル宛先アドレスも手動で設定されます。

```
tsrc 120.46.86.138 tdst 190.68.7.19 up
addif fec0::1234:a00:fe12:528 fec0::5678:a00:20ff:fe12:1234 up
addif 2::1234:a00:fe12:528 2::5678:a00:20ff:fe12:1234 up
```

▼ トンネルインタフェースで通知するためのルーターの設定方法

トンネルごとに次の操作をします。

1. スーパーユーザーになります。
2. **/etc/inet/ndpd.conf** ファイルを編集します。次の手順に従って、エントリを追加します。
   a. トンネルインタフェース経由のルーター通知を有効にします。

      ```
      if ip.tun n AdvSendAdvertisements 1
      ```

   b. 必要に応じてプレフィックスを追加します。

      ```
      prefix interface-address ip.tun n
      ```

3. リブートします。

IPv6 ネームサービス情報の表示

ここでは、IPv6 ネームサービス情報を表示する手順について説明します。

IPv6 ネームサービス情報を表示する (作業マップ)

表 **15–3** IPv6 ネームサービス情報を表示する (作業マップ)

タスク	説明	参照先
IPv6 ネームサービス情報の表示	`nslookup` コマンドで、IPv6 ネームサービス情報を表示する	322 ページの「IPv6 ネームサービス情報の表示方法」
DNS IPv6 PTR レコードの正確な更新の確認	`nslookup` コマンドで DNS IPv6 PTR レコードを表示する。また、`set q=PTR` パラメータを使用する	323 ページの「DNS IPv6 PTR レコードの正確な更新の確認方法」
NIS+ による IPv6 情報の表示	`ypmatch` コマンドで、IPv6 情報を NIS から表示する	323 ページの「NIS による IPv6 情報の表示方法」
NIS+ による IPv6 情報の表示	`nismatch` コマンドを実行して NIS+ で IPv6 情報を表示する	324 ページの「NIS+ による IPv6 情報の表示方法」
ネームサービスに依存しない IPv6 情報の表示	`getent` コマンドで IPv6 情報を表示する	324 ページの「ネームサービスに依存しない IPv6 情報の表示方法」

▼ IPv6 ネームサービス情報の表示方法

`nslookup` コマンドで IPv6 ネームサービス情報を表示するには、次のように操作します。

1. コマンド行で次のコマンドを入力します。

   ```
   % /usr/sbin/nslookup
   ```

   デフォルトサーバー名とアドレスが表示され、`nslookup` コマンドの山括弧 (>) プロンプトが表示されます。

2. 特定のホストの情報を表示するには、山括弧プロンプトに次のコマンドを入力します。

   ```
   >set q=any
   >host-name
   ```

3. **AAAA** レコードだけを表示するには、山括弧プロンプトに次のコマンドを入力します。

   ```
   >set q=AAAA
   ```

4. **exit** を入力して、コマンドを終了します。

例 – nslookup による IPv6 情報の表示

```
%  /usr/sbin/nslookup
Default Server:  space1999.Eng.apex.COM
Address:  120.46.168.78
```

```
> set q=any
> vallejo
Server:  space1999.Eng.apex.COM
Address:  120.46.168.78

vallejo.ipv6.eng.apex.com      IPv6 address = fec0::9256:a00:fe12:528
vallejo.ipv6.eng.apex.com      IPv6 address = 2::9256:a00:fe12:528
> exit
```

▼ DNS IPv6 PTR レコードの正確な更新の確認方法

nslookup コマンドを使用して DNS IPv6 PTR レコードを表示します。

1. コマンド行で次のコマンドを入力します。

   ```
   % /usr/sbin/nslookup
   ```

   デフォルトサーバー名とアドレスが表示され、nslookup コマンドの山括弧プロンプトが表示されます。

2. **PTR** レコードを表示するには、山括弧プロンプトに次のコマンドを入力します。

   ```
   >set q=PTR
   ```

3. **exit** を入力して、コマンドを終了します。

例 – nslookup による PTR レコードの表示

```
%  /usr/sbin/nslookup
Default Server:  space1999.Eng.apex.COM
Address:  120.46.168.78
> set q=PTR
> 8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int

8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int name =
vallejo.ipv6.Eng.apex.COM
ip6.int nameserver = space1999.Eng.apex.COM
> exit
```

▼ NIS による IPv6 情報の表示方法

ypmatch コマンドを実行して NIS で IPv6 情報を表示するには、次のように操作します。

- コマンド行で次のコマンドを入力します。

  ```
  % ypmatch host-name ipnodes.byname
  ```

  host-name に関する情報が表示されます。

例 **15–1** 例 — ypmatch を使用して NIS で IPv6 情報を表示する

```
% ypmatch vallejo ipnodes.byname
fec0::9256:a00:20ff:fe12:528    vallejo
2::9256:a00:20ff:fe12:528       vallejo
```

▼ NIS+ による IPv6 情報の表示方法

nismatch コマンドを実行して NIS で IPv6 情報を表示するには、次のように操作します。

- コマンド行で次のコマンドを入力します。

  % nismatch *host-name* **ipnodes.***org-dir*

  host-name に関する情報が表示されます。

例 **15–2** 例 – nismatch を使用して NIS+ で IPv6 情報を表示する

```
% nismatch vallejo ipnodes.org_dir
vallejo vallejo fec0::9256:a00:20ff:fe12:528
vallejo vallejo 2::9256:a00:20ff:fe12:528
```

▼ ネームサービスに依存しない IPv6 情報の表示方法

- コマンド行で次のコマンドを入力します。

  % getent ipnodes *host-name*

  host-name に関する情報が表示されます。

例 **15–3** 例 – getent を使用したネームサービスに依存しない IPv6 情報の表示

```
% getent ipnodes vallejo
2::56:a00:fe87:9aba         vallejo vallejo
fec0::56:a00:fe87:9aba      vallejo vallejo
```

第 16 章

IPv6 のファイルおよびコマンド (リファレンス)

Solaris の IPv6 の実装は、主にカーネルレベルとユーザーレベルの両方の TCP/IP スタックへの変更から構成されます。新しい IPv6 モジュールにより、トンネル、ルーター検索、ステートレスアドレス自動設定を使用できます。この章では、IPv6 の Solaris 実装に伴う概念について説明します。

この章では、以下の内容について説明します。

Solaris IPv6 実装の概要

IPv4 から IPv6 への移行の一部として、IPv6 では IPv6 パケットを IPv4 パケット内にカプセル化する方式が指定されます。また、IPv6 では、IPv6 パケット内にカプセル化された IPv6 パケットも指定します。その結果、パケットのカプセル化を行う新しいモジュール `tun(7M)` が追加されました。このモジュールはトンネルモジュールと呼び、物理的インタフェースと同様に `ifconfig` ユーティリティで plumb され、設定されます。このモジュールによってトンネルモジュールが IP デバイスと IP モジュール間に配置されます。トンネルデバイスにもシステムインタフェースリストにエントリがあります。

ifconfig(1M) ユーティリティも変更されました。このユーティリティは、IPv6 スタックを作成し、新しいパラメータをサポートします。これらについてはこの章で、あとから説明します。

ルーター検索とステートレスアドレス自動設定を行うため、in.ndpd(1M) デーモンが追加されました。

IPv6 ネットワークインタフェース構成ファイル

IPv4 では起動時に /etc/hostname.*interface* を使用しましたが、IPv6 でも起動時にファイル /etc/hostname6.*interface* を使用してネットワークインタフェースを自動的に定義します。このとき、少なくとも /etc/hostname.* ファイルまたは、/etc/hostname6.* ファイルがローカルマシンに存在している必要があります。これらのファイルは、Solaris インストールプログラムで生成されます。ファイル名の *interface* は、プライマリネットワークインタフェースのデバイス名に置き換えられます。

ファイル名の構文は、次のとおりです。

hostname.*interface*
hostname6.*interface*

interface の構文は、次のとおりです。

dev[.*Module*[.*Module* ...]]*PPA*

Dev	ネットワークインタフェースデバイス。デバイスは le、qe など物理ネットワークインタフェースか、トンネルなどの論理インタフェース。詳細については、337 ページの「IPv6 の Solaris トンネルインタフェース」を参照してください。
Module	結合される際にデバイスにプッシュされるストリームモジュールのリスト
PPA	物理的な接続ポイント

構文 [.[.]] も可能です。

有効なファイル名は、次のとおりです。

```
hostname6.le0
hostname6.ip.tun0
hostname.ip.tun0
```

IPv6 インタフェース構成ファイルのエントリ

IPv6 におけるインタフェースの自動設定では、その所属するリンク層アドレスに基づいてリンクローカルアドレスをノード側で計算できます。そのため、IPv6 インタフェース構成ファイルにはエントリがないことがあります。その場合、起動スクリプトによってインタフェースが設定されます。ノードは近傍検索デーモン `in.ndpd` で他のアドレスやプレフィックスの情報を取り出します。インタフェースに静的アドレスが必要な場合、`ifconfig` ユーティリティのコマンドインタフェースを使用します。その結果、アドレスまたはホスト名が `/etc/hostname6.`*interface* (または `/etc/hostname.`*interface*) に保存されます。インタフェースが構成されるときに、その内容が `ifconfig` に渡されます。

この場合、ファイルに含まれるエントリは 1 つだけです。このエントリは、ネットワークインタフェースに関連付けられたホスト名または IP アドレスです。たとえば、`ahaggar` というマシンの一次ネットワークインタフェースが `smc0` であるとします。その `/etc/hostname6.*` ファイル名は `/etc/hostname6.smc0` となります。そのエントリは `ahaggar` です。

ネットワーキングの起動スクリプトでは、ルーティングデーモンとパケット転送を開始するために、インタフェース数と、`/etc/inet/ndpd.conf` ファイルの有無を調べます。309 ページの「Solaris IPv6 ルーターの設定方法」を参照してください。

`ifconfig` ユーティリティに対する IPv6 拡張機能

`ifconfig` ユーティリティにより、IPv6 インタフェースとトンネルモジュールを結合できるようになりました。`ifconfig(1M)` ユーティリティでは、ioctl の拡張セットで IPv4 ネットワークインタフェースと IPv6 ネットワークインタフェースの両方を設定します。表 16–1 は、このユーティリティに追加されたオプションセットです。このユーティリティによる診断手順については、313 ページの「インタフェースアドレス割り当ての表示方法」を参照してください。

表 16–1 新しい `ifconfig` ユーティリティオプション

オプション	説明
`index`	インタフェースインデックスを設定する
`tsrc/ tdst`	トンネルソース / 宛先を設定する
`addif`	論理インタフェースの次の候補を作成する
`removeif`	指定された IP アドレスの論理インタフェースを削除する
`destination`	インタフェースにポイントツーポイント宛先アドレスを設定する
`set`	インタフェースにアドレスとネットマスクのどちらか、または両方を設定する

表 **16–1** 新しい ifconfig ユーティリティオプション　(続き)

オプション	説明
subnet	インタフェースのサブネットアドレスを設定する
xmit/ -xmit	インタフェースにおけるパケット伝送を使用可能または使用不能する

IPv6 設定手順については、307 ページの「IPv6 ノードを有効にする」を参照してください。

例 – 新しい ifconfig ユーティリティオプション

次に示す ifconfig コマンドは、まず hme0:3 論理インタフェースを 1234::5678/64 IPv6 アドレスに作成します。次に up オプションでインタフェースを使用可能にし、状態を報告し、インタフェースを使用不可にします。最後に、インタフェースを削除します。

例 **16–1** 例 – addif と removeif の使用

```
# ifconfig hme0 inet6 addif 1234::5678/64 up
Created new logical interface hme0:3

# ifconfig hme0:3 inet6
hme0:3: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6 1234::5678/64

# ifconfig hme0:3 inet6 down

# ifconfig hme0 inet6 removeif 1234::5678
```

次に示す ifconfig コマンドは、まず物理インタフェース名に関連付けられたデバイスを開きます。次に TCP/IP がデバイスを使用するために必要なストリームを構成し、デバイスの状態を報告し、トンネルのソースアドレスと宛先アドレスを構成します。最後に、構成後のデバイスの最新状態を報告します。

例 **16–2** 例 – tsrc/tdst と index

```
# ifconfig ip.tun0 inet6 plumb index 13

# ifconfig ip.tun0 inet6
ip.tun0: flags=2200850<POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu
1480 index 13
        inet tunnel src 0.0.0.0
        inet6 fe80::/10 --> ::

# ifconfig ip.tun0 inet6 tsrc 120.46.86.158 tdst 120.46.86.122

# ifconfig ip.tun0 inet6
ip.tun0: flags=2200850<POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu
```

例 **16–2** 例 – tsrc/tdst と index　*(続き)*

```
1480 index 13
        inet tunnel src 120.46.86.158  tunnel dst 120.46.86.122
        inet6 fe80::8192:569e/10 --> fe80::8192:567a
```

複数のネットワークインタフェースがあるノード

ノードに複数のネットワークインタフェースがある場合、追加インタフェース用に /etc/hostname.*interface* ファイルを作成する必要があります。

IPv4 の動作

たとえば、図 4–1 に示すマシン timbuktu について考えてみましょう。このシステムには、2 つのネットワークインタフェースがあり、ルーターとして機能します。プライマリネットワークインタフェース le0 は、ネットワーク 192.9.200 に接続されています。システムの IP アドレスは 192.9.200.70、ホスト名は timbuktu です。Solaris インストールプログラムによって、一次ネットワークインタフェースにファイル /etc/hostname.le0 が作成され、ホスト名 timbuktu がファイルに入力されます。

2 番めのネットワークインタフェースは le1 で、192.9.201 に接続されています。このインタフェースは timbuktu に物理的にインストールされていますが、独自の IP アドレスが必要です。そのため、このインタフェースに対して /etc/hostname.le1 ファイルを手動で作成する必要があります。このファイルのエントリはルーター名 timbuktu-201 です。

IPv6 の動作

IPv6 を設定する場合、/etc/hostname6.le0 と /etc/hostname6.le1 のインタフェースだけが必要です。各インタフェースアドレスは、システムの起動時に自動的に設定されます。

IPv6 デーモン

ここでは、次の IPv6 デーモンについて説明します。

- in.ndpd – IPv6 自動設定用のデーモン
- in.ripngd – IPv6 のネットワークルーティングデーモン
- inetd – インターネットサービスデーモン

in.ndpd デーモン

このデーモンでは、IPv6 用のルーター発見と自動アドレスの設定が実装されます。表 16–4 は、サポートされているオプションを示します。

表 **16–2** in.ndpd デーモンのオプション

オプション	説明
-d	すべてのイベントのデバッグをオンにする
-D	特定のイベントのデバッグをオンにする
-f	設定を読み出す元のファイル (デフォルトファイルのかわり)
-I	インタフェースごとに関連情報を印刷する
-n	ルーター通知をループバックしない
-r	受信パケットを無視する
-v	冗長モード (さまざまな種類の診断メッセージを報告する)
-t	パケット追跡をオンにする

パラメータは、in.ndpd の動作を制御します。これらのパラメータは /etc/inet/ndpd.conf 構成ファイルと /var/inet/ndpd_state*interface* 起動ファイル (存在する場合) に設定されます。

/etc/inet/ndpd.conf が存在すると構文解析され、ノードをルーターとして使用するための設定が行われます。表 16–3 に、このファイルに出現する可能性がある各種キーワードをまとめます。ホストを起動してもルーターがすぐに利用できなかったり、ルーターが通知したパケットがドロップしてホストに届かないことがあります。/var/inet/ndpd_state.*interface* ファイルは状態ファイルです。このファイルはノードごとに定期的に更新されます。ノードに障害が発生し再起動した場合、ルーターがなくてもノードはインタフェースを設定できます。このファイルにはインタフェースアドレス、更新時間、有効期間などの情報が保存されています。また、先のルーター通知で得られた情報も保存されています。

注 – 状態ファイルの内容を変更する必要はありません。このファイルは、in.ndpd デーモンが自動的に管理します。

表 16–3 /etc/inet/ndpd.conf キーワード

キーワード	説明
ifdefault	すべてのインタフェースのルーターの動作を指定する。次の構文を使用してルーターパラメータと対応する値を設定する ifdefault [*variable value*]
prefixdefault	プレフィックス通知のデフォルトの動作を指定する。次の構文を使用してルーターパラメータと対応する値を設定する prefixdefault [*variable value*]
if	インタフェース別パラメータを設定する。構文は次のとおり if *interface* [*variable value*]
prefix	インタフェース別プレフィックス情報を通知する。構文は次のとおり prefix *prefix /length interface* [*variable value*]

注 – ifdefault/prefixdefault エントリは、構成ファイルの if エントリと prefix エントリの前に置く必要があります。

設定変数と設定できる値については、in.ndpd(1M) と ndpd.conf(4) のマニュアルページを参照してください。

例 – /etc/inet/ndpd.conf ファイル

次の例は、コメント行のテンプレートと、キーワードと設定変数の使用方法を示します。

```
# ifdefault       [variable value]*
# prefixdefault [variable value]*
# if ifname     [variable value]*
# prefix prefix/length ifname
#
#  Per interface configuration variables
#
#DupAddrDetectTransmits
#AdvSendAdvertisements
#MaxRtrAdvInterval
#MinRtrAdvInterval
```

```
#AdvManagedFlag
#AdvOtherConfigFlag
#AdvLinkMTU
#AdvReachableTime
#AdvRetransTimer
#AdvCurHopLimit
#AdvDefaultLifetime
#
# Per Prefix:  AdvPrefixList configuration variables
#
#
#AdvValidLifetime
#AdvOnLinkFlag
#AdvPreferredLifetime
#AdvAutonomousFlag
#AdvValidExpiration
#AdvPreferredExpiration

ifdefault AdvReachableTime 30000 AdvRetransTimer 2000
prefixdefault AdvValidLifetime 240m AdvPreferredLifetime 120m

if qe0 AdvSendAdvertisements 1
prefix 2:0:0:56::/64 qe0
prefix fec0:0:0:56::/64 qe0

if qe1 AdvSendAdvertisements 1
prefix 2:0:0:55::/64 qe1
prefix fec0:0:0:56::/64 qe1

if qe2 AdvSendAdvertisements 1
prefix 2:0:0:54::/64 qe2
prefix fec0:0:0:54::/64 qe2
```

in.ripngd デーモン

in.ripngd デーモンは、IPv6 ルーターの RIP 次世代ルーティングプロトコルを実装します。このプロトコルは、IPv6 用の RIP に相当する内容を定義します。RIP は、広く使用されている IPv4 ルーティングプロトコルで、Bellman-Ford 距離ベクトルアルゴリズムに基づいています。表 16–4 は、サポートされているオプションを示します。

表 16–4 in.ripngd デーモンのオプション

オプション	説明
-p *n*	*n* は RIPNG パケットの送受信に使用する代替ポート番号を指定する
-q	ルーティング情報を打ち切る

表 **16–4** in.ripngd デーモンのオプション　(続き)

-s	デーモンがルーターとして動作しているかどうかのルーティング情報の提供を強制する
-P	ポイズンリバースを打ち切る
-S	in.ripngd がルーターとして機能しない場合、各ルーターにはデフォルトのルートだけが指定される

inetd インターネットサービスデーモン

IPv6 有効化サーバーは、IPv4 アドレスか IPv6 アドレスを処理できるサーバーです。IPv6 有効化サーバーは、対応するクライアントで使用しているプロトコルと同じプロトコルを使用します。/etc/inet/inetd.conf ファイルには、inetd(1M) がソケット経由でインターネット要求を受信したときに呼び出すサーバーリストが保存されています。ソケットベースのインターネットサーバーエントリはそれぞれ、次の構文を使用する 1 行です。

service_name socket_type proto flags user server_pathname args

各フィールドに指定できる値については、inetd.conf(4) のマニュアルページを参照してください。Solaris オペレーティング環境の場合、IPv6 有効化としてサービスを /etc/inet/inetd.conf ファイルに指定するには、*proto* フィールドに tcp6 または udp6 を指定します。サービスが IPv4 専用の場合、*proto* フィールドは tcp または udp として指定します。サービスの *proto* 値に tcp6 または udp6 を指定すると、inetd は所定のデーモンに AF_INET6 ソケットを渡します。

inetd.conf ファイルの次のエントリは、IPv4 クライアントアプリケーションと IPv6 クライアントアプリケーションの両方と通信できる udp サーバー (myserver) を表します。

例 **16–3** IPv4 クライアントアプリケーションと IPv6 クライアントアプリケーションの両方と通信するサーバー

```
myserver   dgram   udp6    wait    root    /usr/sbin/myserver    myserver
```

IPv6 有効化サーバーは、AF_INET (IPv4 専用) ソケットまたは AF_INET6 (IPv6 と IPv4) ソケットを inetd から継承できます。サービスの *proto* 値は tcp6 (udp6) または tcp (udp) として指定されます。この種のサーバーでは、2 つの inetd.conf エントリを指定できます。1 つは *proto* を tcp として、もう1 つは *proto* を tcp6 として指定できます。

注 – AF_INET6 ソケットは、IPv4 プロトコルと IPv6 プロトコルのどちらでも使用できるため、*proto* 値 tcp6 (udp6) を指定すれば充分です。

各種 IPv6 有効化サーバーの記述方法については、『プログラミングインタフェース』を参照してください。

Solaris ソフトウェアとともに提供されるサーバーはすべて、*proto* 値を tcp6 または udp6 と指定する inetd エントリが1 つあれば十分です。ただし、リモートシェルサーバー (shell) とリモート実行サーバー (exec) のエントリには、tcp と tcp6 の両方の *proto* 値を指定する必要があります。例 16–4 は、rlogin、telnet、shell、exec 用の inetd エントリです。

例 **16–4** Solaris ソフトウェアで提供されるサーバー用の inetd.conf エントリ

```
login stream    tcp6 nowait root  /usr/sbin/in.rlogind  in.rlogind
telnet stream    tcp6 nowait root /usr/sbin/in.telnetd in.telnetd
shell    stream    tcp    nowait    root    /usr/sbin/in.rshd    in.rshd
shell    stream    tcp6    nowait    root    /usr/sbin/in.rshd    in.rshd
exec    stream    tcp    nowait    root    /usr/sbin/in.rexecd    in.rexecd
exec    stream    tcp6    nowait    root    /usr/sbin/in.rexecd    in.rexecd
```

TCP ラッパーは、telnet などさまざまなネットワークサービスで入力要求を監視、フィルタ処理するためのパブリックドメインユーティリティです。以上のユーティリティの *server_pathname* として TCP ラッパーを指定する場合、*TCP* ラッパーが IPv6 対応である必要があります。対応していない場合、TCP ラッパーで使用するサービスの *proto* を tcp か udp に指定する必要があります。

また、Solaris ユーティリティを別の実装と入れ替える場合、そのサービスの実装が IPv6 をサポートしていることを確認する必要があります。サポートしていない場合、その実装の *proto* を tcp か udp に指定する必要があります。

注 – *proto* 値を tcp か udp のどちらか一方に指定すると、サービスでは IPv4 だけが使用されます。IPv4 接続と IPv6 接続の両方を有効にするには、proto 値を tcp6 か udp6 に指定する必要があります。サービスで IPv6 をサポートしていない場合、tcp6 や udp6 は指定しないでください。

ソケットを使用する IPv6 有効化サーバーについては、『プログラミングインタフェース』のソケット API への IPv6 拡張機能についての説明を参照してください。

既存のユーティリティに対する IPv6 拡張機能

ユーザーレベルインタフェースでは、次のユーティリティの組み込み拡張機能も変更されました。

- netstat(1M)
- snoop(1M)
- route(1M)

- ping(1M)
- traceroute(1M)

ifconfig(1M) ユーティリティも変更されました。詳細については、327 ページの「ifconfig ユーティリティに対する IPv6 拡張機能」を参照してください。

netstat(1M)

IPv4 ネットワーク状態の表示の他、netstat では IPv6 ネットワーク状態も表示できます。/etc/default/inet_type ファイルと -f コマンド行オプションで DEFAULT_IP 値を設定して、表示するプロトコル情報を選択できます。DEFAULT_IP のパラメータ設定では、netstat に IPv4 情報だけが表示されていることを確認できます。この設定は、-f オプションで無効にできます。inet_type ファイルの詳細については、inet_type(4) のマニュアルページを参照してください。

新しい -p オプションでは、net-to-media テーブルが表示されます。これは、IPv4 用の ARP テーブルであり、IPv6 用の近傍キャッシュです。詳細については、netstat(1M) のマニュアルページを参照してください。このコマンドの使用方法については、314 ページの「ネットワーク状態の表示方法」を参照してください。

snoop(1M)

snoop コマンドは、IPv4 パケットと IPv6 パケットの両方を取り込んで、IPv6 ヘッダー、IPv6 拡張ヘッダー、ICMPv6 ヘッダー、近傍検索プロトコルデータを表示できます。 デフォルトで、snoop コマンドは、IPv4 パケットと IPv6 パケットの両方を表示します。ip プロトコルキーワードか ip6 プロトコルキーワードを指定すると、snoop コマンドは IPv4 パケットか IPv6 パケットのどちらかだけを表示します。IPv6 フィルタオプションでは、すべてのパケットをフィルタの対象にでき (IPv4 と IPv6 の両方)、IPv6 パケットだけが表示されます。詳細については、snoop(1M) のマニュアルページを参照してください。このコマンドの使用方法については、318 ページの「IPv6 ネットワークトラフィックの監視方法」を参照してください。

route(1M)

このユーティリティは、IPv4 ルーターと IPv6 ルーターの両方で実行できます。デフォルトで、route は IPv4 ルートで実行します。コマンド行で route コマンドの直後にオプション -inet6 を指定すると、操作が IPv6 ルートで実行されます。詳細については、route(1M) のマニュアルページを参照してください。

ping(1M)

ping コマンドは、IPv4 プロトコルと IPv6 プロトコルの両方で、宛先ホストを調べることができます。プロトコル選択は、指定の宛先ホストのネームサーバーが戻すアドレスに依存します。デフォルトでネームサーバーが、宛先ホストの IPv6 アドレスを

戻すと、ping コマンドは IPv6 プロトコルを使用します。サーバーが IPv4 アドレスだけを戻すと、IPv4 プロトコルを使用します。-A コマンド行オプションで使用するプロトコルを指定すれば、この動作を無効にできます。

その他、-a コマンド行オプションを指定すれば、マルチホーム宛先ホストのアドレスをすべて ping できます。詳細については、ping(1M) のマニュアルページを参照してください。このコマンドの使用方法については、319 ページの「すべてのマルチホームホストアドレスの探査方法」を参照してください。

traceroute(1M)

traceroute コマンドを使用して、指定ホストまでの IPv4 ルートと IPv6 ルートの両方をトレースできます。使用するプロトコルの選択について、traceroute では、ping と同じアルゴリズムを使用します。選択を無効にするには、-A コマンド行オプションを使用します。マルチホームホストのすべてのアドレスまでの各ルートは -a コマンド行オプションでトレースできます。traceroute(1M) のマニュアルページを参照してください。

表示出力の制御

netstat コマンドと ifconfig コマンドによる出力表示の方法を制御できます。

- コマンド行に追加したキーワードで、inet アドレスまたは inet6 アドレスを指定する
- /etc/default/inet_type ファイルの設定変数 DEFAULT_IP を設定する

DEFAULT_IP の値は、IP_VERSION4、IP_VERSION6、BOTH のどれかに設定できます。DEFAULT_IP を指定してこのファイルを作成しない場合、netstat と ifconfig では、両方のバージョンが表示されます。

注 – inet キーワードオプションと inet6 キーワードオプションは、netstat コマンドと ifconfig コマンドの使用時に inet_type ファイルで設定した値を無効にします。

操作については、317 ページの「IPv6 関連コマンドの出力表示の制御方法」を参照してください。

IPv6 の Solaris トンネルインタフェース

トンネルインタフェースのフォーマットは次のとおりです。

`ip.tun` *ppa*

ppa は物理的な接続ポイントです。

注 – Solaris ソフトウェアでは、IPv6 パケット内にパケットをカプセル化できません。

システム起動時に、トンネルモジュール (`tun`) は、(`ifconfig` によって) IP の最上位にプッシュされ、仮想インタフェースが作成されます。このプッシュは、`hostname6.*` ファイルを作成することによって行われます。

たとえば、IPv4 ネットワーク経由で IPv6 パケットをカプセル化するためのトンネルを作成するには、次のファイルを作成します。

`/etc/hostname6.ip.tun0`

このファイルの内容は、インタフェースが結合された後に `ifconfig`(1M) に渡されます。ポイントツーポイントトンネルの設定に必要なパラメータになります。

次のリストは、`hostname6.ip.tun0` ファイルのエントリの例です。

例 **16–5** `hostname6.`*interface* エントリ

```
tsrc 120.68.100.23 tdst 120.68.7.19 up
addif 1234:1234::1 5678:5678::2 up
```

この例の IPv4 ソースと宛先アドレスは、`ip.tun0` インタフェースのソース IPv6 リンクローカルアドレスと宛先 IPv6 リンクローカルアドレスの自動設定に必要なトークンとして機能します。`ip.tun0` インタフェースと、`addif` コマンドによってソース IPv6 アドレスと宛先 IPv6 アドレスが指定された論理インタフェース (`ip.tun0:1`) の、2 つのインタフェースが設定されます。

すでに述べたとおり、システムをマルチユーザーとして起動すると、これらの設定ファイルの内容が変更されずに `ifconfig` に渡されます。上の例は次の内容と同じです。

```
# ifconfig ip.tun0 inet6 plumb
# ifconfig ip.tun0 inet6 tsrc 120.68.100.23 tdst 120.68.7.19 up
# ifconfig ip.tun0 inet6 addif 1234:1234::1 5678:5678::2 up
```

このトンネルにおける `ifconfig -a` の出力は次のとおりです。

```
ip.tun0: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
index 6
```

```
        inet tunnel src 120.68.100.23  tunnel dst 120.68.7.19
        inet6 fe80::c0a8:6417/10 --> fe80::c0a8:713
ip.tun0:1: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
index 5
        inet6 1234:1234::1/128 --> 5678:5678::2
```

次の構文で設定ファイルに行を追加すれば、さらに論理インタフェースを設定できます。

```
addif IPv6-source IPv6-destination up
```

注 – トンネルのどちらかの端は、トンネル経由で 1 つまたは複数のプレフィックスを通知する IPv6 ルーターです。トンネル構成ファイルには `addif` コマンドは必要ありません。他のアドレスは自動設定されるため、必要とされる可能性があるのは `tsrc` と `tdst` だけです。

場合によっては、特定のトンネルについて、固有のソースリンクローカルアドレスと宛先リンクローカルアドレスを手動で設定する必要があることもあります。その場合、設定ファイルの最初の行を変更して、これらのリンクローカルアドレスを組み込みます。次に例を示します。

```
tsrc 120.68.100.23 tdst 120.68.7.19 fe80::1/10 fe80::2 up
```

ソースリンクローカルアドレスには、長さが 10 のプレフィックスがあります。この例では、`ip.tun0` インタフェースは次のようになります。

```
ip.tun0: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
index 6
        inet tunnel src 120.68.100.23  tunnel dst 120.68.7.19
        inet6 fe80::1/10 --> fe80::2
```

`tun` の固有の情報については、`tun(7M)` のマニュアルページを参照してください。IPv6 への移行時のトンネルの概念の一般的な説明については、346 ページの「トンネル機構」を参照してください。トンネルの設定方法については、320 ページの「IPv4 トンネルによる IPv6 の設定方法」を参照してください。

Solaris ネームサービスに対する IPv6 拡張機能

ここでは、Solaris 8 リリースで IPv6 の実装により導入されたネーミングの変更について説明します。 IPv6 アドレスは Solaris ネームサービス (NIS、NIS+、DNS およびファイル) のどれでも保存できます。また、IPv6 RPC トランスポートで NIS と NIS+ を使用して NIS データまたは NIS+ データを検出することもできます。

/etc/inet/ipnodes ファイル

/etc/inet/ipnodes ファイルには、IPv4 と IPv6 のアドレスが格納されています。このファイルはローカルデータベースとして、ホスト名を IPv4 アドレスや IPv6 アドレスに関連付けます。ホスト名やそのアドレスは、/etc/inet/ipnodes などの静的ファイルには保存しないでください。ただし、テスト目的の場合は IPv4 アドレスを /etc/inet/hosts に保存するのと同じ方法で IPv6 アドレスを保存します。ipnodes ファイルでは、hosts ファイルと同じフォーマット変換を使用します。hosts ファイルについては、50 ページの「ネットワークデータベース」を参照してください。ipnodes ファイルについては、ipnodes(4) のマニュアルページを参照してください。

IPv6-aware (IPv6 が利用可能な) ユーティリティでは、新しい /etc/inet/ipnodes データベースを使用します。既存の /etc/hosts データベースには、IPv4 アドレスだけを保存していますが、既存のアプリケーションの便宜上、このデータベースは変更されません。ipnodes データベースがない場合、IPv6-aware ユーティリティでは既存の hosts データベースを使用します。

注 – アドレスを追加する必要がある場合、IPv4 アドレスは hosts ファイルと ipnodes ファイルの両方に追加しなければなりません。IPv6 アドレスは ipnodes ファイルにだけ追加します。

例 – /etc/inet/ipnodes ファイル

```
#
# Internet IPv6 host table
# with both IPv4 and IPv6 addresses
#
::1     localhost
2::9255:a00:20ff:fe78:f37c   fripp.guitars.com fripp fripp-v6
fe80::a00:20ff:fe78:f37c     fripp-11.guitars.com frippll
120.46.85.87                    fripp.guitars.com fripp fripp-v4
2::9255:a00:20ff:fe87:9aba   strat.guitars.com strat strat-v6
fe80::a00:20ff:fe87:9aba     strat-11.guitars.com stratll
120.46.85.177                   strat.guitars.com strat strat-v4 loghost
```

注 – 上記の例のように、ホスト名アドレスは、ホスト名でグループにまとめる必要があります。

IPv6 の NIS 拡張機能

NIS 用に 2 つの新しいマップが追加されました。 `ipnodes.byname` と `ipnodes.byaddr` です。`/etc/inet/ipnodes` と同様に、これらのマップには、IPv4 情報と IPv6 情報の両方が保存されます。既存の `hosts.byname` と `hosts.byaddr` マップは、IPv4 情報だけを保存しています。既存のアプリケーションの便宜上変更されていません。

IPv6 の NIS+ 拡張機能

NIS+ 用に `ipnodes.org_dir` という新しいテーブルが追加されました。このテーブルには、ホスト用の IPv4 アドレスと IPv6 アドレスの両方が保存されています。既存の `hosts.org_dir` テーブルは IPv4 アドレス情報だけを保存しています。このテーブルは、既存のアプリケーションの便宜上変更されていません。

IPv6 の DNS 拡張機能

AAAA レコードとして定義された新しいリソースレコードが、RFC 1886 で定義されています。このAAAA レコードは、ホスト名を 128 ビット IPv6 アドレスにマップします。PTRレコードは IPv6 でも、IP アドレスをホスト名にマップするときに使用されています。128 ビットアドレスの 32 の 4 ビットニブルは、IPv6 アドレス用に反転されています。各ニブルは対応する 16 進 ASCII 値に変換されます。変換後、`ip6.int` が追加されます。

`nsswitch.conf` ファイルへの変更

`/etc/inet/ipnodes` で IPv6 アドレスを調べる機能に加え、IPv6 サポートは、NIS ネームサービス、NIS+ ネームサービス、DNS ネームサービスに追加されています。その結果、`nssiwtch.conf(4)` ファイルは IPv6 検索をサポートするように変更されました。ipnodes 行が `/etc/nsswitch.conf` ファイルに追加されました。この追加により、Solaris ネームサービス (NIS、NIS+、DNS、ファイル) の新しいデータベースで検索が可能になりました。次の太字で示された行は、ipnodes エントリの例です。

```
hosts:  files dns nisplus [NOTFOUND=return]
ipnodes: files dns nisplus [NOTFOUND=return]
```

注 – IPv4 アドレスと IPv6 アドレスでこれらの `ipnodes` データベースを生成してから、複数のネームサービスで `ipnodes` を探すように `/etc/nsswitch.conf` ファイルを変更してください。ホストアドレスの解決時に不要な遅延が発生してしまうからです (起動タイミングの遅れが発生することもあります)。

図 16–1 は、gethostbyname() コマンドと getipnodebyname() コマンドを使用するアプリケーションにおける、nsswitch.conf ファイルと新しいネームサービスデータベースの新しい関係を示します。斜体の項目は新規です。gethostbyname() コマンドは、/etc/inet/hosts に保存されている IPv4 アドレスだけを調べます。getipnodebyname() コマンドは、nsswitch.conf ファイルの ipnodes エントリで指定したデータベースを調べます。検索に失敗すると、nsswitch.conf ファイルの hosts エントリで指定したデータベースを調べます。

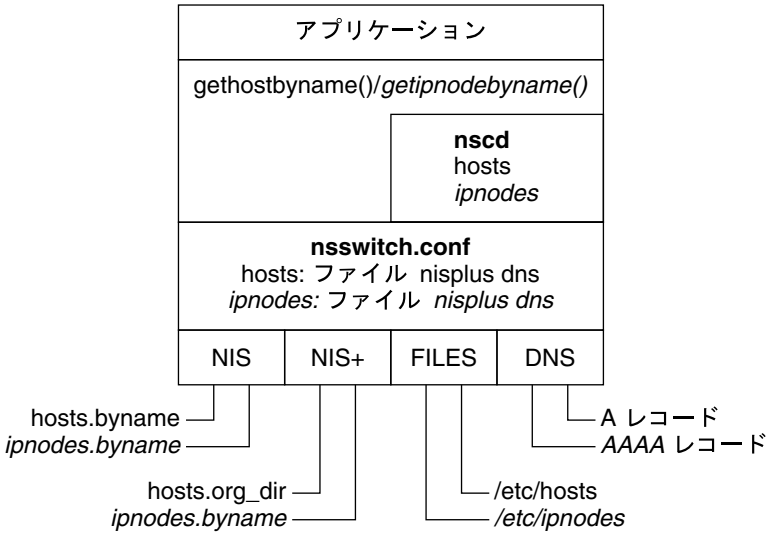


図 **16–1** nsswitch.conf とネームサービスの関係

ネームサービスの詳細については、『*Solaris* のシステム管理 *(*ネーミングとディレクトリサービス *: DNS*、*NIS*、*LDAP* 編*)*』を参照してください。

ネームサービスコマンドの変更

IPv6 をサポートできるように、既存のネームサービスコマンドで IPv6 アドレスを調べることができます。たとえば、ypmatch コマンドは、新しい NIS マップに使用できます。nismatch コマンドは、新しい NIS+ テーブルに使用できます。nslookup コマンドでは、DNS の新しい AAAA レコードを調べることができます。ネームサービスの変更については、340 ページの「IPv6 の NIS 拡張機能」、340 ページの「IPv6 の NIS+ 拡張機能」、および 340 ページの「IPv6 の DNS 拡張機能」を参照してください。

これらのコマンドの使用手順については、321 ページの「IPv6 ネームサービス情報の表示」を参照してください。

NFS と RPC による IPv6 のサポート

NFS と RPC ソフトウェアは、シームレスに IPv6 をサポートします。NFS サービスに関連のある既存のコマンドは変更されていません。ほとんどの RPC アプリケーションが、変更なしで IPv6 で実行できます。トランスポート機能のある一部の高度 RPC アプリケーションに更新が必要な場合があります。

IPv6–Over-ATM サポート

Solaris オペレーティング環境では、IPv6–over-ATM、固定仮想回路 (PVC)、静的な交換仮想回路 (SVC) をサポートするようになりました。

第 17 章

IPv4 から IPv6 への移行 (リファレンス)

IPv6 をサポートするためにホストとルーターをアップグレードした後も、IPv4 だけをサポートしているホストとルーター とのネットワーク経由の相互運用が必要です。この章では、IPv4 から IPv6 への移行と、標準的な解決法の概要について説明します。RFC 1933 でも、移行問題の詳しい解決法を示しています。

この章では、以下の内容について説明します。

移行条件

移行時のグローバルな調整は不要です。サイトとインターネットサービスプロバイダ (ISP) はそれぞれのスケジュールで移行できます。また、移行時の依存条件も最小限に抑えました。たとえば、ホストのアップグレード前にルーターを IPv6 にアップグレードしなくても移行できます。

サイトが異なれば、移行時にはそれぞれの制約が課されます。また、IPv6 の初期アダプタには、IPv6 の製品版ユーザーの場合とは異なる問題があります。RFC 1933 は現在利用できる移行ツールを定義しています。移行の必然性としては、IPv4 アドレス領域の不足または IPv6 の新機能を使用する必要性のどちらか、または両方が考えられます。IPv6 仕様では、移行時には既存のプロトコルとアプリケーションとの完全な互換性が求められます。

移行方式を理解できるように、次の用語を定義します。

- IPv4 専用ノード – IPv4 だけを実装したホストやルーター。IPv4 専用ノードでは IPv6 は認識できない。移行以前に既存の IPv4 ホストとルーターのインストール可能ベースは IPv4 専用ノード
- IPv6/IPv4 ノード – IPv4 と IPv6 の両方を実装するホストとルーター。デュアルスタックとも呼ぶ
- IPv6 専用ノード – IPv6 を実装するホストまたはルーター。IPv4 を実装しない
- IPv6 ノード – IPv6 を実装するホストまたはルーター。IPv6/IPv4 ノードと IPv6 専用ノードは、どちらも IPv6 ノード
- IPv4 ノード – IPv4 を実装するホストまたはルーター。IPv6/IPv4 ノードと IPv4 専用ノードは、どちらも IPv4 ノード
- サイト – インターネットのプライベートトポロジの 1 つ。すなわちあらゆるユーザーを対象としたトラフィック伝送を行わないトポロジ。サイトが物理的に広範囲に展開されることがある。たとえば、多国籍企業のプライベートネットワークは、1 つのサイト

標準移行ツール

RFC 1933 は、次の移行方式を定義しています。

- ホストとルーターを IPv6 にアップグレードするとき、それらの IPv4 の機能を残す。したがって、すべての IPv4 プロトコルおよびアプリケーションとの互換性が確保される。このようなホストおよびルーターをデュアルスタックと呼ぶ
- IPv6 対応ノードに関する情報は、(DNS などの) ネームサービスを利用して伝送する
- IPv6 アドレス形式には、IPv4 アドレスを保存する
- IPv4 パケットで IPv6 パケットをトンネル処理して、IPv6 にアップグレードされていないルーターを通過できる

デュアルスタックの実装

デュアルスタックとは、アプリケーションからネットワーク層に至るプロトコルスタックのすべてのレベルの完全な複製をいいます。デュアルスタックの例として、同じマシンで実行する OSI プロトコルと TCP/IP プロトコルがあります。ただし、 IPv6 移行の観点からは、プロトコルスタックに IPv4 と IPv6 の両方を組み込むことを表します。残りスタックは同一となります。この場合、同じ伝送プロトコル (TCP、UDPなど) が IPv4 と IPv6 の両方で実行します。また、同じアプリケーションも IPv4 と IPv6 の両方で実行します。

次の図は、OSI 層全体にわたるデュアルスタックプロトコルを表します。

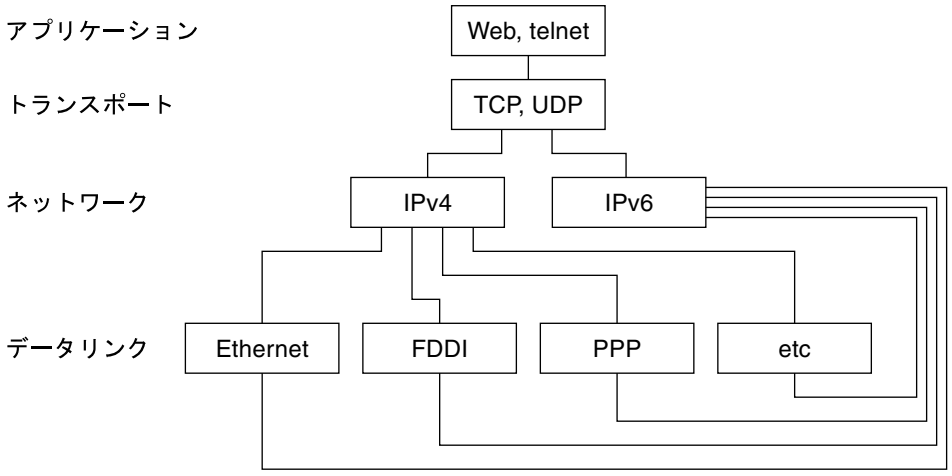


図 **17–1** デュアルスタックプロトコル

デュアルスタック方式では、ホストとルーター両方のサブセットをアップグレードして、IPv4 に加えて IPv6 をサポートします。この方法では、アップグレードされた後のノードからも IPv4 で常に IPv4 専用ノードと相互運用できます。

ネームサービスの設定

デュアルノードでは、ピアが IPv6 と IPv4 のどちらをサポートしているか明確でないと、伝送時にどちらの IP バージョンを使用するのかが決まりません。そこで、ネームサービスでどんな情報を伝達するかを制御すると、デュアルノードで使用する IP バージョンを決定できます。さらに、ネームサービスで IPv4 ノードの IP アドレスと IPv6 ノードの IP アドレスを定義します。それによって、デュアルノードでは、両方のアドレスをネームサービスで使用できます。

IPv6 アドレスをネームサービスに指定した場合も、IPv6 でノードにアクセスできます。ただし、ノードにアクセスできるのは、ネームサービスから情報を得たノードだけです。たとえば、NIS に IPv6 アドレスを指定すると、その IPv6 ホストは IPv6 からアクセスできます。ただし、IPv6 ホストにアクセスできるのは、NIS ドメインに所属する IPv6 とデュアルノードだけです。 グローバル DNS に IPv6 アドレスを指定するには、そのノードがインターネット IPv6 バックボーンからアクセスできることが条件です。これは、IPv4 の場合も同様です。たとえば、メール配信の操作は、IPv4 でアクセスできるノードの IPv4 アドレスがあるかどうかに依存します。 これは、HTTP プロキシの操作の場合も同様です。 たとえば、ファイアウォールなどの理由で IPv4 でアクセスできない場合、ネームサービスは内部ファイアウォールと 外部ファイアウォールのデータベースに分けます。 これにより、IPv4 アドレスがアクセスできる範囲だけで認識できるようになります。

ネームサービスのアクセスに使用するプロトコルは、ネームサービスで検索できるアドレスタイプに依存しません。このネームサービスサポートでは、デュアルスタックとの組み合わせにより、デュアルノードから、IPv4 専用ノードとの通信に IPv4 を使用できます。 また、IPv6 ノードとの通信には IPv6 を使用できます。 ただし、宛先までの IPv6 ルートが必要です。

IPv4 互換アドレスフォーマットの使用

通常 32 ビット IPv4 アドレスは、128 ビット IPv6 アドレスで表現できます。移行機能では、次の 2 つの形式を定義しています。

- IPv4 互換アドレス

000 ... 000	IPv4 アドレス

- IPv4 マップアドレス

000 ... 000	0xffff	IPv4 アドレス

IPv6 ノードは互換フォーマットで表現します。このフォーマットでは、実際の IPv6 アドレスがなくても IPv6 ノードを使用できます。また、IPv4 専用ルーターで自動トンネルを使用できるため、このアドレスフォーマットではさまざまな IPv6 設定の試用が可能です。ただし、IPv6 ステートレスアドレス自動設定機構では、このアドレスは設定できません。IPv6 ステートレスアドレス自動設定機構には、DHCPv4 など既存の IPv4 機構や静的設定ファイルが必要なためです。

マップアドレスフォーマットでは、IPv4 ノードを表現します。現在ソケット API の一部でだけ、このアドレスフォーマットの使用方法が定義されています。アプリケーションでは、IPv6 アドレスと IPv4 アドレスの両方に共通のアドレスフォーマットを使用できます。 共通のアドレスフォーマットは、IPv4 アド レスを 128 ビットマップアドレスで表現します。ただし、IPv4 プロトコルトランスレータと IPv6 プロトコルトランスレータがないと、これらのアドレスは使用できません。

トンネル機構

移行時の依存状態を最小限に抑える目的から、2 つの IPv6 ノード間にあるすべてのルーターで IPv6 をサポートする必要がありません。この機構をトンネルといいます。基本的に IPv6 パケットは IPv4 パケット内部に組み込まれ、IPv4 ルーター間を転送されます。図 17–2 は、IPv4 を使用したルーター (R) 間のトンネル機構を示します。

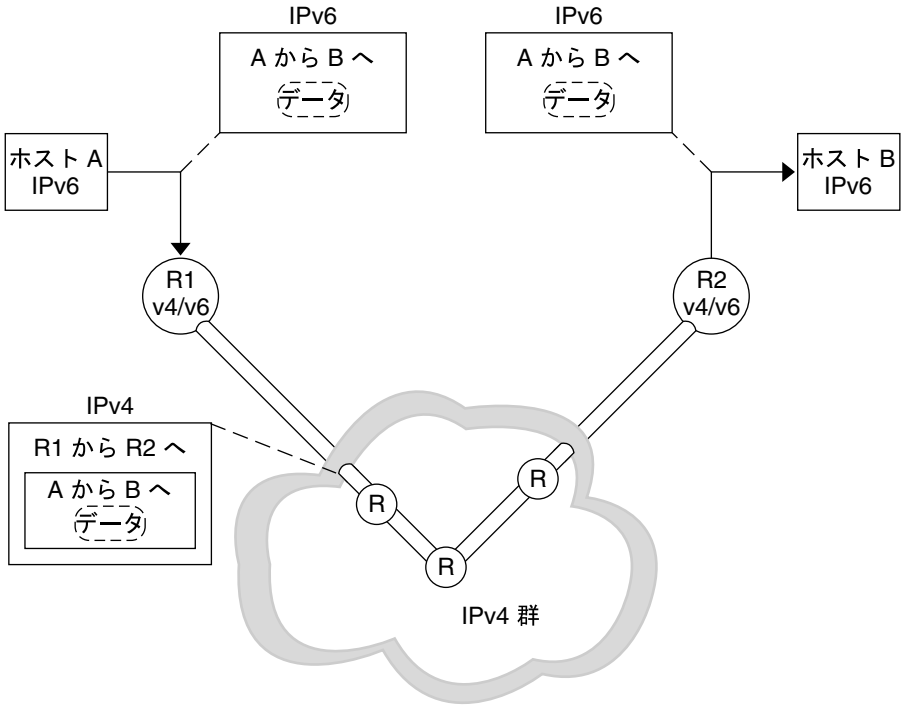


図 **17–2** トンネル機構

その他、移行時には次のようなトンネル機構の使用方法があります。

- 2 つのルーター間で設定したトンネル (上記の図を参照)
- デュアルホストで終了する自動トンネル

設定トンネルは、MBONE (IPv4 マルチキャストバックボーン) など現在はインターネットで他の目的に使用します。設定トンネルの作成手順からいうと、2 つのルーターを設定して、その間に IPv4 ネットワーク経由の仮想ポイントツーポイントリンクを作成します。 近い将来インターネットのさまざまな局面にこの種のトンネルが利用されるでしょう。

自動トンネル

初期の実験的配置では、自動トンネルの使用可能範囲は限定されています。IPv6 ルーターがない場合、自動トンネルには IPv4 互換アドレスが必要であり、IPv6 ノードと接続できることが条件です。 自動トンネルネットワークインタフェースを設定すれば、トンネルの発信元はデュアルホストとデュアルルーターのどちらが発信元でも使用できます。終点は必ずデュアルホストになります。トンネルのはたらきにより、宛先 IPv4 アドレス (トンネルの終点) が IPv4 互換宛先アドレスから抽出されて動的に指定されます。

アプリケーションとの対話

IPv6 にアップグレードしたノードでも、IPv6 を使用できるかどうかはアプリケーション次第です。アプリケーションで、IPv6 アドレスのネームサービスを要求するネットワーキング API を使用しない場合があります。また、アプリケーション側で変更が必要な API (ソケットなど) を使用する場合もあります。さらに、API のプロバイダ (`java.net` クラスなどの実装) が IPv6 アドレスをサポートしていない場合もあります。 どの場合も、ノードが送受信するのは IPv4 ノードのように IPv4 パケットだけです。

次の用語は、インターネットの世界では標準用語として使用されています。

- **IPv6-unaware (**非認識**)** – IPv6 アドレスを処理できないアプリケーション。IPv4 アドレスのないノードとは通信できない
- **IPv6-aware (**認識**)** – IPv4 アドレスがないノードとも通信できるアプリケーション。長い IPv6 アドレスも処理できる。アプリケーションに透過な場合がある。たとえば実際のアドレスの内容や形式が API によって非表示になる場合など
- **IPv6-enabled (**有効化**)** – IPv6-aware であるだけでなく、フローラベルなど IPv6 固有の機能が利用できる。有効化アプリケーションは低下モードで IPv4 も処理できる
- IPv6-required (必須) – IPv6 固有機能が必要なアプリケーション。IPv4 は処理できない

IPv4 と IPv6 の相互運用性

IPv4 から IPv6 に段階的に移行する場合、新しく導入する IPv6 有効化アプリケーションと共に既存の IPv4 アプリケーションも使用しなければなりません。最初の段階では、デュアルスタックで実行する、ということは IPv4 プロトコルスタックと IPv6 プロトコルスタックの両方で機能するホストプラットフォームとルータープラットフォームがベンダーから提供されます。IPv4 アプリケーションは、少なくとも 1 つの IPv6 インタフェースで IPv6 有効化になっているデュアルスタックでも実行できます。アプリケーションの変更 (や移植) は不要です。

デュアルスタックで実行する IPv6 アプリケーションも、IPv4 プロトコルを使用できます。IPv6アプリケーションは、IPv4 マップIPv6 アドレスを使用します。 IPv6 は設計上、(IPv4 と IPv6 で) 別々のアプリケーションは不要です。たとえば、デュアルホストの IPv4 クライアントがなくても IPv4 専用ホストのサーバーと「通信」できます。また独立した IPv6 クライアントがなくても IPv6 サーバーと通信できます。 実装時には IPv4 クライアントアプリケーションを新しい IPv6 API に移植するだけです。クライアントは、IPv4 専用サーバーと通信できます。 また、デュアルホストまたは IPv6 専用ホストで実行中の IPv6 サーバーとも通信できます。

ネームサーバーからクライアントが取り出すアドレスで、IPv6 や IPv4 を使用するかどうかが決まります。たとえば、ネームサーバーにそのサーバーの IPv6 アドレスが指定されている場合、サーバーは IPv6 を処理できます。

表 17–1 に IPv4 と IPv6 のクライアントとサーバー間の相互運用性をまとめます。表 17–1 では、デュアルスタックホストに、IPv4 と IPv6 両方のアドレスがそれぞれのネームサービスデータベースに存在するものとします。

表 **17–1** クライアントサーバーアプリケーション: IPv4 と IPv6 の相互運用性

アプリケーションの種類 (ノードの種類)	**IPv6-unaware (非認識) サーバー (IPv4 専用ノード)**	**IPv6-unaware (非認識) サーバー (IPv6 有効化ノード)**	**IPv6-aware (認識) サーバー (IPv6 専用ノード)**	**IPv6-aware (認識) サーバー (IPv6 有効化ノード)**
IPv6-unaware (非認識) クライアント (IPv4 専用ノード)	IPv4	IPv4	X	IPv4
IPv6-unaware (非認識) クライアント (IPv6 有効化ノード)	IPv4	IPv4	X	IPv4
IPv6-aware (認識) クライアント (IPv6 専用ノード)	X	X	IPv6	IPv6
IPv6-aware (認識) クライアント (IPv6 有効化ノード)	IPv4	(IPv4)	IPv6	IPv6

X は、それぞれのサーバーとクライアント間の通信ができないことを表します。

(IPv4) は、クライアントの選択するアドレスによって相互運用性が決まることを表します。IPv6 アドレスを選択すると、クライアントの処理はエラーになります。 ただし、IPv4 アドレスを選択すると、IPv4 マップ IPv6 アドレスとしてクライアントに戻り、IPv4 データグラム が送信されて処理が成功します。

IPv6 配置の初期段階では、IPv6 のほとんどの実装がデュアルスタックノードで処理されます。一般ベンダーではほとんど、初期状態では IPv6 専用実装をリリースしません。

サイト移行のシナリオ

サイトや ISP では、それぞれ事情が異なり、移行段階の手順が異なります。 ここでは、サイト移行のシナリオの例をいくつか紹介します。

IPv6 へのサイトの移行では、最初に IPv6 アドレスをサポートするためのネームサービスをアップグレードします。DNS の場合、BIND 4.9.4 以降などの新しい AAAA (クアドA) レコードをサポートする DNS サーバーにアップグレードします。2 つの新し

い NIS マップと NIS+ テーブルが IPv6 アドレスを保存するために導入されました。NIS マップと NIS+ テーブルを Solaris システムで作成、管理できます。新しいデータベースの詳細については、338 ページの「Solaris ネームサービスに対する IPv6 拡張機能」 を参照してください。

ネームサービスで IPv6 アドレスを処理できるようになったら、ホストの移行を開始します。ホストは、次の手順で移行します。

- ホストを 1 つずつアップグレードします。IPv4 互換アドレスと自動トンネルを使用します。ルーターのアップグレードは不要です。この方法は最初の試験的な移行に適しています。IPv6 の機能のすべてが利用できるわけではありません。したがって、ステートレスアドレス自動設定や IP マルチキャストは利用できません。このシナリオはアプリケーションが IPv6 で実行できるかどうかを確認するときに使用します。また、アプリケーションが IPv6 IP 層セキュリティを利用できるかどうかを確認するときも使用します。
- サブネットを 1 つずつアップグレードします。ルーター間に設定したトンネルを使用します。このシナリオでは、サブネットごとに少なくとも 1 つのルーターをデュアルにアップグレードします。 サイト内のデュアルルーターは設定したトンネル で結合します。 これで、サブネット上のホストでは、IPv6 の全機能を利用できます。このように段階的にアップグレードしていく中で徐々にアップグレードされるルーターが増加するとともに、設定済みのトンネルは削除できます。
- ホストをアップグレードする前にすべてのルーターをデュアルにアップグレードします。この方法は逐次行われるように思えますが、すべてのルーターがアップグレードされるまでは IPv6 の機能を利用できません。このシナリオでは、段階的な配置方式は制約されます。

その他の移行機構

先に説明した方法では、デュアルノードと IPv4 ノード間で相互運用をします。その場合、デュアルノードには IPv4 アドレスがあります。ただし、その方法では、IPv6 専用ノードと IPv4 専用ノードの間で相互運用しません。また、IPv4 アドレスのないデュアルノードと IPv4 専用ノード間でも相互運用しません。 ほとんどの実装ではデュアルにできます。ただし、デュアル実装には、IPv4 専用ノードとの相互運用が必要なすべてのノードごとに 1 つのアドレスを割り当てるのに充分な IPv4 アドレス領域が必要です。

次に、新しい移行機構がなくても相互運用を実現できる方法を示します。

- IPv6 専用ノードとインターネットの他の要素との間にアプリケーション層ゲートウェイ (ALG) を配置する。現在使用されている ALG としては、HTTP プロキシとメールリレーがある
- IPv4 用の NAT ボックス (ネットワークアドレストランスレータ) をすでに売り出している会社もある。これは、内部のプライベート IP アドレス (ネットワーク 10 など。RFC 1918 参照) と外部の IP アドレスの間の変換を行う。このような会社で

は、IPv6 から IPv4 アドレスへの変換もサポートするように、NAT ボックス をアップグレードする可能性が高い

残念ながら、ALG と NAT のどちらの方法も、弱点があります。これらの方法を使用すると、インターネットの基盤がかなり弱まります。 IETF では、IPv6 専用ノードと IPv4 専用ノードとのより良い相互運用性のために努力しています。1 つの提案としては、必要に応じて IPv4 互換アドレスを割り当てる方法でヘッダートランスレータを使用する方法があります。別の方法としては、必要に応じて IPv4 互換アドレスを割り当て、IPv6 トンネルで IPv4 を利用して IPv6 専用ルーターをブリッジできます。

ステートレスヘッダートランスレータでは、使用中の IPv6 アドレスを IPv4 アドレスとして表現できれば、IPv4 ヘッ ダーフォーマットと IPv6 ヘッダーフォーマットの間の変換が可能です。つまり、アドレスは、IPv4 互換または IPv4 マップアドレスである必要があります。これらのトランスレータのサポートは、IPv6 プロトコ ルに組み込まれています。 暗号化されているパケットを除いて、変換時に情報は失われません。ソースルーティングなどの使用頻度の低い機能は、情報が失われてしまうことがあります。

第 18 章

IPsec (トピック)

第 19 章	IPsec の概要
第 20 章	IPsec の設定手順
第 21 章	IKE の概要とその設定手順

第 19 章

IPsec (概要)

IP セキュリティアーキテクチャー (IPsec) は、IPv4 および IPv6 ネットワークパケットで IP データグラムを暗号化して保護します。具体的には、機密性、データ完全性、部分的なシーケンス (再実行) の完全性を確保する機能、データ認証などがあります。IPsec は、IP モジュール内部で実行され、インターネットアプリケーションの知識の有無に関係なく運用できます。正しく使用すれば、IPsec は、ネットワークトラフィックの保護に有効なツールとなります。

この章では、以下の内容について説明します。

IPsec とは

IPsec では、IP 内に安全なデータグラム認証と暗号化の機構を含むセキュリティアソシエーション (SA) を提供します。IPsec を呼び出すと、IPsec グローバルポリシーファイルで有効にしておいた IP データグラムにセキュリティ機構が適用されます。アプリケーションで IPsec を呼び出すと、ソケット単位レベルで IP データグラムにセキュリティ機構が適用されます。

図 19–1は、IPsec を出力パケットで呼び出したときに、IP アドレス指定パケットが IP データグラムの一部として処理されるようすを示します。フロー図からわかるように、認証ヘッダー (AH) とカプセル化されたセキュリティペイロード (ESP) エンティティをパケットに適用できます。そのあとの節では、認証アルゴリズムと暗号化アルゴリズムとともに、これらのエンティティを適用する手順を説明します。

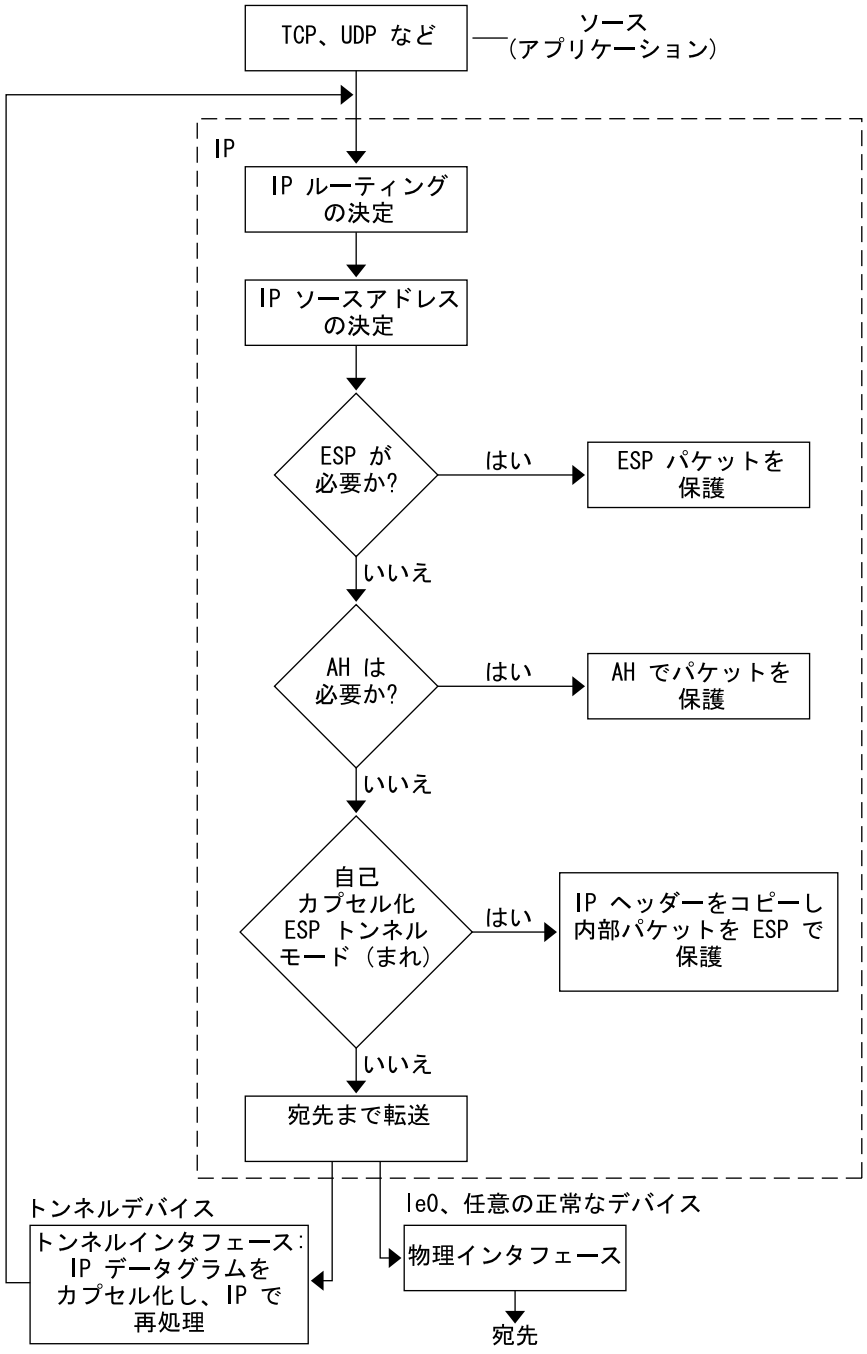


図 **19–1** 出力パケットプロセスに適用された IPsec

図 19–2 は、IPsec 入力プロセスを示したものです。

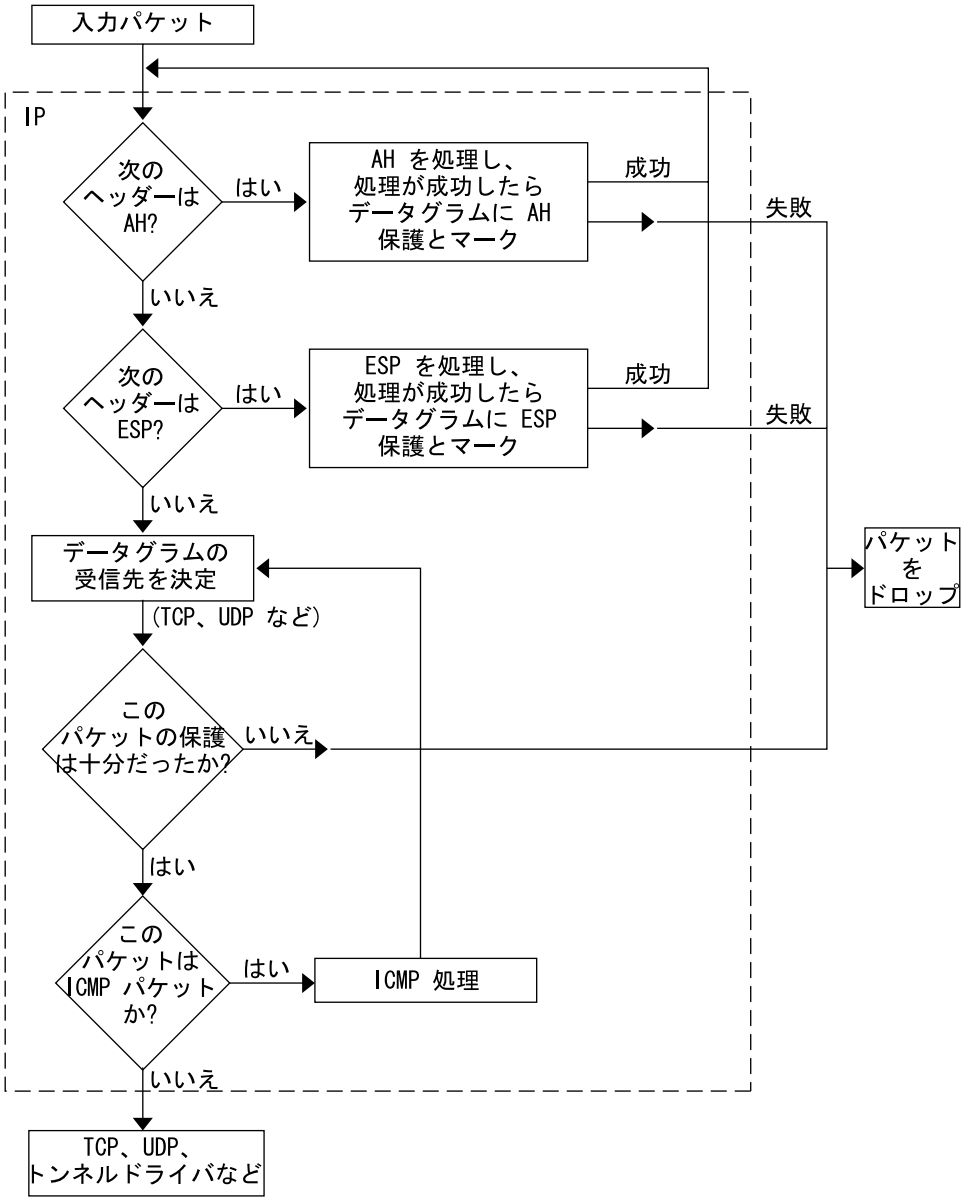


図 **19–2** IPsec を入力パケットプロセスに適用

IPsec セキュリティアソシエーション

IPsec セキュリティアソシエーション (SA) では、ホスト間の通信で認識されるセキュリティ属性を指定します。一般的に、ホスト間で安全に通信するには、2 つの SA が必要です。1 つの SA は、1 方向のデータを保護します。つまり、1 つのホストかグループ (マルチキャスト) アドレスのどちらかです。 ほとんどの通信は、ピアツーピアまたはクライアントとサーバー間で行われるため、両方向のトラフィックを保護するには、2 つの SA が必要です。

AH または ESP 、宛先 IP アドレス、および SPI (セキュリティーパラメータインデックス) は、IPsec SA を識別します。任意の 32 ビット値の SPI は、AH パケットまたは ESP パケットで転送されます。AH および ESP によって保護される範囲については、 `ipsecah(7P)` と `ipsecesp(7P)` のマニュアルページを参照してください。 完全性チェックサム値を使用して、パケットを認証します。認証が失敗すると、パケットがドロップされます。

SA は、SA データベースに保存されます。ソケットベースの管理エンジン `pf_key(7P)` インタフェースにより、特権をもつアプリケーションでそのデータベースを管理できます。`in.iked(1M)` デーモンにより、自動キー管理が可能になります。

キー管理

SA には、キー情報、アルゴリズムの選択、エンドポイントの識別情報、その他のパラメータがあります。認証と暗号化に必要な SA のキー情報の管理をキー管理といいます。IKE (インターネットキー交換) プロトコルにより、キー管理が自動的に行われます。また、`ipseckey(1M)` コマンドを指定して、キー管理を手動で行うこともできます。現在、IPv4 パケットの SA は自動キー管理を使用できますが、IPv6 パケットの SA は手動でキー管理を行う必要があります。

IPv4 ホストの暗号キーを IKE で自動的に管理する方法については、385 ページの「IKE の概要」を参照してください。システム管理者が `ipseckey` コマンドを指定して、暗号キーを手動で管理する方法については、368 ページの「キーユーティリティ」を参照してください。

保護機構

IPsec にはデータ保護機構が 2 つあります。

- 認証ヘッダー (AH)
- セキュリティペイロードのカプセル化 (ESP)

どちらの機構もセキュリティアソシエーションを使用します。

認証ヘッダー

認証ヘッダーは、新しい IP ヘッダーです。強力な完全性、部分的シーケンス完全性 (応答保護)、IP データグラムに対するデータ認証を備えています。AH では対応できる範囲で最大限の IP データグラムを保護します。送信者と受信者の間で不定的に変更されるフィールドは AH では保護できません。たとえば、IP TTL フィールドの変更は予測できないので AH では保護できません。AH は IP ヘッダーとトランスポートヘッダーの間に挿入されます。トランスポートヘッダーの種類としては、TCP、UDP、ICMP、あるいは、トンネルが使用されている場合、もう 1 つ別の IP ヘッダーがあります。トンネルの詳細については、`tun(7M)` のマニュアルページを参照してください。

認証アルゴリズムと AH モジュール

IPsec による実装では、AH は IP の先頭に自動的にプッシュされるモジュールです。`/dev/ipsecah` エントリでは、`ndd(1M)` で AH を調整します。 将来の認証アルゴリズムが AH の先頭にロードできます。現在の認証アルゴリズムには、HMAC-MD5 と HMAC-SHA-1 があります。どちらの認証アルゴリズムにも、それぞれのキーサイズ属性とキーフォーマット属性が用意されています。詳細については、`authmd5h(7M)` と `auhtsha1(7M)` のマニュアルページを参照してください。

セキュリティについて

応答保護を有効にしておかないと、応答時の攻撃が AH をおびやかす原因になります。 AH では盗聴行為には対応できません。AH で保護されたデータであっても、見ようとすれば見ることができます。

セキュリティペイロードのカプセル化

AH によるサービス同様に、ESP でもカプセル化したデータの機密が守られますが、対象はカプセル化したものだけです。ESP の認証サービスはオプションです。これらのサービスでは、冗長になることなく ESP と AH を同じデータグラムで同時に使用できます。ESP は暗号対応技術を使用するため、アメリカ合衆国輸出管理法が適用されます。

ESP はデータをカプセル化し、データグラム内でその先頭に続くデータだけが保護されます。TCP パケットでは、ESP は TCP ヘッダーとそのデータだけをカプセル化します。パケットが IP 内 IP データグラムの場合、ESP は内部 IP データグラムを保護します。ソケット別ポリシーでは、自己カプセル化ができるため、必要に応じて ESP では IP オプションをカプセル化できます。認証ヘッダー (AH) と異なり、ESP では複数

のデータグラム保護が可能です。1 形式だけのデータグラム保護ではデータグラムを守ることはできません。たとえば、ESP で機密は守れますが、機密だけを守っても、応答侵害撃とカットアンドペースト侵害には無防備です。同じく、ESP で完全性だけを保護しても、盗聴に対する対策が不十分なため、その保護能力は AH より弱くなります。

アルゴリズムと ESP モジュール

IPsec ESP では、IP の先頭に自動的にプッシュされるモジュールとして ESP が実装されます。 `/dev/ipsecesp` エントリでは、`ndd(1M)` で ESP を調整します。AH で使用する認証アルゴリズムに加えて、ESP では暗号化アルゴリズムをその先頭にプッシュできます。 暗号化アルゴリズムには、United States Data Encryption Standard (DES)、Triple-DES (3DES)、Blowfish、および AES があります。どちらの暗号化アルゴリズムにも、それぞれのキーサイズ属性とキーフォーマット属性があります。アメリカ合衆国の輸出管理法の適用を受けるので、すべての暗号化アルゴリズムをアメリカ合衆国外で使用できるわけではありません。

セキュリティについて

認証なしで ESP を使用しても、カットアンドペースト暗号化侵害および盗聴侵害に対しては無防備です。 AH の場合と同じく、機密保護なしで ESP を使用しても応答には無防備です。

認証アルゴリズムと暗号化アルゴリズム

IPsec では、認証と暗号化の 2 種類のアルゴリズムを使用します。認証アルゴリズムと DES 暗号化アルゴリズムは、Solaris インストールの主要部分になります。IPsec にサポートされるその他のアルゴリズムを使用する場合には、別の CD によって提供される Solaris Encryption Kit (データ暗号化サプリメント CD) をインストールする必要があります。

認証アルゴリズム

認証アルゴリズムでは、データとキーに基づいて、チェックサム値またはダイジェストが生成されます。認証アルゴリズムのマニュアルページに、ダイジェストとキーのサイズの説明があります。次の表は、Solaris オペレーティング環境でサポートされる認証アルゴリズムを示します。また、IPsec ユーティリティのセキュリティオプションとして認証アルゴリズムを使用する場合のアルゴリズムの形式とそのマニュアルページも示しています。

表 **19–1** サポートされる認証アルゴリズム

アルゴリズムの名前	セキュリティオプションの形式	マニュアルページ
HMAC-MD5	md5、hmac-md5	`authmd5h(7M)`
HMAC-SHA-1	sha、sha1、hmac-sha、hmac-sha1	`authsha1(7M)`

暗号化アルゴリズム

暗号化アルゴリズムでは、キーでデータを暗号化します。暗号化アルゴリズムでは、ブロックサイズごとにデータを処理します。暗号化アルゴリズムのマニュアルページに、ブロックサイズとキーサイズの説明があります。デフォルトでは、DES-CBC アルゴリズムと 3DES-CBC アルゴリズムがインストールされます。IPsec で AES アルゴリズムと Blowfish アルゴリズムを有効にするには、Solaris Encryption Kit をインストールする必要があります。このキットは、Solaris 9 インストールボックスには含まれていない別の CD から入手できます。『*Encryption Kit Installation Guide*』に、Solaris Encryption Kit のインストール方法が説明されています。

次の表に、Solaris オペレーティング環境でサポートされる暗号化アルゴリズムを示します。また、IPsec ユーティリティのセキュリティオプションとして暗号化アルゴリズムを使用する場合のアルゴリズムの形式、そのマニュアルページ、およびそのアルゴリズムが含まれるパッケージも示しています。

表 **19–2** サポートされる暗号化アルゴリズム

アルゴリズムの名前	セキュリティオプションの形式	マニュアルページ	パッケージ
DES-CBC	des、des-cbc	`encrdes(7M)`	SUNWcsr、SUNWcarx.u
3DES–CBC または Triple-DES	3des、3des-cbc	`encr3des(7M)`	SUNWcsr、SUNWcarx.u
Blowfish	blowfish、blowfish-cbc	`encrbfsh(7M)`	SUNWcryr、SUNWcryrx
AES-CBC	aes、aes-cbc	`encraes(7M)`	SUNWcryr、SUNWcryrx

保護ポリシー機構と実施機構

IPsec では、保護ポリシー機構と実施機構を分けています。IPsec ポリシーは、次の範囲で適用できます。

- システム規模レベル

- ソケット単位レベル

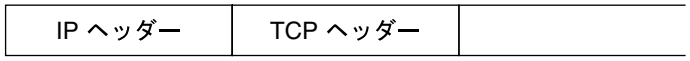
`ipsecconf(1M)` コマンドは、システム規模ポリシーの設定に使用します。

IPsec は、システム規模ポリシーを入力データグラムと出力データグラムに適用します。システムで認識されるデータがあるため、出力データグラムにはその他の規則も適用できます。入力データグラムの処理は、受理されるか拒絶されるかのどちらかです。入力データグラムの受理か拒絶を決定する基準はいくつかありますが、場合によってはその基準が重複したり競合することがあります。競合の解決は、規則の構文解析の順序によって異なります。ただし、ポリシーエントリでトラフィックが他のすべてのポリシーを省略するように指定されている場合は、自動的に受理されます。出力データグラムは、保護付きまたは保護なしで送信されます。保護が適用されると、特定アルゴリズムか汎用アルゴリズムのどちらかになります。ポリシーで標準的にデータグラムを保護する場合、システム規模ポリシーの例外適用時またはソケット単位ポリシーでの省略の要求時に省略できます。

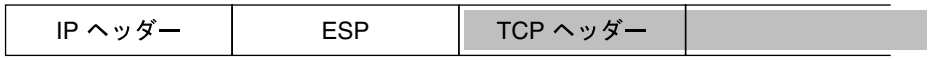
イントラシステム内トラフィックの場合、ポリシーは実施されますが、実際のセキュリティ機構は適用されません。その代わりに、イントラシステム内パケットの出力ポリシーが、セキュリティ機能の適用された入力パケットになります。

トランスポートモードとトンネルモード

IP ヘッダーの後に、ESP または AH を呼び出してデータグラムを保護するときに、トランスポートモードを使用します。たとえば、パケットが次のヘッダーで始まる場合です。



トランスポートモードでは、ESP は次のようにデータを保護します。



□ 暗号化部分

トランスポートモードでは、AH は次のようにデータを保護します。

IP ヘッダー	AH	TCP ヘッダー	

AH はデータがデータグラムに出現する前に、実際データを保護します。その結果、AH による保護は、トランスポートモードでも、IP ヘッダーの一部をカバーします。

データグラム全体が IPsec ヘッダーの保護下にあるとき、IPsec では、トンネルモードでデータグラムを保護しています。AH はその前にある IP ヘッダーの大部分を保護するため、トンネルモードは通常、ESP だけで実行します。先の例のデータグラムは、トンネルモードでは次のように保護されます。

IP ヘッダー	ESP	TCP ヘッダー	TCP ヘッダー

暗号化部分

トンネルモードでは、外部 (保護されていない) IP ヘッダーのソースアドレスと宛先アドレスが、内部 (保護されている) IP ヘッダーのものと異なることがよくあります。それでも、IPsec を認識するネットワークプログラムで ESP の自己カプセル化を使用すれば、内部と外部の IP ヘッダーを一致させることができます。ESP の自己カプセル化により、IP ヘッダーオプションが保護されます。

IPsec の Solaris 実装は基本的にトランスポートモード IPsec 実装であり、トンネルモードはトランスポートモードの特殊ケースとして実装されます。そのため、IP 内 IP トンネルを特殊なトランスポートプロバイダとして処理します。`ifconfig(1M)` 設定オプションを使用してトンネルを設定する場合、オプションは、ソケットのプログラミングでソケットごとの IPsec を使用可能にするときに使用するオプションとほぼ同じです。また、トンネルモードは、ソケットごとの IPsec で使用可能にできます。ソケットごとのトンネルモードでは、内部パケットの IP ヘッダーのアドレスが外部パケットの IP ヘッダーのアドレスと同じになります。ソケットごとのポリシーの詳細については、`ipsec(7P)` のマニュアルページを参照してください。

信頼性の高いトンネル

設定したトンネルは、ポイントツーポイントインタフェースです。これで、IP パケットを IP パケット内にカプセル化できます。トンネルの設定には、トンネルソースとトンネル宛先が必要です。詳細については、`tun(7M)` のマニュアルページと、337 ページの「IPv6 の Solaris トンネルインタフェース」を参照してください。

トンネルでは、IP との見かけ上の物理的インタフェースが作成されます。この物理的リンクの完全性は、基本になるセキュリティプロトコルによって異なります。セキュリティアソシエーションを確実に行えば、信頼性の高いトンネルになります。すなわち、トンネルのデータパケットのソースはトンネル宛先で指定したピアになります。この信頼関係があるかぎり、インタフェース別 IP 送信を利用して仮想プライベートネットワークを作成できます。

仮想プライベートネットワーク

IPsec を使用して、VPN (仮想プライベートネットワーク) を構築できます。 そのためには、インターネットインフラストラクチャを使用してイントラネットを作成します。たとえば、それぞれのネットワークとともに独立したオフィスを持つ組織があって、オフィス間が VPN テクノロジで接続されている場合、IPsec を利用すれば、2 つのオフィス間でトラフィックを安全にやりとりできます。

図 19–3 は、ネットワークシステムに配置した IPsec で、2 つのオフィスがインターネットを利用して VPN を形成する方法を示します。

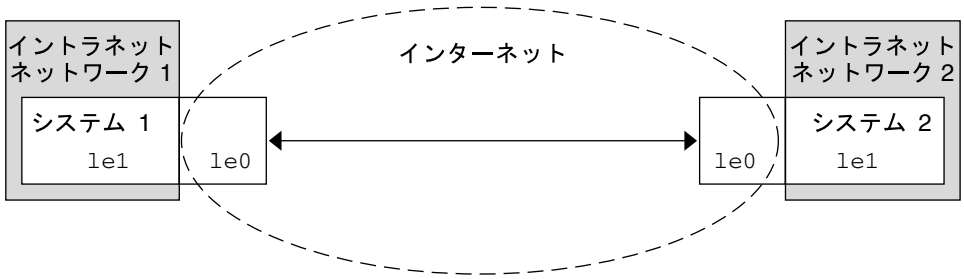


図 **19–3** 仮想プライベートネットワーク

セットアップ手順については、378 ページの「仮想プライベートネットワークの構築」を参照してください。

IPsec ユーティリティおよび IPsec ファイル

この節では、IPsec の初期化構成ファイルと、ネットワーク内で IPsec の管理を行うためのさまざまなコマンドについて説明します。ネットワーク内で IPsec を実装する方法については、373 ページの「IPsec の実装 (作業マップ)」を参照してください。

表 19–3 選択される IPsec ファイルと IPsec コマンドのリスト

IPsec ファイルまたは IPsec コマンド	説明
/etc/inet/ipsecinit.conf ファイル	IPsec ポリシーファイル。このファイルがある場合、IPsec はブート時に起動する
ipsecconf コマンド	IPsec 起動コマンド。 引数として ipsecinit.conf ファイルで ipsecconf を呼び出すと、IPsec ポリシーが起動します。現在の IPsec ポリシーの表示および変更や、テストを行うときに役立ちます。
pf_key() インタフェース	SA データベースのインタフェース。 手動キー管理および自動キー管理を処理する
ipseckey コマンド	IPsec SA で使用するキーの起動コマンド。 ipseckey を指定すると、IPsec SA のキー情報が表示されます。
/etc/inet/secret/ipseckeys ファイル	IPsec SA のキー。ipsecinit.conf ファイルがある場合、このファイルはブート時に自動的に読み込まれます。
/etc/inet/ike/config ファイル	IKE 構成およびポリシーファイル。このファイルがある場合、IKE デーモン により、/etc/inet/ike/config ファイルが開始およびロードされます。388 ページの「IKE ユーティリティおよび IKE ファイル」を参照してください。

IPsec ポリシーコマンド

ipsecconf(1M) コマンドを使用して、ホストの IPsec ポリシーを構成します。 このコマンドを実行してポリシーを構成すると、IPsec ポリシーエントリを保存する ipsecpolicy.conf という名前の一時ファイルが作成されます。そのファイルを使用して、すべての外方向および内方向の IP データグラムがポリシーに沿っているかが検査されます。転送されたデータグラムは、このコマンドで追加されたポリシー検査の対象外になります。転送されたパケットを保護する方法については、ifconfig(1M) と tun(7M) のマニュアルページを参照してください。

ipsecconf コマンドは、スーパーユーザーとして実行する必要があります。このコマンドは、両方向のトラフィックを保護するエントリ、および 1 方向のみのトラフィックを保護するエントリを受け入れます。

方向を指定しないポリシーエントリに laddr host1 (ローカルアドレス) と raddr host2 (リモートアドレス) というパターンが含まれていると、指定されたホストに対して両方向のトラフィックが保護されます。そのため、各ホストにエントリを 1 つだけ設定すれば済みます。saddr host1 daddr host2 (ソースアドレスから宛先アドレスへ) というパターンのポリシーエントリは、1 方向のみのトラフィックを保護します。つまり、外方向または内方向のどちらかです。したがって、両方向のトラフィックを保護するには、saddr host2 daddr host1 という先ほどとは逆方向のエントリを ipsecconf コマンドに渡す必要があります。

引数を指定しないで ipsecconf コマンドを実行すると、システムに構成されているポリシーを確認できます。各エントリが、インデックスとその後に番号が付いて表示されます。-d オプションでインデックスを指定すると、システム内の指定されたポリシーが削除されます。このコマンドで表示されるエントリの順序はエントリが追加された順であり、必ずしもトラフィックを照合する順序ではありません。トラフィックの照合が行われる順序を確認するには、-l オプションを使用します。

ipsecpolicy.conf ファイルは、システムのシャットダウン時に削除されます。マシンのブート時に IPsec ポリシーを起動させるには、マシンのブート時に inetinit スクリプトによって読み込まれる IPsec ポリシーファイル /etc/inet/ipsecinit.conf を作成する必要があります。

IPsec ポリシーファイル

Solaris オペレーティング環境を起動したときに IPsec セキュリティポリシーを呼び出すには、特定の IPsec エントリを利用して IPsec 初期化構成ファイルを作成します。ファイルの名前は、/etc/inet/ipsecinit.conf とします。ポリシーエントリとその形式の詳細については、ipsecconf(1M) のマニュアルページを参照してください。 ポリシーの構成後、ipsecconf コマンドを使用してポリシーを一時的に削除したり、既存の構成を表示したりすることができます。

例 – ipsecinit.conf ファイル

Solaris ソフトウェアには、サンプルの ipsecinit.conf ファイルが組み込まれており、自分で ipsecinit.conf ファイルを作成するときのテンプレートとして利用できます。このサンプルの名前は、ipsecinit.sample であり、次のエントリを含みます。

```
#
#ident    "@(#)ipsecinit.sample    1.6  01/10/18 SMI"
#
# Copyright (c) 1999,2001 by Sun Microsystems, Inc.
# All rights reserved.
#
# This file should be copied to /etc/inet/ipsecinit.conf to enable IPsec
# systemwide policy (and as a side-effect, load IPsec kernel modules).
# Even if this file has no entries, IPsec will be loaded if
# /etc/inet/ipsecinit.conf exists.
#
# Add entries to protect the traffic using IPsec. The entries in this
# file are currently configured using ipsecconf from inetinit script
# after /usr is mounted.
#
# For example,
#
#     {rport 23} ipsec {encr_algs des encr_auth_algs md5}
#
# Or, in the older (but still usable) syntax
```

```
#
#    {dport 23} apply {encr_algs des encr_auth_algs md5 sa shared}
#    {sport 23} permit {encr_algs des encr_auth_algs md5}
#
# will protect the telnet traffic originating from the host with ESP using
# DES and MD5. Also:
#
#     {raddr 10.5.5.0/24} ipsec {auth_algs any}
#
# Or, in the older (but still usable) syntax
#
#    {daddr 10.5.5.0/24} apply {auth_algs any sa shared}
#    {saddr 10.5.5.0/24} permit {auth_algs any}
#
# will protect traffic to or from the 10.5.5.0 subnet with AH
# using any available algorithm.
#
#
# To do basic filtering, a drop rule may be used. For example:
#
#    {lport 23 dir in} drop {}
#    {lport 23 dir out} drop {}
#
# will disallow any remote system from telnetting in.
#
#
# WARNING:    This file is read before default routes are established, and
#        before any naming services have been started. The
#        ipsecconf(1M) command attempts to resolve names, but it will
#        fail unless the machine uses files, or DNS and the DNS server
#        is reachable via routing information before ipsecconf(1M)
#        invocation.  (that is, the DNS server is on-subnet, or DHCP
#        has loaded up the default router already.)
#
#        It is suggested that for this file, use hostnames only if
#        they are in /etc/hosts, or use numeric IP addresses.
#
#        If DNS gets used, the DNS server is implicitly trusted, which
#        could lead to compromise of this machine if the DNS server
#        has been compromised.
#
```

セキュリティについて

たとえば、/etc/inet/ipsecpolicy.conf ファイルを、NFS マウントファイルシステムから送信すると、ファイル内のデータが不正に変更される可能性があります。また、設定ポリシーも変更される可能性があります。そのため、ipsecinit.conf ファイルのコピーをネットワークで送信しないでください。

ポリシーは、connect(3SOCKET) または accept(3SOCKET) を実行した TCP/UDP ソケットに対して変更することはできません (“ラッチ”されます)。 新しいポリシーエントリを追加しても、ラッチされたソケットは変更されません。 このラッチ機能は将来変更される可能性があるので、この機能に依存するような設定をしないでください。

ポリシーは通信を開始する前にセットアップしてください。新しいポリシーエントリを追加すると既存の接続が影響を受けることがあるためです。同じ理由から、通信の途中ではポリシーを変更しないでください。

ネットワークで参照できるホストがソースアドレスで、指定システム自体の安全性に問題がある場合、使用される名前は信頼できません。

セキュリティの弱点は、ツール自体ではなく、ツールの使用方法にあります。ipsecconf コマンドを使用するときは注意が必要です。各操作の最も安全なモードでコンソールを使用するか、ハード接続の TTY を使用してください。

IPsec セキュリティアソシエーションデータベース

IPsec セキュリティサービスのキー情報は、セキュリティアソシエーションデータベース (SADB) に保存されます。 セキュリティアソシエーションは、入力パケットと出力パケットを保護します。ユーザープロセス (場合によってはマルチ連携プロセス) では、特殊なソケットからのメッセージを送信することで SADB を管理します。これは、route(7P) のマニュアルページで説明した方法に類似しています。SADB にアクセスできるのはスーパーユーザーだけです。

出力データグラムの新しい SA に対する要求などの外部イベントに対する応答として、あるいは既存の SA の期限切れを報告するために、オペレーティングシステムからメッセージが自動的に発信されることがあります。先に説明したソケットコールを使用して、SADB 制御メッセージを伝えるためのチャンネルを開いてください。システムごとに複数のキーソケットを開くことができます。

メッセージには、小さいベースヘッダー、その後に続いて多くの (0 以上) 拡張メッセージが含まれています。メッセージの中には、追加データが必要なものもあります。ベースメッセージと拡張メッセージのいずれも 8 バイト配列である必要があります。たとえば GET メッセージの場合、ベースヘッダー、SA 拡張メッセージ、ADDRESS_DST 拡張メッセージが必要です。詳細については、pf_key(7P) を参照してください。

キーユーティリティ

IKE プロトコルは、IPv4 アドレスの自動キーユーティリティです。IKE の設定方法については、第 21 章 を参照してください。 手動でキーを操作するユーティリティには、ipseckey(1M) コマンドがあります。

ipseckey コマンドを使用して、ipsecah(7P) と ipsecesp(7P) の保護機構で SA データベースを手動で操作できます。 また、自動キー管理が無効な場合に、通信パーティ間の SA をセットアップするときも、ipseckey コマンドを使用します。例としては、IPv6 アドレスを持つ通信パーティ間が挙げられます。

ipseckey コマンドには少数の一般オプションしかありませんが、多くのコマンド言語をサポートしています。マニュアルキー操作に固有のプログラムインタフェースで要求を配信するように指定することもできます。詳細については、pf_key(7P) のマニュアルページを参照してください。引数なしで ipseckey を呼び出すと、対話モードになり、エントリを入力できるプロンプトが表示されます。コマンドによっては、明示的なセキュリティアソシエーション (SA) タイプが必要ですが、それ以外は、ユーザーが SA を指定すれば、すべての SA タイプで動作します。

セキュリティについて

ipseckey コマンド使用すると、特権ユーザーは微妙な暗号キー情報を入力できます。 場合によっては、不正にこの情報にアクセスして IPsec トラフィックのセキュリティを損なうことも可能です。キー情報を扱う場合および ipseckey コマンドを使用する場合には、次のことに注意してください。

1. キー情報を更新しているかどうか。定期的にキーを更新することが、セキュリティの基本作業となります。キーを変更することで、アルゴリズムとキーの脆弱性が暴かれないように保護し、公開されたキーの侵害を制限します。
2. TTY がネットワークに接続されているか (対話モードになっているか)。
   - TTY が対話モードの場合、キー情報のセキュリティは、TTY のトラフィックに対応するネットワークパスのセキュリティになります。 clear-text telnet や rlogin セッションでは、ipseckey コマンドを使用しないでください。
   - ローカルウィンドウでも、ウィンドウを読み取ることのできる隠密プログラムからの攻撃には無防備です。
3. ファイルがネットワーク経由でアクセス状態にあるか、または外部から読み取り可能な状態になっているか (-f オプション)。
   - ネットワークマウントファイルの読み取り時に、不正に読み取ることができます。外部から読み取れるファイルにキー情報を保存して使用しないでください。
   - ネットワークで参照できるホストがソースアドレスで、指定システム自体の安全性に問題がある場合、使用される名前は信用できません。

セキュリティの弱点は、ツール自体ではなく、ツールの使用方法にあります。ipseckey コマンドを使用するときには注意が必要です。各操作の最も安全なモードでコンソールを使用するか、ハード接続の TTY を使用してください。

その他のユーティリティに対する IPsec 拡張機能

ifconfig コマンドには、トンネルインタフェースで IPsec ポリシーを管理するオプションがあります。また、snoop コマンドを使用して AH ヘッダーと ESP ヘッダーを構文解析できます。

ifconfig コマンド

IPsec をサポートするため、ifconfig(1M) に次のオプションが追加されました。

- auth_algs
- encr_auth_algs
- encr_algs

auth_algs

このオプションを設定すると、指定した認証アルゴリズムで、トンネルにIPsec AH を使用できます。auth_algs オプションの書式は次のとおりです。

auth_algs *authentication_algorithm*

アルゴリズムは番号またはアルゴリズム名です。特定のアルゴリズムが指定されないようにするパラメータ *any* も使用できます。IPsec トンネル属性は、すべて同じコマンド行に指定します。トンネルセキュリティを無効にするには、次のオプションを指定します。

auth_alg none

サポートされる認証アルゴリズムとその詳細を説明したマニュアルページのリストについては、表 19–1 を参照してください。

encr_auth_algs

このオプションでは、認証アルゴリズムを指定してトンネルの IPsec ESP を有効にします。encr_auth_algs オプションの書式は次のとおりです。

encr_auth_algs *authentication_algorithm*

アルゴリズムには、番号またはアルゴリズム名を指定できます。特定のアルゴリズムが指定されないようにするパラメータ *any* も使用できます。ESP 暗号化アルゴリズムを指定し、認証アルゴリズムを指定しない場合、ESP 認証アルゴリズム値はデフォルトのパラメータ *any* になります。

サポートされる認証アルゴリズムとそのアルゴリズムの詳細を説明したマニュアルページのリストについては、表 19–1 を参照してください。

encr_algs

このオプションでは、暗号化アルゴリズムを指定したトンネルで IPsec ESP を有効にできます。オプションの書式は次のとおりです。

```
encr_algs encryption_algorithm
```

このアルゴリズムの場合、番号またはアルゴリズム名を指定できます。IPsec トンネル属性は、すべて同じコマンド行に指定します。トンネルセキュリティを無効にするには、次のオプションを指定します。

```
encr_alg none
```

ESP 認証アルゴリズムを指定し、暗号化アルゴリズムを指定しない場合、ESP 暗号化アルゴリズム値はデフォルトのパラメータ *null* になります。

サポートされる暗号化アルゴリズムとその詳細を説明したマニュアルページのリストについては、`ipsecesp(7P)` のマニュアルページまたは表 19–2 を参照してください。

snoop コマンド

`snoop` コマンドでも、AH ヘッダーと ESP ヘッダーを構文解析できるようになりました。ESP はそのデータを暗号化するので、`snoop` は ESP で暗号化されて保護されたヘッダーを読み取ることができませんが、AH ではデータは暗号化されないので、`snoop` でトラフィックを確認できます。パケットに AH が使用されている場合、`snoop -V` オプションで表示できます。詳細については、`snoop(1M)` のマニュアルページを参照してください。

第 20 章

IPsec の管理 (手順)

この章では、ネットワークに IPsec を実装する手順について説明します。

この章では、以下の内容について説明します。

IPsec の概要については、第 19 章を参照してください。`ipsecconf(1M)`、`ipseckey(1M)`、`ifconfig(1M)` の各マニュアルページにも、個別の例に応じた説明があります。

IPsec の実装 (作業マップ)

表 20–1 IPsec の実装 (作業マップ)

タスク	説明	操作方法の掲載箇所
IPv6 システム間のトラフィックの保護	`/etc/inet/ipnodes` ファイルに対するアドレスの追加、`/etc/inet/ipsecinit.conf` ファイルに対する IPsec ポリシーの入力、`ipseckey` コマンドを使用した手動によるキーの追加、`ipsecinit.conf` ファイルの呼び出し	374 ページの「2 つのシステム間のトラフィックを保護」

表 **20–1** IPsec の実装 (作業マップ)　(続き)

タスク	説明	操作方法の掲載箇所
IPsec ポリシーによる Web サーバーの保護	`ipsecinit.conf` ファイルに対するさまざまなポートの異なるセキュリティ要件の入力とそのファイルの呼び出しによる、保護トラフィックだけの有効化	377 ページの「Web サーバーの保護方法」
仮想プライベートネットワークのセットアップ	IP 送信のオフ、IP の厳密宛先マルチホーム、大半のネットワークサービスとインターネットサービスの無効化、セキュリティアソシエーションの追加、保護トンネルの設定、IP 送信のオン、デフォルトルートの設定、ルーティングプロトコルの実行	378 ページの「仮想プライベートネットワークの構築」
現在のセキュリティアソシエーションの変更	現在のセキュリティアソシエーションのフラッシュと、影響する各システムにおける新しいセキュリティアソシエーションの入力	382 ページの「現在のセキュリティアソシエーションの変更」

IPsec 作業

この節では、2 つのシステム間のトラフィックを保護し、IPsec ポリシーで Web サーバーを保護し、仮想プライベートネットワークをセットアップするための手順について説明します。システム名 `enigma` と `partym` は一例として使用しているだけです。よって、`enigma` と `partym` を各自使用しているシステムの名前に置き換えてください。

▼ 2 つのシステム間のトラフィックを保護

この手順を行う前に、各システムに 2 つのアドレス (IPv4 アドレスと IPv6 アドレス) があり、有効なアルゴリズムのいずれかを使用して AH (Authentication Header) 保護を呼び出しているものとします。 また、SA (Security Association) は共有されているものとします。 すなわち、2 つのシステムを保護するのに必要なのは 1 組だけの SA です。

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

---

2. システムごとに、他のシステムのアドレスとホスト名を **/etc/inet/ipnodes** ファイルに追加します。次のように、**1** つのシステムのエントリは連続してそのファイルに入力します。

   a. **partym** という名前のシステムでは、次のように入力します。

   ```
   # Secure communication with enigma
   192.168.66.1 enigma
   fec0::10:20ff:fea0:21f7 enigma
   ```

   b. **enigma** という名前のシステムでは、次のように入力します。

   ```
   # Secure communication with partym
   192.168.55.2  partym
   fec0::9:a00:20ff:fe7b:b667 partym
   ```

   システムの名前は、一例として使用しているだけです。実際にシステム間のトラフィックを保護する場合には、各自のシステムの名前を使用してください。

   これで、起動スクリプトでは、存在しないネーミングサービスに依存することなくシステム名を使用できます。

3. システムごとに、**/etc/inet/ipsecinit.conf** ファイルを編集して **IPsec** ポリシーエントリを追加します。

   a. **enigma** システムでは、次のように入力します。

   ```
   {laddr enigma raddr partym} ipsec {auth_algs any sa shared}
   ```

   b. **partym** システムでは、次のように入力します。

   ```
   {laddr partym raddr enigma} ipsec {auth_algs any sa shared}
   ```

4. システムごとに、**2** つのシステム間のセキュリティアソシエーションの組を追加します。

   システムごとに、読み取り専用 (600 のアクセス権) の /etc/inet/secret/ipseckeys ファイルを編集し、このファイルに次の行を入力します。

   ```
   add ah spi random-number dst local-system authalg an_algorithm_name \
           authkey random-hex-string-of-algorithm-specified-length
   add ah spi random-number dst remote-system authalg an_algorithm_name \
           authkey random-hex-string-of-algorithm-specified-length
   ```

   注 – キーと SPI は、セキュリティアソシエーションごとに変更できますが、同じにしてはいけません。

   ---

5. 各システムをリブートします。

   ```
   # /etc/reboot
   ```

   リブートでは、起動プロセスの終了前に /etc/inet/secret/ipseckeys ファイルが読み取られます。 キーを変更する場合は、両方のシステムで

ipseckeys ファイルが変更されていることを確認してください。

例 — IPv4 アドレス間のトラフィックの保護

次の例では、IPv4 アドレスを持つシステム間のトラフィックを保護する方法について説明します。この例では、自動キー管理 (IKE) を使用してセキュリティアソシエーションを作成します。IKE は、管理者が介在する必要が少なく、大量のトラフィックを容易に保護するようにスケーリングします。

1. 次のように、374 ページの「2 つのシステム間のトラフィックを保護」にある手順 2 の /etc/inet/ipnodes ファイルを /etc/hosts ファイルに置き換えます。

   partym という名前のシステムでは、次のように enigma を追加します。

   ```
   # echo "192.168.66.1 enigma">> /etc/hosts
   ```

   enigma という名前のシステムでは、次のように partym を /etc/hosts ファイルに追加します。

   ```
   # echo "192.168.55.2 partym">> /etc/hosts
   ```

2. ipsecinit.conf ファイルを編集して IPsec ポリシーエントリを追加します。
3. ipseckey コマンドではなく、ike.config(4) ファイルを使用して、セキュリティアソシエーションを追加します。 この手順については、393 ページの「IKE 作業」を参照してください。

   注 – 374 ページの「2 つのシステム間のトラフィックを保護」の手順 4 で説明されているように、キーを手動で作成することもできます。

   ---

4. リブートします。

例 — リブートなしでの IPv6 アドレス間のトラフィックの保護

次の例では、IPv6 アドレスを持つシステム間の保護トラフィックをテストする方法について説明します。

1. 374 ページの「2 つのシステム間のトラフィックを保護」の手順 4 まで実行します。
2. リブートしないで、引数として ipseckeys ファイルを指定する ipseckey コマンドを入力することで、セキュリティアソシエーションをデータベースに追加します。

   ```
   # ipseckey -f /etc/inet/secret/ipseckeys
   ```

3. 次のように、ipsecconf コマンドを使用して IPsec ポリシーを有効にします。

```
# ipsecconf -a /etc/inet/ipsecinit.conf
```

注 – このコマンドの実行時には警告を読んでください。ソケットがすでに使用中 (ラッチされた) の場合には、システムのセキュリティが低下します。

▼ Web サーバーの保護方法

セキュリティ保護された Web サーバーでは、Web クライアント要求でないすべての入力トラフィックが、セキュリティ検査を通ることが要求されます。 この手順では、Web サーバーで処理する Web トラフィックと、この Web サーバーからの DNS クライアント要求の省略 (bypass) について説明します。他のすべてのトラフィックには、3DES アルゴリズムと SHA-1 アルゴリズムでは ESP を要求し、出力トラフィックに共有 SA を使用します。また、SA の共有により、セキュリティアソシエーションが多くなり過ぎないようにします。

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

---

2. セキュリティポリシー検査を省略するサービスを指定します。

   Web サーバーの場合、TCP ポート 80 (HTTP) と 443 (保護 HTTP) が該当します。Web サーバーが DNS 名検査をするときは、TCP と UDP の両方にポート 53 も組み込む必要がある場合もあります。

3. たとえば、**IPsecWebInitFile** のように、選択したファイル名で読み取り専用ファイルを作成し、このファイルに次の行を入力します。

```
# Web traffic that Web server should bypass.
    {sport 80 ulp tcp} bypass {dir out}
    {dport 80 ulp tcp} bypass {dir in}
    {sport 443 ulp tcp} bypass {dir out}
    {dport 443 ulp tcp} bypass {dir in}

    # Outbound DNS lookups should also be bypassed.
    {dport 53} bypass {dir out}
    {sport 53} bypass {dir in}

    # Require all other traffic to use ESP with 3DES and SHA-1.
    # Use a shared SA for outbound traffic, in order to avoid a
    # large supply of security associations.
    {} permit {encr_algs 3des encr_auth_algs sha}
    {} apply {encr_algs 3des encr_auth_algs sha sa shared}
```

これで、保護トラフィックだけがシステムにアクセスできるようになります。ただし、先の手順で説明した、検査を省略するトラフィックは例外です。

4. 先の手順で作成したファイルを **/etc/inet/ipsecinit.conf** に読み込みます。

```
# vi  /etc/inet/ipsecinit.conf
 :r IPsecWebInitFile
 :wq!
```

5. リブートします。

   ipsecconf コマンドを呼び出しても、すでに確立した TCP 接続には影響せず、そのポリシーはラッチされます。システムをリブートすると、IPsec ポリシーがすべての TCP 接続に適用されます。リブート時に、 IPsec ポリシーのファイルで指定したように TCP 接続でポリシーがラッチされます。

```
# reboot
```

   こうして、Web サーバーでは、Web サーバートラフィックと出力 DNS 要求と応答だけを処理します。他のサービスは、IPsec をリモートシステムで有効にしないと機能しません。キー情報を自動的に処理する場合には、IKE デーモンにより、IPv4 アドレスを持つリモートシステムで IPsec を有効にします。 IPv6 アドレスを持つリモートシステムでは、ipseckey(1M) コマンドを使用してリモートシステムで IPsec を有効にします。

▼ 仮想プライベートネットワークの構築

この手順では、インターネットで VPN を構築して組織内の 2 つのネットワークを接続する方法について説明します。また、そのネットワーク間のトラフィックを IPsec で保護する方法について説明します。前提条件として、VPN リンクを実装した 2 つのシステム上で、ネットワークの le1 インタフェースは VPN 内部にあり、le0 インタフェースは VPN 外部にあるものとします。

また、この操作では、DES と MD5 で ESP を使用します。使用するアルゴリズムによってキーの長さが異なり、DES の場合は 64 ビット (56 ビット + 8 ビットパリティ)、MD5 の場合は 128 ビットになります。インターネットでゲートウェイになる 2 つのシステムには、次の操作をします。VPN については、364 ページの「仮想プライベートネットワーク」を参照してください。

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

---

2. 次のコマンドを入力して **IP** 送信をオフにします。

```
# ndd -set /dev/ip ip_forwarding 0
```

IP 送信をオフにすると、このシステムを経由したネットワーク間のパケット送信ができなくなります。

3. 次のコマンドを入力して **IP** の厳密宛先マルチホームをオンにします。

```
# ndd -set /dev/ip ip_strict_dst_multihoming 1
```

IP 厳密宛先マルチホームをオンにすると、システムの宛先アドレスのうちの 1 つに宛てたパケットは、そのアドレスを割り当てたインタフェースに必ず到着します。

`ndd(1M)` コマンドを使用して IP 送信をオフにし、IP 厳密宛先をオンにすると、マルチホームによってシステム自体へのパケット以外はすべてパケットがシャットダウンされ、宛先 IP アドレスに対応するインタフェースにだけパケットが到着します。

4. 必要に応じて、次の手順で **Solaris** システム上の大部分 **(**すべてでない場合**)** のネットワークサービスを無効にします。

注 **–** VPN ルーターは、ほとんどの入力要求を受け付けません。入力トラフィックを受け付けるすべてのプロセスを無効にする必要があります (`inetd.conf` ファイルの行をコメントにするか、SNMP を終了するなど)。 また、377 ページの「Web サーバーの保護方法」のような方法を実行する必要があります。

---

a. **`inetd.conf`** を編集して、重要なサービス以外のすべてのサービスを削除した場合、次のコマンドを入力します。

```
# pkill -HUP inetd
```

b. 重要なサービス以外のすべてのサービスを削除するための **`inetd.conf`** の編集をしていない場合は、次のコマンドを入力します。

```
# pkill inetd
```

c. 必要に応じて、次の例のようなコマンドを **1** つまたは複数入力して **SNMP**、**NFS** など他のインターネットサービスを無効にします。

```
# /etc/init.d/nfs.server stop
# /etc/init.d/sendmail stop
```

ネットワークサービスを無効にすると、IP パケットによるシステムへの妨害がなくなります。たとえば、SNMP デーモン、telnet、rlogin を最大限に活用できます。

5. システムごとに、**2** つのシステム間のセキュリティアソシエーションの組を追加します。

システムで IPv4 アドレスを使用している場合には、セキュリティアソシエーションを作成するように IKE を設定すると、IKE デーモンにより、セキュリティアソシエーションが自動的に作成されます。VPN に IKE を設定するには、394 ページの「事前共有鍵による IKE の設定方法」、399 ページの「自己署名付き公開証明書による IKE の設定方法」、または401 ページの「認証局による署名付き

公開鍵による IKE の設定方法」のいずれかの手順を実行します。

システムで IPv6 アドレスを使用している場合には、次の手順を実行して手動でセキュリティアソシエーションを作成する必要があります。

a. 次のコマンドを入力して **ipseckey** コマンドモードを有効にします。

```
# ipseckey
>
```

> プロンプトは、ipseckey コマンドモードになったことを示します。

b. 次のコマンドを入力します。

```
> add esp spi random-number src system1_addr dst system2_addr \
auth_alg md5 encr_alg des \
authkey random-hex-string-of-32-characters \
encrkey random-hex-string-of-16-characters
```

c. **Return** キーを押してコマンドを実行します。

d. 次のコマンドを入力します。

```
> add esp spi random-number src system2_addr dst system1_addr \
auth_alg md5 encr_alg des \
authkey random-hex-string-of-32-characters \
encrkey random-hex-string-of-16-characters
```

注 – キーと SPI は、セキュリティアソシエーションごとに変更できますが、同じにしてはいけません。

---

e. **Ctrl-D** キーを押すか、 **quit** を入力してこのモードを終了します。

6. 次の手順を実行して、**IP** から見たもう **1** つの物理的インタフェースを追加する保護トンネル **ip.tun0** を設定します。

a. システム **1** で、次のコマンドを入力します。

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 system1-taddr system2-taddr \
tsrc system1-addr tdst system2-addr encr_algs des encr_auth_algs md5

# ifconfig ip.tun0 up
```

b. システム **2** で、次のコマンドを入力します。

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 system2-taddr system1-taddr \
tsrc system2-addr tdst system1-addr encr_algs des encr_auth_algs md5

# ifconfig ip.tun0 up
```

7. システムごとに、次のコマンドを入力して **le1:ip_forwarding** と **ip.tun0:ip_forwarding (**この例では**)** をオンにします。

```
# ndd -set /dev/ip le1:ip_forwarding 1

# ndd -set /dev/ip ip.tun0:ip_forwarding 1
```

ip_forwarding は、インタフェースから到着したパケットを転送できることを意味します。またこのインタフェースから転送されるパケットは別のインタフェースが発信元であることを表します。パケットを正しく転送するには、受信インタフェースと送信インタフェースの ip_forwarding をオンにしておきます。

le1 はイントラネットの内部にあり、ip.tun0 はインターネットを経由して 2 つのシステムを接続するので、これら 2 つのインタフェースでは、ip_forwarding をオンにしておきます。

le0 インタフェースの ip_forwarding はまだオフです。そのため、外部 (インターネット内) からパケットが保護イントラネットに侵入するのを防ぐことができます。

8. システムごとに、次のコマンドを入力して、ルーティングプロトコルによってイントラネット内のデフォルトのルートが通知されていないことを確認します。

```
# ifconfig le0 private
```

le0 の ip_forwarding がオフになっていても、ルーティングプロトコルの実装のどれか (in.routed など) で、le0 がイントラネット内のピアにパケットを転送するときの有効なインタフェースであることが通知されている可能性があります。インタフェースの private フラグを設定すれば、この通知を削減できます。

9. システムごとに、次のコマンドを入力して **le0** 経由のデフォルトルートを手動で追加します。

```
# pkill in.rdisc

# route add default router-on-le0-subnet
```

le0 はイントラネットの一部ではありませんが、インターネットを介してそのピアマシンにアクセスする必要があります。そのため、インターネットルーティング情報が必要です。インターネットの残りの要素にとって、VPN システムはルーターに対するホストのようなものなので、デフォルトルーターを使用するか、ルーター発見を実行すれば十分です。

10. システムが再起動するときに、**in.rdisc** が再開するのを防ぐため次の操作をします。

    a. **le0** サブネットのデフォルトルーターの **IP** アドレスを **/etc/defaultrouter** ファイルに指定します。

    この手順により、in.rdisc がリブート時に開始しなくなります。

    b. 起動シーケンスの初期にルーティングが起こらないようにし、セキュリティの脆弱性が軽減されます。

```
# touch /etc/notrouter
```

c. **/etc/hostname.ip.tun0** ファイルを編集して次の行を追加します。

```
system1-taddr system2-taddr tsrc system1-addr \
     tdst system2-addr encr_algs des encr_auth_algs md5 up
```

d. **/etc/rc3.d/S99vpn_setup** ファイルを作成して、次の行を入力します。

```
ndd -set /dev/ip le1:ip_forwarding 1
ndd -set /dev/ip ip.tun0:ip_forwarding 1
ifconfig le0 private
in.routed
```

11. システムごとに、次のコマンドを入力してルーティングプロトコルを実行します。

```
# in.routed
```

暗号システムの不正侵入者の時間的な余裕をなくすため、手順 5 で作成するセキュリティアソシエーションは定期的に新しいセキュリティアソシエーションに変更します。現在のセキュリティアソシエーションを変更するには、次の手順を実行します。IPv4 ネットワークを実行している場合、IKE モジュールでセキュリティアソシエーションの変更が管理されます。

▼ 現在のセキュリティアソシエーションの変更

この手順では、現在のセキュリティアソシエーションを変更します。暗号システムの不正侵入者の時間的な余裕をなくすためにこの操作を定期的に行います。

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

---

2. システムごとに、**ipseckey** コマンドモードで次のコマンドを入力して、現在のセキュリティアソシエーションをフラッシュします。

```
# ipseckey

> flush
>
```

3. 次のコマンドを入力して、出力パケットに新しいセキュリティアソシエーションを設定します。

```
> add esp spi new-random-number src local-system dst remote-system \
auth_alg the_algorithm-name encr_alg the_algorithm-name \
authkey random-hex-string-of-algorithm-specified-length \
encrkey random-hex-string-of-algorithm-specified-length
```

4. **Return** キーを押します。

コマンドが実行され、ipseckey コマンドモードプロンプトが再表示されます。

5. 次のコマンドを入力して、入力パケットに新しいセキュリティアソシエーションを設定します。

```
> add esp spi new-random-number src remote-system dst local-system \
auth_alg the_algorithm-name encr_alg the_algorithm-name \
authkey random-hex-string-of-algorithm-specified-length \
encrkey random-hex-string-of-algorithm-specified-length
```

注 – キーと SPI は、セキュリティアソシエーションごとに変更できますが、同じにしてはいけません。

6. **Ctrl-D** を押すか、 **quit** を入力してこのモードを終了します。

例 — ipseckeys ファイルのセキュリティアソシエーションの変更

次の例では、partym と enigma のシステムでキーを更新します。そのトラフィックは374 ページの「2 つのシステム間のトラフィックを保護」でセキュリティ保護されています。 両方のシステムでは、AH に SHA1 アルゴリズムを使用し、IPv6 アドレスを使用しているものとします。

1. 現在のキーをフラッシュします。
2. 両方のシステムで ipseckeys ファイルを編集して、既存の SPI と authkey の値を変更します。

   a. 次のように、partym で ipseckeys ファイルを編集します。

```
# for inbound packets
add ah spi 0x55142 dst partym authalg sha1 \
        authkey 012345678921001234abcdeffedcba9876543210
# for outbound packets
add ah spi 0x235211 dst enigma authalg sha1 \
        authkey 21001234abcdef9876543210012345678 9fedcba
```

   b. 次のように、enigma で ipseckeys ファイルを編集します。

```
# for inbound packets
add ah spi 0x235235 dst enigma authalg sha1 \
        authkey 123456780123456789abcdeffedcba9876543210
# for outbound packets
add ah spi 0x123456 dst partym authalg sha1 \
        authkey abcdef9876543210012345678 9fed12345678bac
```

3. そのラッチされたソケットが新しいキーを使用していることを確認するには、両方のシステムをリブートします。 ブート時に ipseckeys ファイルが自動的に読み込まれます。

```
# /usr/sbin/reboot
```

テストする場合には、各システムをリブートしないで新しいキーをセキュリティデータベースに配置できます。

```
# ipseckey -f /etc/inet/secret/ipseckeys
```

第 21 章

インターネットキー交換

IP データグラムのセキュリティ保護された伝送に必要な IPsec SA (セキュリティアソシエーション) のキー情報の管理をキー管理といいます。 自動キー管理では、キーの作成、認証、および交換に通信のセキュリティ保護されたチャネルを要求します。Solaris オペレーティング環境では、インターネットキー交換 (IKE) を使用してキー管理を自動化します。IKE を使用すれば、セキュリティ保護されたチャネルを大量のトラフィックに割り当てるために容易にスケーリングできます。IPv4 パケットの IPsec SAでは、IKE の利点を生かすことができます。

この章では、以下の内容について説明します。

IKE の概要

インターネットキー交換 (IKE) デーモン `in.iked(1M)` では、保護された方法でセキュリティアソシエーションのキー情報のネゴシエーションと認証を行います。また、SunOS™ によって提供される内部機能からキーのランダムシードを使用します。IKE は、PFS (Perfect Forward Secrecy) をサポートしています。つまり、データ伝送を保護するキーを使用しないで追加キーを取得し、データ伝送のキーの作成に使用するシードを再利用しません。

IKE デーモンでリモートホストの公開暗号鍵が検出されると、ローカルシステムでは暗号鍵が検出されたリモートホスト宛てのメッセージを暗号化できます。IKE デーモンでは、そのジョブを交換と呼ばれる 2 つのフェーズで実行します。

フェーズ 1 交換

フェーズ 1 交換はメインモードといいます。フェーズ 1 交換では、IKE は公開鍵暗号方式を使用して、ピア IKE エンティティによる IKE 自体を認証します。その結果が ISAKMP (Internet Security Association and Key Management Protocol) セキュリティアソシエーションで、IKE で IP データグラムのキー情報のネゴシエーションを行うためのセキュリティ保護されたチャネルとなります。IPsec SA とは異なり、ISAKMP セキュリティアソシエーションは双方向であるため、1 つだけ必要です。

IKE でキー情報のネゴシエーションを行う方法は、フェーズ 1 交換で設定可能です。IKE では、`/etc/inet/ike/config` ファイルから設定情報を読み取ります。設定情報には、影響するインタフェース、使用するアルゴリズム、認証方式、および PFS 使用の有無が含まれています。認証方式には、事前共有鍵と公開鍵証明書の 2 つがあります。 公開鍵証明書は自己署名付きにするか、PKI (Public Key Infrastructure) ベンダーから 認証局 (CA) によって発行できます。 ベンダーには、iPlanet™ Certificate Management System、Entrust、および Verisign があります。

フェーズ 2 交換

フェーズ 2 交換はクイックモードといいます。フェーズ 2 交換では、IKE は IKE デーモンを実行するホスト間の IPsec SA を作成および管理します。また、フェーズ 1 で作成したセキュリティ保護されたチャネルを使用して、キー情報の伝送を保護します。IKE デーモンでは、乱数発生関数 (`/dev/random`) によってキーを作成してキーを一定の割合で更新し (構成可能)、キー情報を IPsec ポリシー構成ファイルで指定したアルゴリズムに提供します。

IKE のネゴシエーション

2 つの IKE デーモンがある場合、相互認証を行うには、有効な IKE 構成ポリシーファイル `ike.config(4)` とキー情報が必要です。 ポリシーファイルには、フェーズ 1 交換の認証に事前共有鍵または公開鍵証明書を使用するかどうかを決定する IKE ポリシーエントリが含まれています。

鍵のペア `auth_method preshared` は、事前共有鍵が使用されることを示します。`auth_method` の値が `preshared` 以外の場合には、公開鍵証明書が使用されることを示します。公開鍵証明書は自己署名付きにするか、PKI ベンダーから発行できます。

事前共有鍵の使用

事前共有鍵は、1 つのシステムの管理者によって作成され、通信するシステムの管理者とアウトオブバンドで共有します。管理者は、大量のランダム鍵の作成、そのファイルとアウトオブバンド伝送の保護に十分注意する必要があります。鍵は、各システムの `/etc/inet/secret/ike.preshared` ファイルに保存されます。IPsec の場合は `ipseckeys` ファイルですが、IKE の場合は `ike.preshared(4)` ファイルとなります。`ike.preshared` ファイルにある鍵に問題があると、その鍵から導出されるすべての鍵に問題が発生します。

1 つのシステムの事前共有鍵は、通信するシステムの鍵と同一にする必要があります。鍵は特定の IP アドレスに連結され、そのセキュリティ保護は管理者が通信するシステムを制御する場合に最も強化されます。

公開鍵証明書の使用

公開鍵証明書を使用すると、通信するシステムが秘密鍵情報をアウトオブバンドで共有する必要がなくなります。公開鍵では、鍵の認証とネゴシエーションに Diffie-Helman 方式を採用します。公開鍵証明書には、自己署名付きまたは認証局 (CA) による認証の 2 つの方法があります。

自己署名付き公開鍵証明書は、管理者によって作成されます。`ikecert local -ks` コマンドを実行して、システムの公開鍵と非公開鍵のペアの非公開部分を作成します。その後、管理者は通信するシステムから X.509 形式で自己署名付き証明書の出力を取得します。通信するシステムの証明書は、鍵のペアの公開部分の `ikecert certdb` コマンドに入力されます。自己署名付き証明書は、通信するホストの `/etc/inet/ike/publickeys` ディレクトリに保存されます。

自己署名付き証明書は、事前共有鍵と CA 間の中間ポイントになります。事前共有鍵とは異なり、自己署名付き証明書は移動体システムまたは再番号付け可能なシステムで使用できます。これを可能にするには、管理者は DNS (`www.example.org`) または EMAIL (`root@domain.org`) の代替名を使用します。

公開鍵は、PKI または CA ベンダーで配信できます。 公開鍵とそれに関連する CA は、管理者によって `/etc/inet/ike/publickeys` ディレクトリに格納されます。また、ベンダーは証明書無効リスト (CRL) も発行します。管理者は鍵と CA を格納するだけでなく、CRL を `/etc/inet/ike/crls` ディレクトリに格納する責任があります。

CA にはサイトの管理者ではなく、外部のベンダーによって認証されるといった特長があります。その点では、CA は公証された証明書となります。自己署名付き証明書と同様に、CA は移動体システムまたは再番号付け可能なシステムで使用できます。その一方、自己署名付き証明書とは異なり、CA は通信する多くのシステムを保護するために容易にスケーリングします。

IKE ユーティリティおよび IKE ファイル

この節では、IKE 構成ファイルと IKE を実装するさまざまなコマンドについて説明します。IPv4 ネットワークに IKE を実装する方法の手順については、393 ページの「IKE の実装 (作業マップ)」を参照してください。

表 **21–1** IKE ファイルおよび IKE コマンドのリスト

ファイルまたはコマンド	説明
`in.iked(1M)` デーモン	インターネットキー交換 (IKE) デーモン。自動キー管理を有効にします
`ikeadm(1M)`	IKE 管理コマンド。IKE ポリシーの表示および変更に使用します
`ikecert(1M)`	認証データベース管理コマンド。ローカル公開鍵の認証データベースの操作に使用します
`/etc/inet/ike/config` ファイル	IKE ポリシー構成ファイル。インバウンド IKE 要求のマッチングとアウトバウンド IKE 要求の準備に関するサイトの規則が含まれています。このファイルがある場合には、`in.iked` デーモンがブート時に自動的に開始します
`/etc/inet/secret/ike.preshared` ファイル	事前共有鍵のファイル。フェーズ 1 認証の秘密鍵情報が含まれています
`/etc/inet/secret/ike.privatekeys` ファイル	非公開鍵のディレクトリ。公開鍵と非公開鍵のペアの非公開部分が含まれています
`/etc/inet/ike/publickeys` ディレクトリ	公開鍵と証明書ファイルを保存するディレクトリ。デフォルトでは、Sun 証明書が含まれます公開鍵と非公開鍵のペアの公開部分が含まれています
`/etc/inet/ike/crls` ディレクトリ	公開鍵と証明書ファイルの無効リストを保存するディレクトリ

IKE デーモン

`in.iked(1M)` デーモンを実行すると、Solaris ホスト上の暗号キーの管理が自動化されます。また、同じプロトコルを実行するリモートホストとのネゴシエーションを行い、認証されたキー情報が、保護された方法でセキュリティアソシエーションに提供されます。そのデーモンは、セキュリティ保護された通信を行うすべてのホストで実行する必要があります。IKE 構成ポリシーファイル `/etc/inet/ike/config` がある場合には、IKE デーモンがブート時に自動的にロードされます。

IKE デーモンを実行すると、システムではそのピア IKE エンティティに対してそのシステム自体を認証します (フェーズ 1)。そのピアは、認証方式として IKE ポリシーファイルに定義されています。その後、セッションのキーが設定されます (フェーズ 2)。 ポリシーファイルで指定した時間間隔で、IKE キーが自動的に更新されます。 `in.iked` デーモンを実行すると、ネットワークからの着信 IKE 要求と `PF_KEY` ソケット経由の出力トラフィックの要求を待機します。詳細については、`pf_key(7P)` マニュアルページを参照してください。

2 つのプログラムで IKE デーモンをサポートします。`ikeadm(1M)` コマンドを実行すると、管理者は IKE ポリシーを表示および変更できます。 `ikecert(1M)` コマンドを実行すると、管理者は公開鍵データベース `ike.privatekeys` と `publickeys` を表示および管理できます。

IKE ポリシーファイル

IKE 構成ポリシーファイル `/etc/inet/ike/config` により、IKE デーモン自体のキー情報、およびそのデーモンが管理する IPsec SAのキー情報が提供されます。IKE デーモン自体は、フェーズ 1 交換でキー情報を要求します。`ike/config` ファイルにある規則に基づいてキー情報が設定されます。ポリシーファイルにある有効な規則にはラベルが含まれています。その規則により、キー情報を使用するホストまたはネットワークが特定され、認証方式が指定されます。有効なポリシーファイルの例については、393 ページの「IKE 作業」を参照してください。そのパラメータの例と説明については、`ike.config(4)` のマニュアルページを参照してください。

IPsec SA は、IPsec 構成ポリシーファイル `/etc/inet/ipsecinit.conf` で設定されるポリシーに従って保護される IP データグラムで使用されます。 IKE ポリシーファイルにより、IPsec SA の作成時に PFS を使用するかどうかが決定されます。

`ike/config` ファイルのセキュリティについては、`ipsecinit.conf` ファイルのセキュリティと同様です。詳細については、367 ページの「セキュリティについて」を参照してください。

IKE 管理コマンド

`ikeadm` コマンドを実行すると、IKE 構成ファイルの構文チェック、IKE デーモンプロセスの要素の表示、および IKE デーモンに渡すパラメータの変更を行うことができます。また、統計情報の収集、IKE プロセスのデバッグを行うこともできます。それらのオプションの例と詳細については、`ikeadm(1M)` のマニュアルページを参照してください。実行する IKE デーモンの権限レベルにより、表示および変更可能な IKE デーモンの要素が決まります。権限レベルは 3 つあります。

- 0x0 (基本レベル) — 権限の基本レベルでは、キー情報を表示または変更できません。 基本レベルは、`in.iked` デーモン実行時のデフォルトレベルになります。
- 0x1 (modkeys レベル) — 権限の modkeys レベルでは、事前共有鍵を削除、変更、または追加できます。

- 0x2 (keymat レベル) — 権限の keymat レベルでは、ikeadm コマンドを指定して実際のキー情報を表示できます。

ikeadm コマンドのセキュリティについては、ipseckey コマンドのセキュリティと同様です。 詳細については、369 ページの「セキュリティについて」を参照してください。

事前共有鍵ファイル

/etc/inet/secret/ ディレクトリには、ISAKMP SA と IPsec SA の事前共有鍵が格納されています。管理者が共有鍵を手動で作成すると、ike.preshared ファイルには ISAKMP SA の事前共有鍵、ipseckeys ファイルには IPsec SA の事前共有鍵が格納されます。 secret ディレクトリは 0700 で、その中にあるファイルは 0600 で保護されています。

- ike.config ファイルが事前共有鍵を要求したときに、管理者は ike.preshared ファイルを作成します。そのファイルには、ISAKMP SA (つまり、IKE 認証) のキー情報が含まれています。IKE では、フェーズ 1 交換の認証に事前共有鍵を使用するため、 in.iked デーモンの開始前に ike.preshared ファイルを有効にする必要があります。
- ipseckeys ファイルには、IPsec SA のキー情報が含まれています。IPv6 ホストの場合、管理者はそのファイルにあるキーを手動で作成および更新します。そのファイルを手動で管理する例については、374 ページの「IPsec 作業」を参照してください。IKE デーモンでは、このファイルを使用しません。IKE によって IPsec SA に対して生成されるキー情報は、カーネルに保存されます。

IKE 公開鍵のデータベースおよびコマンド

ikecert(1M) コマンドを実行して、ローカルホストの公開鍵データベースを操作します。IKE では、ike.config ファイルが公開鍵証明書を要求するときに、それらのデータベースを使用してフェーズ 1 交換を認証するため、in.iked デーモンを起動する前にそれらのデータベースを格納したディレクトリを生成する必要があります。 3 つのサブコマンド certlocal、certdb、certrldb をそれぞれ実行して、3 つのデータベースを処理します。

ikecert certlocal コマンド

certlocal サブコマンドを実行して、/etc/inet/secret/ike.privatekeys ディレクトリにある非公開鍵データベースを管理します。このサブコマンドを選択すると、非公開鍵の追加、表示、および削除を行うことができます。また、自己署名付き証明書または証明書要求のいずれかを作成できます。-ks オプションを選択すると、自己署名付き証明書が作成され、-kc オプションを選択すると、証明書要求が作成されます。

非公開鍵を作成する場合、certlocal サブコマンドに渡すパラメータは、次の表に示すように、ike.config ファイルに反映する必要があります。

表 **21–2** ike certlocal の値と ike.config の値の対応表

certlocal オプション	**ike.config** エントリ	注
-A 対象の代替名	cert_trust 対象代替名	証明書を一意に識別するニックネーム。指定可能な値は IP アドレス、電子メールアドレス、およびドメイン名です。
-D *X.509* 識別名	cert_root *X.509* 識別名	国、組織名、組織単位、共通名を含む認証局のフルネーム。
-t dsa-sha1	auth_method dss_sig	RSA よりもわずかに遅くなります。 特許は登録されていません。
-t rsa-md5 -t rsa-sha1	auth_method rsa_sig	DSA よりもわずかに速くなります。 特許の期限切れは 2000 年 9 月です。 RSA 公開鍵は、最大ペイロードを暗号化できるようにその長さを十分長くする必要があります。一般的に識別名などの ID ペイロードが最大になります。
-t rsa-md5 -t rsa-sha1	auth_method rsa_encrypt	RSA 暗号化により、IKE にある ID が不正侵入者から保護されますが、IKE ピアには互いの公開鍵の認識が要求されます。

ikecert certlocal -kc コマンドを指定して証明書要求を実行する場合、そのコマンドの出力をベンダーに送信します。その後、ベンダーがキー情報を作成します。certdb と certrldb のサブコマンドへの入力としてベンダーのキー情報を使用します。

ikecert certdb コマンド

certdb サブコマンドを実行して、公開鍵データベース /etc/inet/ike/publickeys を管理します。そのサブコマンドを選択すると、公開鍵と証明書を追加、表示、および削除できます。また、通信するシステムで ikecert certlocal -ks コマンドを実行して作成された証明書を入力として受け入れます。手順については、399 ページの「自己署名付き公開証明書による IKE の設定方法」を参照してください。さらに、PKI または CA から受信する証明書も入力として受け入れます。手順については、401 ページの「認証局による署名付き公開鍵による IKE の設定方法」を参照してください。

ikecert certrldb コマンド

certrldb サブコマンドを実行して、証明書無効リスト (CRL; Certificate Revocation List) データベース /etc/inet/ike/crls を管理します。crls データベースには、公開鍵の無効リストが保存されています。よって、このリストには、すでに有効でな

い証明書が明記されます。PKI によって CRL が提供されるときに、ikecert certrldb コマンドを指定して CRL データベースにそれらの CRL を格納します。手順については、403 ページの「証明書無効リストを更新する方法」を参照してください。

/etc/inet/ike/publickeys ディレクトリ

/etc/inet/ike/publickeys ディレクトリには、公開鍵と非公開鍵のペアの公開部分とファイルにあるその証明書、つまり「スロット」が格納されています。/etc/inet/ike ディレクトリは 0755 で保護されます。そのディレクトリに格納される公開鍵データベースは、 各国で読み取り可能です (0644)。 ikecert certdb コマンドを使用して、そのディレクトリを読み込みます。

そのファイルには、別のシステムで生成された証明書の X.509 識別名がコード化形式で含まれています。自己署名付き証明書を使用する場合、そのコマンドへの入力として、通信するシステムの管理者から受信する証明書を使用します。PKI からの証明書を使用する場合、ベンダーからこのデータベースに 2 つのキー情報 (ベンダーに送信した情報に基づいた証明書、およびベンダーからの CA) を格納します。

iPlanet CMS の評価コピー PKI は、インストールパッケージの Media Kit で使用できます。

/etc/inet/secret/ike.privatekeys ディレクトリ

ike.privatekeys ディレクトリには、公開鍵と非公開鍵のペアの一部である非公開鍵ファイル、ISAKMP SA のキー情報が格納されています。このディレクトリは 0700 で保護されています。このデータベースにある非公開鍵は、publickeys データベースの公開鍵とペアにする必要があります。ikecert certlocal コマンドを実行して、コマンドのディレクトリを読み込みます。非公開鍵は、ペアとなる公開鍵、自己署名付き証明書や CA が /etc/inet/ike/publickeys ディレクトリに格納されてから有効になります。

/etc/inet/ike/crls ディレクトリ

/etc/inet/ike/crls ディレクトリには、証明書無効リスト (CRL) ファイルが含まれています。各ファイルは、/etc/inet/ike/publickeys/ ディレクトリにある公開鍵証明書ファイルに対応しています。PKI ベンダーにより、それらの証明書の CRL が提供されます。ikecert certrldb コマンドを使用して、そのデータベースを読み込みます。

IKE の実装 (作業マップ)

ikeadm(1M)、ikecert(1M) と ike.config(4) のマニュアルページには、個別に例に応じた説明があります。

表 **21–3** IKE の実装 (作業マップ)

タスク	説明	参照先
事前共有鍵による IKE の設定	有効な IKE ポリシーファイルと ike.preshared ファイルを作成します。 また、システムをブートして IKE によって生成された鍵を使用する前に、IPsec ファイルも設定します。	394 ページの「事前共有鍵による IKE の設定方法」
実行中の IKE システムでの事前共有鍵の更新	IKE 権限レベルをチェックし、通信するシステムの最新キー情報に応じて ipseckeys ファイルを編集します。	396 ページの「既存の事前共有鍵を更新する方法」
実行中の IKE システムへの事前共有鍵の追加	IKE 権限レベルをチェックし、通信するシステムの最新キー情報に応じて ikeadm コマンドを実行します。	397 ページの「新しい事前共有鍵を追加する方法」
自己署名付き公開鍵証明書による IKE の設定	ikecert certlocal -ks コマンドを指定して自己署名付き証明書を作成し、ikecert certdb コマンドを指定して通信するシステムからの公開鍵を追加します。	399 ページの「自己署名付き公開証明書による IKE の設定方法」
PKI 認証局による IKE の設定	ikecert certlocal -kc コマンドから PKI に出力を送信し、ベンダーから公開鍵、CA、CRL を格納します。	401 ページの「認証局による署名付き公開鍵による IKE の設定方法」
CA 無効リストの更新	ikecert certrldb コマンドを指定して PKI ベンダーの CRL を格納します。	403 ページの「証明書無効リストを更新する方法」

IKE 作業

この節では、IPv4 アドレスを使用する 2 つのシステム間でトラフィックを保護する鍵を自動的に管理する手順について説明します。IKE 実装では、鍵の長さが異なるさまざまなアルゴリズムが提供されます。鍵の長さは、サイトのセキュリティに応じて選択します。一般的に、鍵の長さが長いほど、セキュリティが高くなります。

▼ 事前共有鍵による IKE の設定方法

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

---

2. 正常に実行するために、システムごとに、グローバルパラメータと **ipsecinit.conf** の **IPsec** ポリシーを有効にする規則を指定して **/etc/inet/ike/config**ファイルを作成します。たとえば、次のように指定します。

```
### ike/config file on enigma, 192.168.66.1

## Global parameters
#
## Phase 1 transform defaults
p1_lifetime_secs 14400
p1_nonce_len 40
#
## Defaults that individual rules can override.
p1_xform { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2
#
## The rule to communicate with partym

{ label "Enigma-Partym"
  localid 192.168.66.1
  remoteid 192.168.55.2
  p1_xform
      { auth_method preshared  oakley_group 5  auth_alg md5
 encr_alg des }
  p2_pfs 5
    }

### ike/config file on partym, 192.168.55.2
## Global Parameters
#
p1_lifetime_secs 14400
p1_nonce_len 40
#
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2

## The rule to communicate with enigma

{ label "Partym-Enigma"
  localid 192.168.55.2
  remoteid 192.168.66.1
```

```
  p1_xform
    { auth_method preshared  oakley_group 5  auth_alg md5  encr_alg des }
  p2_pfs 5
}
```

注 – システム名は一例として使用しているだけです。システム間でトラフィックを保護する場合には、各自のシステムの名前とアドレスを使用してください。

---

3. システムごとに、次のように指定してファイルが有効であるかどうかをチェックします。

```
# /usr/lib/inet/in.iked -c -f /etc/inet/ike/config
```

4. ランダム鍵を生成します。

   Solaris システムでは、od コマンドを使用できます。たとえば、次のように指定します。

```
# od -x </dev/random | head -4
0000000 df97 6d2f 4ef5 2c28 02d5 02aa f9de 481d
0000020 2ae8 b949 67e6 b9b0 dd16 e6d4 b7ea 7278
0000040 ac07 7cc6 99c1 7055 848a 3cf3 4377 980a
0000060 5ad7 5b40 b428 9f3a da20 7daa 65a4 83fe
```

5. システムごとに**/etc/inet/secret/ike.preshared** ファイルを作成し、各ファイルに事前共有鍵を書き込みます。

   この例 (手順 2 を参照) では、暗号化アルゴリズムは DES であるため、事前共有鍵は少なくとも 64 ビットにする必要があります。 鍵の長さが長いほど、セキュリティが高くなります。たとえば、次のように指定します。

```
# ike.preshared on enigma, 192.168.66.1
{ localidtype IP
      localid 192.168.66.1
      remoteidtype IP
      remoteid 192.168.55.2
      # enigma and partym's shared key in hex (128 bits)
      key ac077cc699c17055848a3cf34377980a
    }

# ike.preshared on partym, 192.168.55.2
{ localidtype IP
      localid 192.168.55.2
      remoteidtype IP
      remoteid 192.168.66.1
      # partym and enigma's shared key in hex (128 bits)
      key ac077cc699c17055848a3cf34377980a
    }
```

注 – 事前共有鍵は同一にする必要があります。

---

6. システムごとに、他のシステムのアドレスとホスト名を **/etc/hosts** ファイルに追加します。たとえば、次のように指定します。

   partym という名前のシステムでは、次のように指定します。

   ```
   # Secure communication with enigma
   192.168.66.1 enigma
   ```

   enigma という名前のシステムでは、次のように指定します。

   ```
   # Secure communication with partym
   192.168.55.2  partym
   ```

7. 各システムごとに、次の行を追加して **/etc/inet/ipsecinit.conf** ファイルを編集します。

   enigma システムでは、次のように指定します。

   ```
   {laddr enigma raddr partym} ipsec {auth_algs any sa shared}
   ```

   partym システムでは、次のように指定します。

   ```
   {laddr partym raddr enigma} ipsec {auth_algs any sa shared}
   ```

8. 各システムをリブートすることで、セキュリティ保護された通信を可能にします。

   ```
   # /usr/sbin/reboot
   ```

▼ 既存の事前共有鍵を更新する方法

この手順では、既存の事前共有鍵を変更するものとします。3DES、AES、Blowfish などの強力な暗号化アルゴリズムを使用すると、両方のシステムのリブート時に備えて、鍵を変更するスケジュールを作成できる場合があります。この手順は、トラフィックの保護に DES などのアルゴリズムを使用するシステムに適用されます。

1. システムコンソールからスーパーユーザーになります。

   注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

   ---

2. ランダム鍵を生成してそれらのいずれか **1** つを選択します。

   Solaris システムでは、od コマンドを使用できます。

   ```
   # od -x </dev/random | head -2
   0000000 305e c563 69ca 62c2 ae80 4690 c571 3e18
   0000020 be43 9533 d50f ec49 c7fe cf3c 8f13 91c0
   ```

3. システムごとに **/etc/inet/secret/ike.preshared** ファイルを編集して、現在の鍵を新しい鍵に変更します。

たとえば、enigma と partym のホストでは、key の値を be439533d50fec49c7fecf3c8f1391c0 のような新しい番号に変更します。

4. **in.iked** デーモンがキー情報の変更を許可するかどうか確認します。

```
# /usr/sbin/ikeadm get priv
Current privilege level is 0x2, access to keying material enabled
```

コマンドから 0x1 または 0x2 の権限レベルが戻された場合には、キー情報を変更できます。レベル 0x0 の場合には、キー情報を操作できません。デフォルトでは、in.iked デーモンは 0x0 の権限レベルで実行されます。

5. **in.iked** デーモンを実行してキー情報を変更できるようにするには、**ike.preshared** ファイルの新しいバージョンを読み取ります。

たとえば、次のように指定します。

```
# ikeadm read preshared
```

6. **in.iked** デーモンを実行してキー情報を変更できないようにするには、そのデーモンを消去してから再起動します。

そのデーモンを開始すると、ike.preshared ファイルの新しいバージョンを読み取ります。

たとえば、次のように指定します。

```
# pkill in.iked
# /usr/lib/inet/in.iked
```

▼ 新しい事前共有鍵を追加する方法

in.iked デーモンを実行するシステムでは、そのデーモンを呼び出した後に ipsecinit.conf ファイルに追加したインタフェースに対する事前共有鍵を追加できます。この手順では、両方のシステムの /etc/hosts ファイルと /etc/inet/ipsecinit.conf ファイルに新しいインタフェースをすでに追加し、各システムに ipsecinit.conf ファイルをまだ読み込んでいないものとします。

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

---

2. **in.iked** デーモンがキー情報の変更を許可するかどうか確認します。

```
# /usr/sbin/ikeadm get priv
Current privilege level is 0x2, access to keying material enabled
```

コマンドから 0x1 または 0x2 の権限レベルが戻された場合には、キー情報を変更できます。レベル 0x0 の場合には、キー情報を操作できません。デフォルトでは、

in.iked デーモンは 0x0 の権限レベルで実行されます。

3. **in.iked** デーモンを実行してキー情報を変更できないようにするには、そのデーモンを消去してから正確な権限レベルで開始します。

   たとえば、次のように指定します。

```
# pkill in.iked
# /usr/lib/inet/in.iked -p 2
Setting priv/usr/lib/inet/in.iked -pilege level to 2!
```

4. ランダム鍵を生成してそれらのいずれか **1** つを選択します。

   Solaris システムでは、od コマンドを使用できます。

```
# od -x </dev/random | head -2
0000000 2d86 b6f6 eb7a e8a9 3d83 58b2 cd17 4164
0000020 8be4 fea4 b456 933a 46dd 149a 0a10 b2e4
```

5. 各システムで **ikeadm** コマンドを入力して、新しいキー情報を追加します。

   たとえば、enigma システムではホスト nemesis 192.163.55.8 のキーを次のように追加します。

```
# ikeadm
ikeadm> add preshared { localidtype ip localid 192.168.66.1
remoteidtype ip remoteid 192.163.55.8 ike_mode main
key 2d86b6f6eb7ae8a93d8358b2cd174164 }
ikeadm: Successfully created new preshared key.
```

   ホスト nemesis では、管理者は次のように同一の鍵を追加します。

```
# ikeadm
ikeadm> add preshared { localidtype ip localid 192.163.55.8
remoteidtype ip remoteid 192.168.66.1 ike_mode main
key 2d86b6f6eb7ae8a93d8358b2cd174164 }
ikeadm: Successfully created new preshared key.
```

注 – Error: invalid preshared key definition というメッセージは、add preshared コマンドに入力ミスがあったか、パラメータが省略されたことを示しています。コマンドを正確に再入力して鍵を追加してください。

---

6. **ikeadm** コマンドモードを終了します。

```
ikeadm> exit
#
```

7. システムごとに、**in.iked** デーモンの権限レベルを低くします。

```
# ikeadm set priv base
```

8. システムごとに、**ipsecinit.conf** ファイルを有効にして、追加したインタフェースを保護します。

```
# ipsecconf -a /etc/inet/ipsecinit.conf
```

注 – このコマンドの実行時には警告を読んでください。ソケットがすでに使用中 (ラッチされた) の場合には、システムへの背面ドアが保護されません。

▼ 自己署名付き公開証明書による IKE の設定方法

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

2. **ikecert certlocal -ks** コマンドを使用して、自己署名付き証明書を **ike.privatekeys** データベースに追加します。たとえば、次のように指定します。

```
# ikecert certlocal -ks -m 1024 -t rsa-md5 \
-D "C=US, O=ExampleCompany, OU=US-Example, CN=Example" \
-A IP=192.168.10.242
Generating, please wait...
Certificate:
Certificate generated.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIICLTCCAZagAwIBAgIBATANBgkqhkiG9w0BAQQFADBNMQswCQYDVQQGEwJVUzEX
...
6sKTxpg4GP3GkQGcd0r1rhW/3yaWBkDwOdFCqEUyffzU
-----END X509 CERTIFICATE-----
```

3. その証明書を、通信するシステムの管理者に送信します。
   その証明書は、次のようにして電子メールにカット&ペーストできます。

```
To: root@us.example.com
From: root@un.example.com
Message: -----BEGIN X509 CERTIFICATE-----
MIICLTCCAZagAwIBAgIBATANBgkqhkiG9w0BAQQFADBNMQswCQYDVQQGEwJVUzEX
...
6sKTxpg4GP3GkQGcd0r1rhW/3yaWBkDwOdFCqEUyffzU
-----END X509 CERTIFICATE-----
```

4. **/etc/inet/ike/config** ファイルを編集して、通信するシステムからの公開鍵を認識します。 たとえば、次のように指定します。

```
# Explicitly trust the following self-signed certs
# Use the Subject Alternate Name to identify the cert

cert_trust "192.168.10.242"
```

```
cert_trust "192.168.11.241"

## Parameters that may also show up in rules.

p1_xform { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 5

{
 label "UN-Example to US-Example"
 local_id_type dn
 local_id "C=US, O=ExampleCompany, OU=UN-Example, CN=Example"
 remote_id_type dn
 remote_id "C=US, O=ExampleCompany, OU=US-Example, CN=Example"

 local_addr 192.168.10.242
 remote_addr 192.168.11.241

 p1_xform
  { auth_method rsa_encrypt oakley_group 2  auth_alg md5  encr_alg des }
}
```

5. 次の手順を実行して、通信するシステムの公開鍵を追加します。

   a. 管理者の電子メールから公開鍵をコピーします。

   b. 次のように **ikecert certdb -a** コマンドと **<Return>** を入力します。

      ```
      # ikecert certdb -a <Return>
      ```

   c. 次のように公開鍵をペーストして **<Return>** と入力します。

      ```
      -----BEGIN X509 CERTIFICATE-----
      MIICL...
      ...
      KgDid/nxWPlWQU5vMAiwJXfa0sw/A12w448JVkVmEWaf
      -----END X509 CERTIFICATE----- <Return>
      ```

   d. **<Control-D>** を入力して入力を終了します。

      ```
      <Control-D>
      ```

6. 通信するシステムの管理者と一緒にキーが改ざんされていないことを確認します。
   たとえば、その管理者に電話で連絡して以下に示す公開鍵ハッシュの値を比較できます。

```
# ikecert certdb -l
        Certificate Slot Name: 0   Type: if-modn
        Subject Name: <C=US, O=ExampleCo, OU=UN-Example, CN=Example>
        Key Size: 1024
        Public key hash: 2239A6A127F88EE0CB40F7C24A65B818

other system # ikecert certlocal -l
Local ID Slot Name: 1   Type: if-modn
        Key Size: 1024
        Public key hash: 2239A6A127F88EE0CB40F7C24A65B818
```

注 – 上記の公開鍵ハッシュは、使用しているシステムで生成される公開鍵ハッシュとは異なります。

▼ 認証局による署名付き公開鍵による IKE の設定方法

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

2. **ikecert certlocal -kc** コマンドを使用して、信頼されたルート証明書を **ike.privatekeys** データベースに追加します。

   たとえば、次のように指定します。

```
# ikecert certlocal -kc -m 1024 -t rsa-md5 \
-D "C=US, O=ExampleCompany\, Inc., OU=US-Example, CN=Example" \
-A "DN=C=US, O=ExampleCompany\, Inc., OU=US-Example"
Generating, please wait...
Certificate request generated.
-----BEGIN CERTIFICATE REQUEST-----
MIIByjCCATMCAQAwUzELMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGVDb21w
...
lcM+tw0ThRrfuJX9t/Qa1R/KxRlMA3zckO80mO9X
-----END CERTIFICATE REQUEST-----
```

3. その要求を外部の認証局または **PKI** に依頼します。

   ベンダーは、各データベースに入力される 2 つの証明書と CRL を発行します。

   - 公開鍵証明書 – この証明書はベンダーに依頼した要求に基づいて作成されます。この証明書によって一意に識別されます。
   - 認証局 – ベンダーの署名です。CA によって公開鍵証明書が正規のものであることが確認されます。
   - 証明書無効リスト – ベンダーが無効にした証明書の最新リストです。

4. **ikecert** コマンドで **3** つの証明書を引数として入力します。

   a. システムコンソールからスーパーユーザーになります。

   b. 次のように **ikecert certdb -a** コマンドと **<Return>** を入力します。

```
# ikecert certdb -a <Return>
```

c. 次のようにベンダーから受信した証明書をペーストして、 **<Return>** と入力します。

```
-----BEGIN X509 CERTIFICATE-----
...
-----END X509 CERTIFICATE-----<Return>
```

d. **<Control-D>** を入力して入力を終了します。

```
<Control-D>
```

e. 次のように **ikecert certdb -a** コマンドと **<Return>** を入力します。

```
# ikecert certdb -a <Return>
```

f. 次のようにベンダーの **CA** をペーストして **<Return>** と入力してから、**<Control-D>** と入力して入力を終了します。

```
-----BEGIN X509 CERTIFICATE-----
...
-----END X509 CERTIFICATE-----<Return>
<Control-D>
```

g. 次のように **ikecert certrldb -a** コマンドと **<Return>** を入力します。

```
# ikecert certrldb -a <Return>
```

h. 次のようにベンダーの **CRL** をペーストして **<Return>** と入力してから、**<Control-D>** と入力して入力を終了します。

5. **/etc/inet/ike/config** ファイルを編集して、ベンダーを認識します。
ベンダーから利用するように通知された名前を使用します。たとえば、次のように指定します。

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

## Parameters that may also show up in rules.

p1_xform { auth_method rsa_sig oakley_group 1 auth_alg sha1 encr_alg 3des }
p2_pfs 2

{
 label "UN-Example to US-Example - Example PKI"
 local_id_type dn
 local_id "C=US, O=ExampleCompany, OU=UN-Example, CN=Example"
 remote_id_type dn
 remote_id "C=US, O=ExampleCompany, OU=US-Example, CN=Example"

 local_addr 192.168.10.242
 remote_addr 192.168.11.241
```

```
 p1_xform
  { auth_method rsa_encrypt oakley_group 2  auth_alg md5  encr_alg des }
}
```

6. 今までと同じ操作を、通信するシステムでも実行します。

   上記の例に従って "C=US, O=ExampleCompany, OU=US-Example, CN=Example" システムで `ikecert` コマンドを実行します。その `ike.config` ファイルでは、ローカルパラメータにはローカル情報、リモートパラメータには使用しているシステムの情報を使用します。

   たとえば、次のように指定します。

```
# Trusted root cert
# This certificate is from Example PKI

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

## Parameters that may also show up in rules.

p1_xform { auth_method rsa_sig oakley_group 1 auth_alg sha1 encr_alg 3des }
p2_pfs 2

{
 label "US-Example to UN-Example - Example PKI"
 local_id_type dn
 local_id "C=US, O=ExampleCompany, OU=US-Example, CN=Example"
 remote_id_type dn
 remote_id "C=US, O=ExampleCompany, OU=UN-Example, CN=Example"

 local_addr 192.168.11.241
 remote_addr 192.168.10.242

 p1_xform
  { auth_method rsa_sig oakley_group 2  auth_alg md5  encr_alg des }
}
```

   `/etc/hosts` ファイルと `/etc/inet/ipsecinit.conf` ファイルを変更して、保護されたインタフェースを組み込み、システムをリブートすると、IKE デーモンを実行して公開鍵と CA による IKE 自体の認証を行います。

注 – RSA 暗号化認証方式により、IKE の ID が不正侵入者から保護されるため、IKE ではピアの証明書を検出しません。したがって、その方式では、IKE ピアが互いの公開鍵を認識することが必要になります。よって、`ike.config` ファイルの `auth_method rsa_encrypt` を使用する場合には、ピアの証明書を公開鍵データベースに追加する必要があります。

▼ 証明書無効リストを更新する方法

1. システムコンソールからスーパーユーザーになります。

注 – リモートログインすると、セキュリティ上重要なトラフィックが盗聴される恐れがあります。何らかの方法でリモートログインを保護していても、システム全体のセキュリティがリモートログインセッションレベルに低下します。

2. 無効となった証明書を抽出する方法については、ベンダーからの指示に従ってください。
3. 次の手順を実行して、無効となった証明書を **CRL** データベースに追加します。
   a. **`ikecert certrldb -a`** コマンドと **<Return>** を入力します。

      ```
      # ikecert certrldb -a <Return>
      ```

   b. **PKI** ベンダーから無効となった証明書をペーストして **<Return>** と入力してから、**<Control-D>** と入力して入力を終了します。
4. 無効リストにある **CRL** ごとにこの手順を繰り返します。

第 22 章

モバイル IP (トピック)

第 23 章

モバイル IP (概要)

モバイルインターネットプロトコル (IP) は、モバイルコンピュータ間での情報の送受信を可能にします。モバイルコンピュータには、ラップトップや無線通信機器などがあります。モバイルコンピュータは外部のネットワークに移動できます。 外部のネットワークに移動しても、モバイルコンピュータは元のネットワークにアクセスし、通信することができます。モバイル IP の Solaris による実装では、IPv4 のみをサポートしています。

この章では、以下の内容について説明します。

概要

インターネットプロトコル (IP) の現在のバージョンでは、コンピュータがインターネットあるいはネットワークに接続する場所は固定されているものと仮定しています。また、IP はその IP アドレスが接続しているネットワークを識別するものと仮定しています。データグラムは、IP アドレスに含まれる場所情報に基づいてコンピュータに送信されます。使用されている多くのインターネットプロトコルは、ノードの IP アドレスを変更しない状態にしておく必要があります。よって、インターネットプロ

トコルのアプリケーションをモバイル IP コンピュータデバイスで実行すると、そのアプリケーションは失敗します。TCP 接続が一時的なものでない場合は、HTTP も失敗します。 IP アドレスの更新と Web ページの更新は、場所を移動して行うことはできません。

モバイルコンピュータ、つまり「モバイルノード」が IP アドレスを変更せずに新たなネットワークに移動すると、そのアドレスは新しい接続点を反映しません。その結果、既存の経路指定プロトコルではデータグラムをモバイルノードに正しく送り届けることができません。このような場合、モバイルノードを新しい場所を表す別の IP アドレスに再構成しなければなりません。ただし、別の IP アドレスを割り当てるには手間がかかります。このように現在のインターネットプロトコルでは、モバイルノードがアドレスを変更せずに移動すれば、その経路を失います。また、アドレスを変更すれば、今までの接続を失ってしまいます。

モバイルIP では、この問題を 2 つの IP アドレスをモバイルノードに与えることで解決します。1 つ目のアドレスは固定の「ホームアドレス」です。2 つ目のアドレスは各接続点で変わる「気付アドレス (care-of address)」です。モバイル IP より、1 つのコンピュータはインターネットを自由に移動することが できます。 また、1 つのコンピュータは同じホームアドレスを維持しながら、企業ネットワークを自由に移動することができます。その結果、ユーザーがコンピュータの接続点を変更した場合でも、コンピュータ動作が中断されることはありません。ネットワークはモバイルノードの新しい場所に関する情報を更新します。モバイル IP に関連する用語の定義については、「用語集」を参照してください。

図 23–1 にモバイル IP の一般的なトポロジを示します。

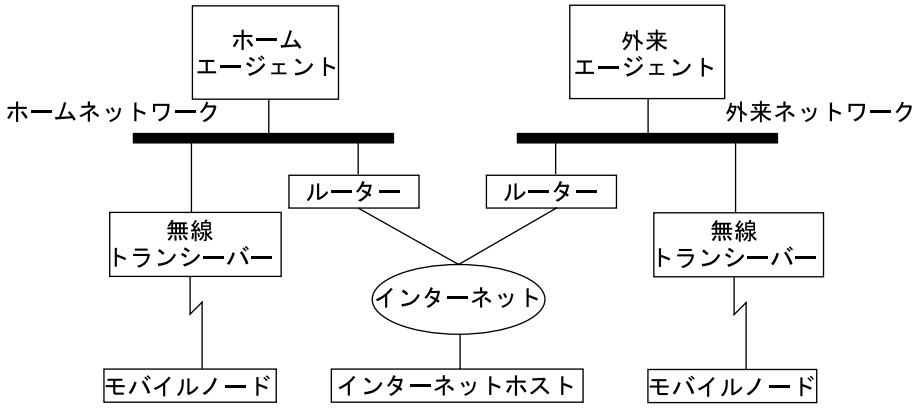


図 **23–1** モバイル IP トポロジ

図 23–1 のモバイル IP トポロジを使って、データグラムがどのようにモバイル IP フレームワーク内のある点から別の点に移動するかを説明します。

1. インターネットホストはモバイルノードのホームアドレスを使って、データグラムをモバイルノードへ送信します (通常の IP 経路指定処理)。

2. モバイルノードがホームネットワーク上にある場合、データグラムは通常の IP 処理でモバイルノードに配信されます。 それ以外の場合は、ホームエージェントがデータグラムを取得します。
3. モバイルノードが外部ネットワーク上にある場合、ホームエージェントがデータグラムを外来エージェントに転送します。 外来エージェントが外部 IP ヘッダーに表示されるように、ホームエージェントでは IP 内 IP の方法でそのデータグラムをカプセル化する必要があります。
4. 外来エージェントはデータグラムをモバイルノードに配信します。
5. モバイルノードからデータグラムは、通常の IP 経路指定手順でインターネットホストへ送信されます。モバイルノードが外部ネットワーク上にある場合は、パケットは外来エージェントに配信されます。外来エージェントはデータグラムをインターネットホストに転送します。
6. 進入フィルタがある場合には、データグラムの送信元であるサブネットに対して、発信元アドレスをトポロジとして正しくしないと、ルーターがデータグラムを転送できません。この状況がモバイルノードと通信ノード間のリンクで発生する場合、外来エージェントで逆方向トンネリングを使用する必要があります。その後、外来エージェントはモバイルノードがそのホームエージェントに送信する各データグラムを配信します。ホームエージェントは、データグラムが通過するパスを経由してそのデータグラムをホームネットワーク上にあるモバイルノードに転送します。この処理により、確実に発信元アドレスは、データグラムが横断する必要のあるすべてのリンクに対して正しくなります。

無線通信の場合、図 23–1 では無線トランシーバを使用してデータグラムをモバイルノードに送信します。また、インターネットホストとモバイルノード間で送受信されるすべてのデータグラムは、モバイルノードのホームアドレスを使用します。モバイルノードが外部ネットワークにある場合でも、ホームアドレスを使用します。その際、気付アドレスはモバイルエージェントとの通信にだけ使用されます。気付アドレスでは、インターネットホストが 関わることはありません。

モバイル IP の構成要素

モバイル IP は次のような新しい構成要素を使用します。

- モバイルノード **(MN)** – モバイルノードの IP ホームアドレスを使用して既存するすべての通信を維持する間、ネットワークに応じて接続点を変更するホストまたはルーター
- ホームエージェント **(HA)** – モバイルノードのホームネットワーク上のルーターまたはサーバー。ルーターは、モバイルノード宛てのデータグラムを取得します。その後、そのデータグラムを気付アドレスに転送します。ホームエージェントは、モバイルノードの現在の場所情報も保持しています。
- 外来エージェント **(FA)** – モバイルノードの移動先である外部ネットワーク上にあるルーターまたはサーバー。そのモバイルノー ドに経路指定サービスを提供します。また、モバイルノードが登録されている間には、モバイルノードに気付アドレ

スも提供します。

モバイル IP の動作

モバイル IP により、IP データグラムをモバイルノードへ経路指定できます。モバイルノードのホームアドレスは、モバイルノードの接続場所に関係なく、常にモバイルノードを指します。ホームから離れているときは、気付アドレスにモバイルノードのホームアドレスを関連付けます。気付アドレスが、モバイルノードの現在の接続点に関する情報を提供します。モバイル IP は、登録機構を利用して気付アドレスをホームエージェントに登録します。

ホームエージェントは、データグラムをホームネットワークからその気付アドレスに転送します。ホームエージェントは、モバイルノードの気付アドレスを含む新しい IP ヘッダーを宛先 IP アドレスとして作成します。この新しいヘッダーは元の IP データグラムをカプセル化します。その結果、モバイルノードのホームアドレスは、カプセル化されたデータグラムが気付アドレスに到達するまで、その経路指定に影響を与えません。このようなカプセル化を「トンネリング」とも呼びます。気付アドレスに到達後、データグラムはカプセル化を解除されます。その後、データグラムはモバイルノードに配信されます。

図 23–2 では、外部ネットワーク B に移動する前の、ホームネットワーク A 上にあるモバイルノードを示しています。どちらのネットワークもモバイル IP をサポートしています。モバイルノードは、モバイルノードのホームアドレス `128.226.3.30` によって常に関連付けられています。

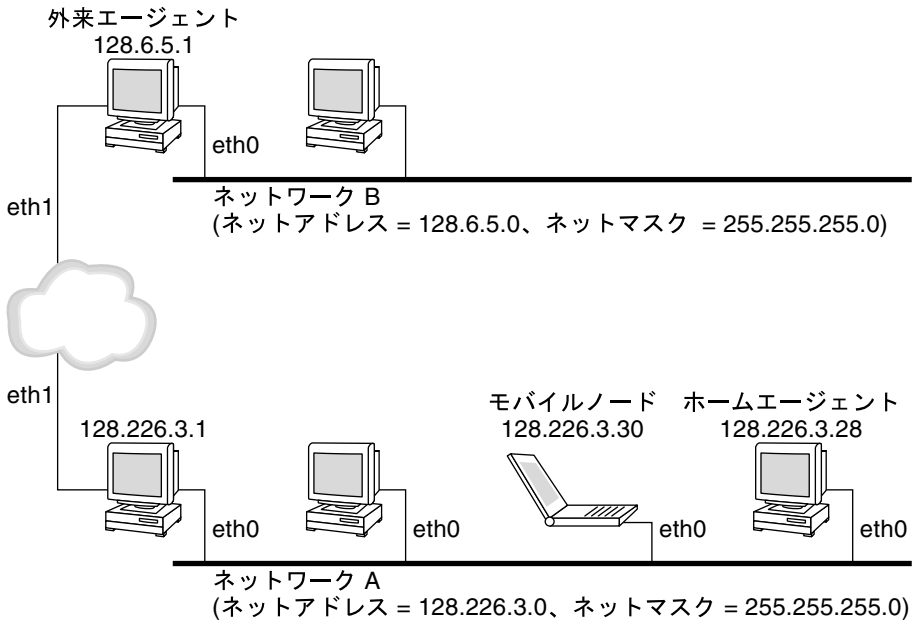


図 **23–2** ホームネットワーク上にあるモバイルノード

次の図では、外部ネットワーク B に移動したモバイルノードを示しています。モバイルノード宛てのデータグラムはホームネットワーク A 上のホームエージェントが取得し、カプセル化します。そのデータグラムをネットワーク B 上の外来エージェントに転送します。カプセル化されたデータグラムを受信すると、外来エージェントは外側のヘッダーを取り除きます。その後、そのデータグラムをネットワーク B にあるモバイルノードに配信します。

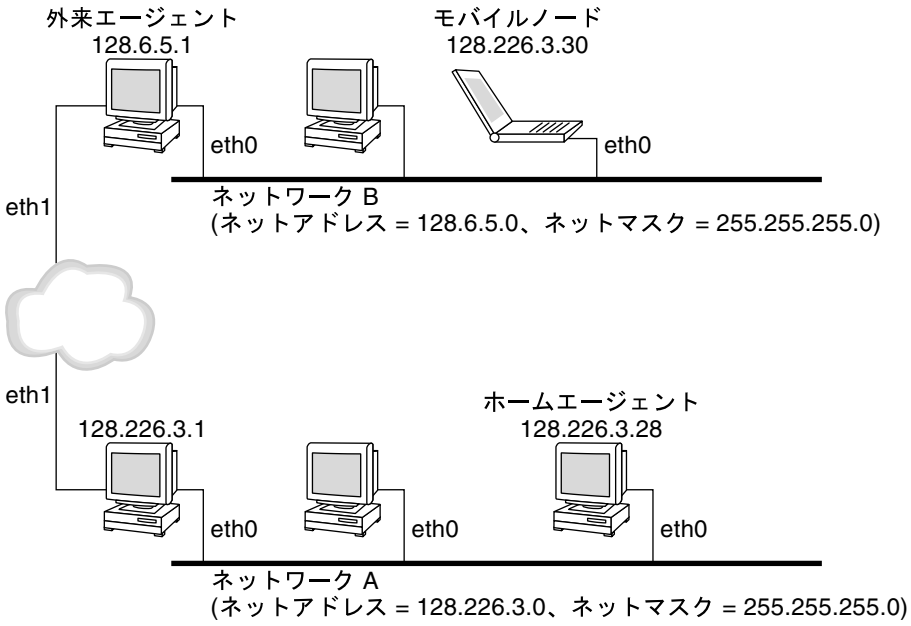


図 **23–3** モバイルノードの外部ネットワークへの移動

気付アドレスは外来エージェントに含まれる場合があります。また、動的ホスト構成プロトコル (DHCP) またはポイントツーポイントプロトコル (PPP) を使ってモバイルノードにより取得される場合もあります。PPP により取得される場合、モバイルノードは、共存気付アドレスを持っています。

モビリティエージェント (ホームエージェントと外来エージェント) は「エージェント通知」メッセージを使用してその存在を通知します。オプションとしてモバイルノードは、エージェント通知メッセージを要請できます。モバイルノードは、「エージェント要請」メッセージによって、ローカルに接続されている任意のモビリティエージェントを使用します。モバイルノードは、そのエージェント通知を受信して、モバイルノードがホームネットワーク上または外部ネットワーク上にあるのかを判断します。

モバイルノードは、特別な「登録」処理を使用して現在の場所に関する情報をホームエージェントに提供します。また、常に存在を通知するモビリティエージェントを「待機」します。さらに、それらの通知を利用して、モバイルノードが別のサブネットに移動する時期を判断します。モバイルノード自体がサブネットに移動したと判断すると、新しい外来エージェントを使用して登録メッセージをホームエージェントに転送します。ある外部ネットワークから別の外部ネットワークにモバイルノードが移動したときにも、モバイルノードでは同じ処理を行います。

モバイルノード自体がホームネットワークにいることを判断すると、モビリティサービスを利用せずに動作します。モバイルノードがホームネットワークに戻ると、ホームエージェントの「登録を解除」します。

エージェントの発見

モバイルノードは次の情報を調べ「エージェントの発見」をします。

- ノードがネットワーク間をいつ移動したか
- ネットワークがホームネットワークまたは外部ネットワークかどうか
- ネットワーク上の各外来エージェントによって提供される外来エージェント気付アドレス
- モビリティエージェントによって提供されるモビリティサービスは、フラグとして通知されます。これは、エージェント通知の追加拡張です。

モビリティエージェントは、「エージェント通知」を送信してネットワークにサービスを通知します。エージェント通知がない場合は、モバイルノードは通知を要請できます。これを「エージェント要請」といいます。モバイルノードで共存気付アドレスをサポートしていない場合、モバイルノードはエージェント要請に通常のルーター通知を使用できます。次の節を参照してください。

エージェント通知

モバイルノードは、エージェント通知を使用してインターネットまたは組織のネットワークへの現在の接続点を決めます。エージェント通知とは、モビリティエージェント通知拡張も送信するように拡張されたインターネットコントロールメッセージプロトコル (ICMP) ルーター通知のことです。

外来エージェント (FA) は、忙しすぎて新たなモバイルノードを処理できない場合があります。しかし、外来エージェントはエージェント通知を継続して送信しなければなりません。このようにして、外来エージェントに登録済みのモバイルノードが、外来エージェントの有効範囲から外れていないことを認識できます。また、外来エージェントに障害が発生していないことも認識できます。外来エージェントに登録済みのモバイルノードが外来エージェントからエージェント通知を受信しない場合には、その外来エージェントと通信できないと認識します。

動的インタフェースによるエージェント通知

外来エージェントの実装を設定して、動的に作成されたインタフェースによって通知を送信できます。通知するインタフェースによる、要請されていない通知を制限するかどうかを決定できるオプションがあります。動的に作成されたインタフェースは、モバイル IP エージェントの開始後に設定されるインタフェースとしてのみ定義されます。動的インタフェースによる通知は、モバイルインタフェースを一時的にサポートするアプリケーションに有用です。さらに、要請されていない通知を制限することで、ネットワークの帯域幅を節約できます。

エージェント要請

各モバイルノードはエージェント要請を実装する必要があります。モバイルノードは、ICMP ルーターの要請メッセージ用に指定されたものと同じエージェント要請用の手順、デフォルト値、および定数を使用します。

モバイルノードが要請を送信する頻度は、モバイルノードによって制限されます。モバイルノードはエージェントの検索時に、1 秒間に最大 3 回初期要請を送信できます。モバイルノードをエージェントに登録した後は、要請を送信する頻度を減少させ、ローカルネットワークのオーバーヘッドを制限します。

気付アドレス

モバイル IP は、気付アドレスを取得するために次の代替モードを提供します。

- 外来エージェントは、エージェント通知メッセージを通してモバイルノードに通知される「気付アドレス」を提供します。通常、気付アドレスは、その通知を送信する外来エージェントの IP アドレスです。この場合、外来エージェントはトンネルのエンドポイントです。外来エージェントはトンネルを経由してデータグラムを受信し、そのデータグラムのカプセル化を解除します。その後、内部データグラムをモバイルノードに配信します。その結果、多数のモバイルノードが共存気付アドレスを共有できます。無線リンクでは、帯域幅が重要となります。モバイル IP サービスを高帯域幅の固定リンクに提供できる外来エージェントの中では、無線リンクがかなり有効です。
- モバイルノードは、「共存気付アドレス」をローカル IP アドレスとして取得します。その後、モバイルノードはこのアドレスをモバイルノードのネットワークインタフェースの1 つに関連付けます。また、DHCP を使ってこのアドレスを一時的アドレスとして取得することもできます。このアドレスを、モバイルノードが長期間アドレスとして所有する場合もあります。さらに、このアドレスが属するサブネットに移動している間だけそのアドレスを使用する場合もあります。共存気付アドレスを使用する場合、モバイルノードはトンネルの終点として機能します。その上、モバイルノードにトンネリングされたデータグラムのカプセル化を解除します。

共存気付アドレスにより、モバイルノードは外来エージェントなしで機能できます。その結果、モバイルノードは外来エージェントを配置していないネットワークで共存気付アドレスを使用できます。

共存気付アドレスをモバイルノードが使用している場合、モバイルノードはその気付アドレスのネットワーク接頭辞によって識別されるリンク上になければなりません。リンク上にないと、その気付アドレス宛てのデータグラムを配信できません。

逆方向トンネリングを使用するモバイル IP

前述の説明では、インターネット上の経路指定は、データパケット発信元アドレスから独立したものと想定されています。しかし、中間ルーターは、トポロジとして正しい発信元アドレスを確認します。中間ルーターが確認する場合は、モバイルノードで逆方向トンネルを設定しなければなりません。モバイルノードの気付アドレスからホームエージェントへ逆方向トンネルを設定することで、IP データパケットについてトポロジとして正しいソースアドレスを確保することができます。逆方向トンネルのサポートは、外来エージェントとホームエージェントによって通知されます。モバイルノードは、登録時に外来エージェントとホームエージェントの間に逆方向トンネルを要求できます。逆方向トンネルは、モバイルノードの気付アドレスで始まり、ホームエージェントで終わるトンネルです。図 23–4 に逆方向トンネルを使用するモバイル IP トポロジを示します。

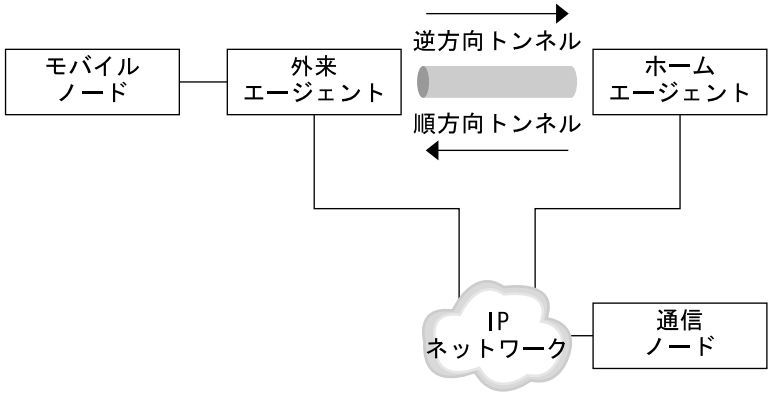


図 **23–4** 逆方向トンネルを使用するモバイル IP

専用アドレスの制限付きサポート

専用アドレスを持ち、インターネットを経由してグローバルに経路指定できないモバイルノードには、逆方向トンネルが必要です。Solaris モバイル IP は、専用アドレスを持つモバイルノードをサポートします。Solaris モバイル IP がサポートしない機能については、443 ページの「Solaris モバイル IP 実装の概要」を参照してください。

外部との接続が必要でない場合、ネットワークでは専用アドレスを使います。専用アドレスは、インターネットを通る経路指定ができません。専用アドレスを持つモバイルノードは、データグラムをそのホームエージェントに逆方向トンネリングを設定することによって通信ノードとだけ通信できます。 通常、モバイルノードがホームにあ

るときにデータグラムが配信される場合でも、ホームエージェントはデータグラムを通信ノードに配信します。次の図は、専用アドレスが指定された 2 つのモバイルノードのネットワークトポロジを示します。その 2 つのモバイルノードは、同じ外来エージェントに登録されたときに同じ気付アドレスを使用します。

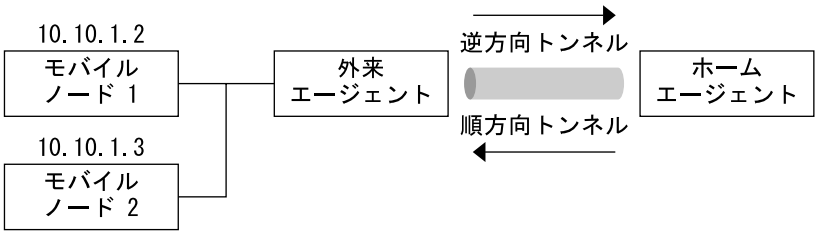


図 **23–5** 同じ外部ネットワーク上にある、専用アドレスが指定されたモバイルノード

気付アドレスとホームエージェントの IP アドレスが公衆インターネットによって接続される異なるドメインに属する場合、それらのアドレスはグローバルに経路指定できるアドレスでなければなりません。

同じ外部ネットワーク上に、同じ IP アドレスを持つ、専用アドレスが指定された 2 つのモバイルノードを持つことは可能です。ただし、各モバイルノードが異なるホームエージェントを持っていなければなりません。さらに、各モバイルノードが共通の 1 つの外来エージェントの異なる通知サブネット上になければなりません。図 23–6 は、このような状況を表わすネットワークトポロジを示しています。

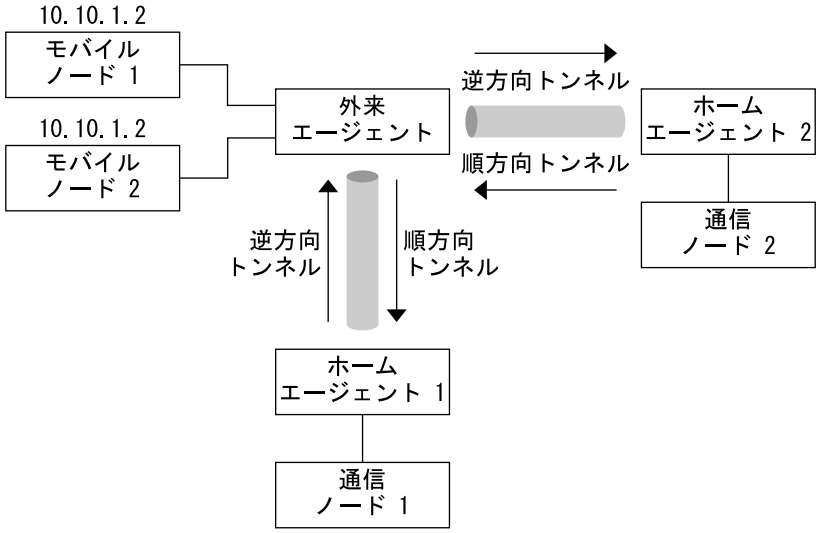


図 **23–6** 異なる外部ネットワーク上にある、専用アドレスが指定されたモバイルノード

モバイル IP の登録

モバイルノードは、エージェント通知を利用してサブネット間を移動した時期を検出します。モバイルノードは、その場所を変更したことを示すエージェント通知を受信すると、外来エージェントを経由して登録します。モバイルノードは、共存気付アドレスを取得できる場合でも、この機能によってサイトはモビリティサービスへのアクセスを制限できます。

モバイル IP 登録機能は、モバイルノードの現在の到達可能情報をホームエージェントに通知するための融通性のある機構を提供します。登録処理によってモバイルノードは次の作業を実行できます。

- 外部ネットワークに移動する際の要求転送サービス
- ホームエージェントへの現在の気付アドレスの通知
- 期間が満了する登録の更新
- ホームに戻る際の登録解除
- 逆方向トンネルの要求

登録メッセージは、モバイルノード、外来エージェント、およびホームエージェント間の情報を交換します。登録によってホームエージェントでのモビリティ結合を作成または変更します。指定された期間にモバイルノードのホームアドレスをその気付アドレスに関連付けます。

登録処理によってモバイルノードは次の機能を実行できます。

- 複数の外来エージェントへ登録する
- 他のモビリティ結合を維持しながら特定の気付アドレスの登録を解除する
- モバイルノードがこの情報で構成されていない場合にホームエージェントのアドレスを発見する

モバイル IP は、モバイルノードに対して次の登録処理を定義します。

- モバイルノードが外来エージェントの気付アドレスを登録する場合、モバイルノードはその外来エージェントを使用して到達可能なホームエージェントを通知する。
- モバイルノードが外来エージェントを使用してその登録を要求するエージェント通知を受信する場合でも、モバイルノードは共存気付アドレスを取得できる。モバイルノードは、その外来エージェントあるいはこのリンク上の別の外来エージェントに登録することもできる。
- モバイルノードが共存気付アドレスを使用する場合、自分のホームエージェントに直接登録する。
- モバイルノードがホームネットワークに戻るときにホームエージェントでの登録を解除する。

これらの処理には登録要求および登録応答メッセージの交換が伴います。外来エージェントを使用して登録する場合、登録処理は次の手順で行われます。

1. モバイルノードは、可能性がある外来エージェントに登録要求を送信して、登録処理を開始します。
2. 外来エージェントは登録要求を処理し、その要求をホームエージェントに転送します。
3. ホームエージェントは登録応答を外来エージェントに送信し、要求を承認または否認します。
4. 外来エージェントは登録応答を処理し、その応答をモバイルノードに転送して、その要求を処理したことを通知します。

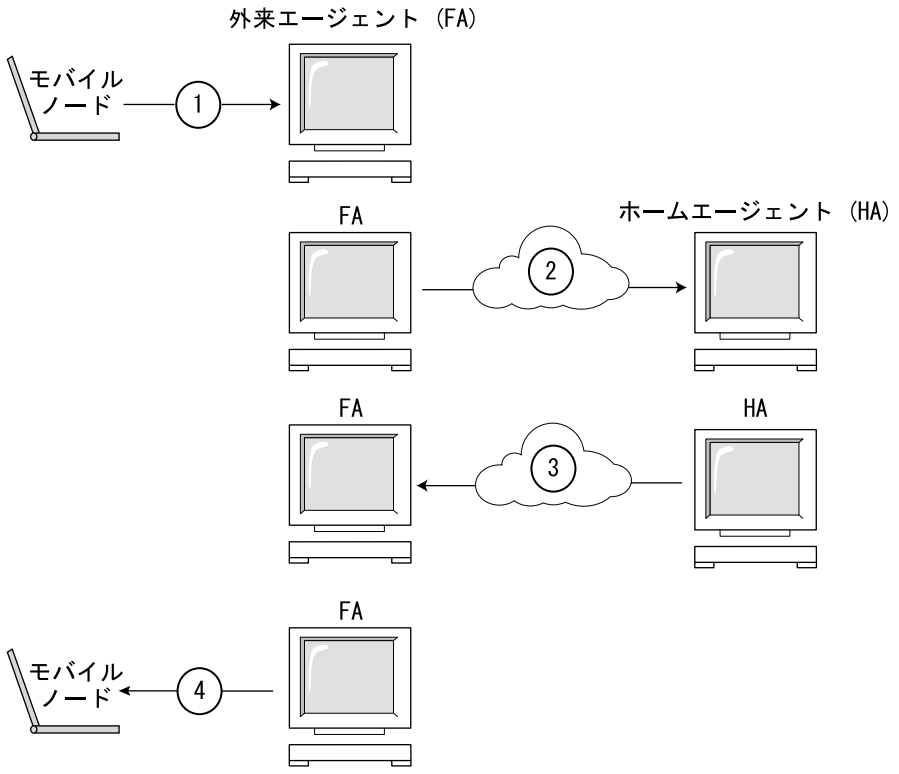


図 **23–7** モバイル IP の登録処理

モバイルノードがホームエージェントに直接登録する場合、登録処理には次の手順が必要です。

- モバイルノードが登録取り消し要求をホームエージェントに送信する。
- ホームエージェントが登録応答をモバイルノードに送信して、要求を承認または否認する。

また、逆方向トンネルが外来エージェントまたはホームエージェントのいずれかに要求されます。外来エージェントが逆方向トンネリングをサポートする場合、モバイルノードは登録処理を使用して、逆方向トンネルを要求します。モバイルノードは、登録要求で逆方向トンネルフラグを設定することによって、逆方向トンネルを要求します。

ネットワークアクセス識別子 (NAI)

インターネット内で使用している AAA サーバーは、ダイアルアップコンピュータ用の認証および承認サービスを提供します。これらのサービスは、ノードが AAA サーバーにより外部ドメインに接続しようとしているときにモバイル IP を使用しているモバイルノードにも、同様に価値がある可能性があります。AAA サーバーは、ネットワークアクセス識別子(NAI)を使ってクライアントを特定します。モバイルノードは NAI をモバイル IP 登録要求に含めることによって自分自身を識別することができます。

NAI は通常モバイルノードを特定するために使用されるので、モバイルノードのホームアドレスが必ずしもこの機能を提供する必要はありません。したがって、モバイルノードでそれ自体を認証します。その結果、モバイルノードではホームアドレスがない場合でも、外部ドメインへ接続するための承認を得ることができます。ホームアドレスの割り当てを要求するために、モバイルノードの NAI 拡張を含むメッセージは登録要求内でホームアドレスをゼロに設定することができます。

モバイル IP メッセージの認証

各モバイルノード、外来エージェント、およびホームエージェントは、さまざまなモバイル IP 構成要素間のモビリティセキュリティアソシエーションを提供します。セキュリティアソシエーションは、セキュリティパラメータインデックス (SPI) と IP アドレスで索引付けされています。モバイルノードの場合、このアドレスはモバイルノードのホームアドレスです。モバイルノードとそのホームエージェント間の登録メッセージは、モバイルホーム間認証拡張により認証されます。必須であるモバイルホーム間認証拡張に加え、ユーザーは任意のモバイルと外来エージェント間、およびホームと外来エージェント間認証を使用できます。

モバイルノード登録要求

モバイルノードは、「登録要求」メッセージを使用してそのホームエージェントに登録します。 このようにして、ホームエージェントが (たとえば新しい有効期間をもつ) そのモバイルノード用のモビリティ結合を作成または変更できるようにします。外来エージェントは登録要求をホームエージェントに転送できます。ただしモバイルノードが、共存気付アドレスを登録している場合には、モバイルノードはその登録要求を直接ホームエージェントに送信できます。外来エージェントが、登録メッセージを送信する必要があることを通知する場合、モバイルノードは登録要求を外来エージェントに送信しなければなりません。

登録応答メッセージ

モビリティエージェントは、登録要求メッセージを送信したモバイルノードに「登録応答」メッセージを返します。モバイルノードが外来エージェントにサービスを要求している場合、その外来エージェントはホームエージェントから応答を受信します。その後、外来エージェントはその応答をモバイルノードに転送します。応答メッセージには、登録要求の状態についてモバイルノードと外来エージェントに通知するのに必要なコードが含まれています。また、ホームエージェントにより許可されている有効期間も含まれています。有効期間は元の要求よりも短い可能性があります。登録応答には動的ホームアドレス割り当てが含まれることがあります。

外来エージェント

外来エージェントは、ほとんどの場合モバイル IP の登録において受動的役割を果たします。また、ビジターテーブルに登録されているモバイルノードをすべて追加します。外来エージェントは、登録要求をモバイルノードと ホームエージェント間で転送します。また、気付アドレスをサポートしている場合は、データグ ラムをカプセル化解除してモバイルノードに配信します。さらに、周期的エージェント通知メッセージを送信して外来エージェントの存在を通知します。

ホームエージェントと外来エージェントが逆方向トンネルをサポートし、モバイルノードが逆方向トンネルを要求する場合、外来エージェントはすべてのパケットをモバイルノードからホームエージェントへトンネリングします。 その後、ホームエージェントはそのパケットを通信ノードに送信します。この処理は、モバイルノードへの配信用にホームエージェントがモバイルノードのすべてのパケットを外来エージェントにトンネリングする場合と逆です。逆方向トンネルをサポートしている外来エージェントは、登録のために逆方向トンネルをサポートしていることを通知します。ローカルポリシーにより、外来エージェントは、逆方向トンネルフラグが設定されていないときに、登録要求を拒否することができます。また、モバイルノードが外来エージェント上の異なる 2 つのインタフェースに移動するときに、外来エージェントが特定できるのは、同じ (専用) IP アドレスを持つ複数のモバイルノードだけです。順方向トンネルの場合、外来エージェントは、着信側のトンネルインタフェースを調べることによって、同じ専用アドレスを共有する複数のモバイルノードを特定します。 着信トンネルインタフェースは、固有のホームエージェントのアドレスに対応します。

ホームエージェント

ホームエージェントは、モバイル IP の登録処理において能動的役割を果たします。ホームエージェントは、モバイルノードから登録要求を受信します。 登録要求は、外来エージェントによって転送されます。また、このモバイルノードに対するモビリティ結合の記録を更新します。 さらに、各登録要求に対して適切な登録応答を発行します。その上、モバイルノードがホームネットワークから離れているときには、そのモバイルノードにパケットを転送します。

ホームエージェントは、モバイルノード用に構成された物理サブネットを持たなければいけないわけではありません。ただし、ホームエージェントは、登録を承認するときに mipagent.conf ファイルまたは他の機構を使用してモバイルノードのホームアドレスを認識しなければなりません。

専用アドレスが指定されたモバイルノードをサポートするには、mipagent.conf ファイルで専用アドレスが指定されたモバイルノードを設定します。ホームエージェントで使用されるホームアドレスは一意にする必要があります。

動的ホームエージェントの発見

モバイルノードは、登録しようとする際にそのホームエージェントのアドレスを認識していないことがあります。モバイルノードがそのホームエージェントのアドレスを認識していない場合、動的ホームエージェントアドレス解決を使用してホームエージェントのアドレスを認識できます。この場合、モバイルノードは登録要求のホームエージェントフィールドをモバイルノードのホームネットワークのサブネット指定のブロードキャストアドレスに設定します。ブロードキャスト宛先アドレスが指定された登録要求を受信した各ホームエージェントは、拒否登録応答を返信することによってモバイルノードの登録を拒否します。こうすることによってモバイルノードは、拒否応答に示された、ホームエージェントのユニキャスト IP アドレスを次に登録を行う際に使用できます。

モバイルノードに対するデータグラムの経路指定

モバイルノード、ホームエージェント、および外来エージェントが協力して、外部ネットワークに接続されているモバイルノードへのデータグラムの経路を指定する方法を説明します。Solaris オペレーティング環境でサポートされているモバイル IP の機能については、443 ページの「Solaris モバイル IP 実装の概要」を参照してください。

カプセル化の種類

ホームエージェントおよび外来エージェントは、利用可能なカプセル化の方法のいずれか 1 つを使用してデータグラムのトンネリングを提供します。定義されているカプセル化の方法は、IP 内 IP (IP-in-IP) カプセル化、最小カプセル化、および汎用経路指定カプセル化です。外来エージェントおよびホームエージェント (つまり、モバイルノードとホームエージェントが間接的に共存する場合)では、同じカプセル化の方法をサポートする必要があります。また、すべてのモバイル IP エントリが IP 内 IP カプセル化をサポートする必要があります。

ユニキャストデータグラムの経路指定

外部ネットワークに登録された場合、モバイルノードは次に示す規則を使用してデフォルトのルーターを選択します。

- モバイルノードが外来エージェントの気付アドレスを使用して登録された場合、その処理は直線的な順方向になります。モバイルノードは、ICMP ルーター通知メッセージのいずれかで通知されているものの中からデフォルトのルーターを選択します。また、エージェント通知の IP 発信元アドレスをデフォルトルーターの IP アドレスに対するもう1 つの選択候補とすることができます。
- モバイルノードは、共存気付アドレスを使用して直接ホームエージェントに登録できます。その後、モバイルノードが受信する ICMP ルーター通知メッセージのいずれかで通知されているものの中からデフォルトのルーターを選択します。 選択したデフォルトルーターのネットワーク接頭辞は、モバイルノードが外部で取得した、共存気付アドレスに一致しなければなりません。よって、そのアドレスは、ネットワーク接頭辞でのエージェント通知 の IP 発信元アドレスに一致します。さらに、モバイルノードはその IP 発信元アドレスをデフォルトルーターの IP アドレスに対するもう1 つの選択候補とすることができます。
- モバイルノードが登録されている場合、逆方向トンネルをサポートする外来エージェントは、モバイルノードから逆方向トンネルを経由してホームエージェントにユニキャストデータグラムを経路指定します。モバイルノードが逆方向トンネルをサポートする外来エージェントに登録されている場合には、デフォルトルーターとしてその外来エージェントを使用する必要があります。

ブロードキャストデータグラム

ホームエージェントがブロードキャストデータグラムまたはマルチキャストデータグラムを受信したときは、ホームエージェントが受信するモバイルノードに対してそのデータグラムだけを転送します。ブロードキャストデータグラムおよびマルチキャストデータグラムをモバイルノードに転送する方法は、主に 2 つの要素によって異なります。モバイルノードで外来エージェントが提供する気付アドレスを使用するか、その独自の共存気付アドレスを使用するかという 2 つの要素です。気付アドレスを使用する場合、データグラムを二重カプセル化する必要があります。最初の IP ヘッダーは、データグラムの配信先となるモバイルノードを示します。最初の IP ヘッダーは、ブロードキャストデータグラムまたはマルチキャストデータグラムには存在しないので注意してください。2 番目の IP ヘッダーは、気付アドレスを示し、その通常のトンネルヘッダーとなります。独自の共存気付アドレスを使用する場合、モバイルノードはその独自のデータグラムのカプセル化を解除し、そのデータグラムを通常のトンネル経由のみで送信する必要があります。

マルチキャストデータグラムの経路指定

モバイルノードが外部サブネットの移動時に、マルチキャストトラフィックの受信を開始するには、次のいずれかの方法でマルチキャストグループを結合します。

- モバイルノードが共存気付アドレスを使用している場合には、このアドレスをインターネットグループ管理プロトコル (IGMP) 結合メッセージの発信元 IP アドレスとして使用します。ただし、マルチキャストルーターが移動先のサブネットに存在していなければなりません。
- モバイルノードがそのホームサブネットから ICMP グループを結合する場合、逆方向トンネルを使用して IGMP 結合メッセージをホームエージェントに送信する必要があります。ただし、モバイルノードのホームエージェントをマルチキャストルーターにする必要があります。ホームエージェントはその後、マルチキャストデータグラムをトンネルを通してモバイルノードまで転送します。
- モバイルノードが、共存気付アドレスを使用している場合には、このアドレスを IGMP 結合メッセージの発信元 IP アドレスとして使用します。ただし、マルチキャストルーターが移動先のサブネットに存在していなければなりません。結合されると、モバイルノードは移動先のネットワークに直接独自のマルチキャストパケットを送信することによって加入できます。
- 移動先ネットワークに直接送信する
- トンネルを通して自分のホームエージェントに送信する

マルチキャストの経路指定は IP 発信元アドレスに依存しています。マルチキャストデータグラムを送信するモバイルノードは、そのリンクで有効な発信元アドレスからそのデータグラムを送信する必要があります。したがって、マルチキャストデータグラムを移動先ネットワークに直接送信するモバイルノードは、共存気付アドレスを IP 発信元アドレスとして使用します。また、モバイルノードはそのアドレスに関連付けられるマルチキャストグループを結合する必要もあります。同様に、移動前にホームサブネットでマルチキャストデータグラムを結合する、またはホームエージェントへの逆方向トンネルを通して移動中にマルチキャストグループを結合するモバイルノードは、そのホームアドレスをマルチキャストデータグラムの IP 発信元アドレスとして使用します。したがって、モバイルノードはそのホームサブネットにそれらのデータグラムを逆方向トンネルで送信する必要があると同様に、その共存気付アドレスを使用してモバイルノード自体または外来エージェントの逆方向トンネルのいずれかも使用します。

モバイルノードが移動先のサブネットから常に結合している方が効率的であると思われる場合、モバイルノードのままにします。よって、モバイルノードはサブネットに移動するたびにその結合を繰り返すことになります。モバイルノードがそのホームエージェントを通して結合した方が効率的である場合には、このオーバーヘッドを処理する必要はありません。また、マルチキャストセッションはホームサブネットで有効な場合に限り存在します。さらに、特定の方法でモバイルノードが加入するためには、その他にも留意点があります。

セキュリティについて

多くの場合、モバイルコンピュータは無線リンクを利用してネットワークに接続されます。無線リンクは特に盗聴、攻撃に対して脆弱です。

モバイルIP はこの脆弱性を低下あるいは除去することはできないため、それらの攻撃に対してモバイル IP 登録メッセージを保護するために認証形式を使用します。使用しているデフォルトのアルゴリズムは、128 ビットの鍵を採用した MD5 です。デフォルトの動作モードでは、ハッシュしようとするデータの前後にこの 128 ビット鍵がある必要があります。外来エージェントは、MD5 を使用して認証をサポートします。また、128 ビット以上の鍵サイズ、および手動による鍵配布を使用し た認証もサポートしています。モバイル IP では、より多くの認証アルゴリズム、アルゴリズムモード、鍵の配布方法、および鍵サイズをサポートできます。

これらの方法により、モバイル IP 登録メッセージの改ざんを防止します。さらに、前のモバイル IP 登録メッセージと重複するメッセージを受信した場合、モバイル IP はモバイル IP の要素を警告する応答保護形式も使用します。この保護方法を使用しないと、登録メッセージの受信時にモバイルノードとそのホームエージェントが同期をとることができなくなります。そのため、モバイル IP はその状態を更新します。たとえば、モバイルノードが外来エージェントを通して登録している間に、ホームエージェントが重複する登録解除メッセージを受信したとします。その場合、ナンス (Nonce) と呼ばれる方法またはタイムスタンプによって、応答保護を確立します。ナンスおよびタイムスタンプは、モバイル IP 登録メッセージ内でホームエージェントとモバイルノードによって交換されます。また、前に説明した認証機構によって変更されないように保護されます。その結果、ホームエージェントまたはモバイルノードが重複するメッセージを確認した場合、そのメッセージを破棄できます。

トンネリングは非常に攻撃されやすく、特に登録が認証されていない場合に脆弱です。また、アドレス解決プロトコル (ARP) は認証されていないため、別のホストのトラフィックを盗むために利用される可能性があります。

モバイル IP による IPsec の使用

一般的には、ホームエージェントと外来エージェントは固定要素であるため、IPsec 認証または暗号化を使用して両方のモバイル IP 登録メッセージを保護し、トラフィックを順方向および逆方向にトンネリングします。この処理は、モバイル IP から完全に独立して行われますが、IPsec の機能を実行するワークセンタの能力には依存します。また、モバイルノードは IPsec 認証を使用してその登録トラフィックを保護します。モバイルノードが外来エージェントに登録する場合には通常、IPsec 暗号化を使用できません。その理由は、外来エージェントで登録パケットの情報をチェック可能にする必要があるためです。外来エージェントでその必要がない場合には、IPsec 暗号化を使用できますが、この問題は共存により発生します。IPsec とは、IP レベルのセキュリティ関係です。したがって、ホームエージェントが事前の情報メッセージ

または登録メッセージなしでモバイルノードの共存アドレスを認識する必要があります。この情報の必要性をなくすことができるプロトコルもありますが、このマニュアルでは説明していません。IPsec の詳細については、第 19 章または第 20 章を参照してください。

第 24 章

モバイル IP の管理 (手順)

この章では、モバイル IP 構成ファイルのパラメータの変更、追加、削除、および表示の方法について説明します。また、モビリティエージェント状態の表示方法についても説明します。

この章では、以下の内容について説明します。

モバイル IP 構成ファイルの構成

`mipagent.conf`ファイルを最初に構成するときには、次の作業を実行する必要があります。

1. ユーザーの組織のホスト条件によって、モバイル IP エージェントが提供できる機能を決めます。
   - 外来エージェント機能のみ
   - ホームエージェント機能のみ
   - 外来エージェントとホームエージェント機能の両方
2. `/etc/inet/mipagent.conf` ファイルを作成し、この節で説明する手順に従って必要な設定を入力します。次に示すファイルの 1 つを `/etc/inet/mipagent.conf` にコピーし、要求条件に応じて変更することもできます。
   - 外来エージェント機能用には、`/etc/inet/mipagent.conf.fa-sample` をコピーします。

- ホームエージェント機能用には、/etc/inet/mipagent.conf.ha-sample をコピーします。
- 外来エージェントとホームエージェントの両機能用には、/etc/inet/mipagent.conf-sample をコピーします。

3. システムをリブートして mipagent デーモンを起動するブートスクリプトを呼び出します。次のコマンドをコマンド行で入力して mipagent を起動することもできます。

```
# /etc/inet.d/mipagent start
```

モバイル IP 構成ファイルの構成 (作業マップ)

表 24–1 に、この節で説明している作業の概要を示します。

表 **24–1** モバイル IP 構成ファイルの構成 (作業マップ)

タスク	説明	参照先
モバイル IP 構成ファイルの作成	/etc/inet/mipagent.conf ファイルの作成またはサンプルファイルの 1 つのコピーを含む。	429 ページの「モバイル IP 構成ファイルを作成する方法」
General セクションの構成	バージョン番号のモバイル IP 構成ファイルの General セクションへの挿入を含む。	429 ページの「General セクションを構成する方法」
Advertisements セクションの構成	ラベルおよび設定値の追加、またはモバイル IP 構成ファイルの Advertisements セクション内のラベルおよび設定値の変更を含む。	430 ページの「Advertisements セクションを構成する方法」
GlobalSecurityParameters セクションの構成	ラベルおよび設定値の追加、またはモバイル IP 構成ファイルの GlobalSecurityParameters セクション内のラベルおよび設定値の変更を含む。	430 ページの「GlobalSecurityParameters セクションを構成する方法」
Pool セクションの構成	ラベルおよび設定値の追加、またはモバイル IP 構成ファイルの Pool セクション内のラベルおよび設定値の変更を含む。	430 ページの「Pool セクションを構成する方法」
SPI セクションの構成	ラベルおよび設定値の追加、またはモバイル IP 構成ファイルの SPI セクションのラベルおよび設定値の変更を含む。	431 ページの「SPI セクションを構成する方法」

表 **24–1** モバイル IP 構成ファイルの構成 (作業マップ)　(続き)

タスク	説明	参照先
Address セクションの構成	ラベルおよび設定値の追加、またはモバイル IP 構成ファイルの Address セクションのラベルおよび設定値の変更を含む。	431 ページの「Address セクションを構成する方法」

▼ モバイル IP 構成ファイルを作成する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. 必要に応じて次のいずれかの手順を実行します。
   - /etc/inet ディレクトリで mipagent.conf という空ファイルを作成します。
   - 次のリストから必要な機能を提供するサンプルファイルを選択して、/etc/inet/mipagent.conf にコピーします。
     - /etc/inet/mipagent.conf.fa-sample
     - /etc/inet/mipagent.conf.ha-sample
     - /etc/inet/mipagent.conf-sample
3. 構成条件に応じて **/etc/inet/mipagent.conf** ファイル内に構成パラメータを追加または変更します。実行する手順については、後の節で説明します。

▼ General セクションを構成する方法

サンプルファイルをコピーした場合は、サンプルファイルにこの項目があるのでこの手順を省略できます。

- **/etc/inet/mipagent.conf** ファイルを編集して次の行を追加します。

```
[General]
     Version = 1.0
```

注 – /etc/inet/mipagent.conf ファイルには、上記の項目が入っていなければなりません。

---

449 ページの「General セクション」では、この節で使用しているラベルおよび設定値について説明しています。

▼ Advertisements セクションを構成する方法

- **/etc/inet/mipagent.conf** ファイルを編集し、構成に必要な設定値を使用して次の行を追加または変更します。

```
[Advertisements Interface-name]
     HomeAgent = <yes/no>
     ForeignAgent = <yes/no>
     PrefixFlags = <yes/no>
     AdvertiseOnBcast = <yes/no>
     RegLifetime = n
     AdvLifetime = n
     AdvFrequency = n
     ReverseTunnel = <yes/no/FA/HA/both>
     ReverseTunnelRequired = <yes/no/FA/HA>
```

注 – モバイル IP サービスを提供するローカルホストの各インタフェースには、それぞれ異なる Advertisements セクションを指定しなければなりません。

---

449 ページの「Advertisements セクション」では、この節で使用しているラベルおよび設定値について説明しています。

▼ GlobalSecurityParameters セクションを構成する方法

- **/etc/inet/mipagent.conf** ファイルを編集し、構成に必要な設定値を使用して次の行を追加または変更します。

```
[GlobalSecurityParameters]
     MaxClockSkew = n
     HA-FAauth = <yes/no>
     MN-FAauth = <yes/no>
     Challenge = <yes/no>
     KeyDistribution = files
```

451 ページの「GlobalSecurityParameters セクション」では、この節で使用しているラベルおよび設定値について説明しています。

▼ Pool セクションを構成する方法

- **/etc/inet/mipagent.conf** ファイルを編集し、構成に必要な設定値を使用して次の行を追加または変更します。

```
[Pool Pool-identifier]
     BaseAddress = IP-address
     Size = size
```

452 ページの「Pool セクション」では、この節で使用しているラベルおよび設定値について説明しています。

▼ SPI セクションを構成する方法

- **/etc/inet/mipagent.conf** ファイルを編集し、構成に必要な設定値を使用して次の行を追加または変更します。

```
[SPI SPI-identifier]
     ReplayMethod = <none/timestamps>
     Key = key
```

注 – 配置した各セキュリティコンテキストに対して異なる SPI セクションを指定しなければなりません。

---

453 ページの「SPI セクション」では、この節で使用しているラベルおよび設定値について説明しています。

▼ Address セクションを構成する方法

- **/etc/inet/mipagent.conf** ファイルを編集し、構成に必要な設定値を使用して次の行を追加または変更します。
  - モバイルノード用

```
[Address address]
     Type = node
     SPI = SPI-identifier
```

  - エージェント用

```
[Address address]
     Type = agent
     SPI = SPI-identifier
     IPsecRequest = action {properties} [: action {properties}]
     IPsecReply = action {properties} [: action {properties}]
     IPsecTunnel = action {properties} [: action {properties}]
```

    action と *{properties}* は、ipsec(7P) のマニュアルページで定義される、任意のアクションプロパティと関連付けられたプロパティです。

注 – 先に構成される SPI は、RFC 2002 によって要求される MD5 保護機構に対応します。 ただし、IPsec で使用される SPI には対応しません。 IPsec の詳細については、第 19 章と第 20 章を参照してください。 また、`ipsec(7P)` マニュアルページも参照してください。

- 自分の **NAI** で識別されるモバイルノード用

```
[Address NAI]
     Type = Node
     SPI = SPI-identifier
     Pool = Pool-identifier
```

- デフォルトのモバイルノード用

```
[Address Node-Default]
     Type = Node
     SPI = SPI-identifier
     Pool = Pool-identifier
```

454 ページの「`Address` セクション」では、この節で使用しているラベルおよび設定値について説明しています。

モバイル IP 構成ファイルの変更

この節では、`mipagentconfig(1M)` コマンドを使用してモバイル IP 構成ファイルを変更する方法を説明します。パラメータの宛先の現在の設定値を表示する方法についても説明します。

458 ページの「モビリティ IP エージェントの構成」では、`mipagentconfig(1M)` コマンドの使用法について説明しています。使用法については、`mipagentconfig(1M)` のマニュアルページでも説明しています。

モバイル IP 構成ファイルの変更 (作業マップ)

表 **24–2** モバイル IP 構成ファイルの変更 (作業マップ)

タスク	説明	参照先
`General` セクションの変更	`mipagentconfig change` コマンドを使用してモバイル IP 構成ファイルの `General` セクション内のラベル値を変更します。	433 ページの「`General` セクションを変更する方法」

表 **24–2** モバイル IP 構成ファイルの変更 (作業マップ)　(続き)

タスク	説明	参照先
Advertisements セクションの変更	mipagentconfig change コマンドを使用してモバイル IP 構成ファイルの Advertisements セクション内のラベル値を変更します。	434 ページの「Advertisements セクションを変更する方法」
GlobalSecurityParameters セクションの変更	mipagentconfig change コマンドを使用してモバイル IP 構成ファイルの GlobalSecurityParameters セクション内のラベル値を変更します。	434 ページの「GlobalSecurityParameters セクションを変更する方法」
Pool セクションの変更	mipagentconfig change コマンドを使用してモバイル IP 構成ファイルの Pool セクション内のラベル値を変更します。	435 ページの「Pool セクションを変更する方法」
SPI セクションの変更	mipagentconfig change コマンドを使用してモバイル IP 構成ファイルの SPI セクション内のラベル値を変更します。	435 ページの「SPI セクションを変更する方法」
Address セクションの変更	mipagentconfig change コマンドを使用してモバイル IP 構成ファイルの Address セクション内のラベル値を変更します。	435 ページの「Address セクションを変更する方法」
パラメータの追加または削除	mipagentconfig add または delete コマンドを使用して新しいパラメータ、ラベル、または設定値を追加、あるいはモバイル IP 構成ファイルの任意のセクション内の既存の項目を変更します。	436 ページの「構成ファイルのパラメータを追加または削除する方法」
パラメータ宛先の現在の設定値の表示	mipagentconfig get コマンドを使用してモバイル IP 構成ファイルの任意のセクション内の現在の設定値を表示します。	437 ページの「構成ファイルの現在のパラメータ設定を表示する方法」

▼ General セクションを変更する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. コマンド行で、**General** セクション内の変更したい各ラベルに対して次のコマンドを入力します。

```
# mipagentconfig change <label> <value>
```

例 24–1 では、構成ファイルの General セクション内のバージョン番号を変更する方法を示しています。

例 **24–1** General セクションのパラメータの変更

```
# mipagentconfig change version 2
```

▼ Advertisements セクションを変更する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. コマンド行で、**Advertisements** セクション内の変更したい各ラベルに対して次のコマンドを入力します。

   ```
   # mipagentconfig change adv device-name <label> <value>
   ```

たとえば、エージェントの通知された有効期間をデバイス le0 に対して 300 秒に変更したい場合、次のコマンドを使用して変更します。

```
# mipagentconfig change adv le0 AdvLifetime 300
```

では、構成ファイルの Advertisements セクション内のその他のパラメータを変更する方法を示しています。

例 **24–2** Advertisements セクションのパラメータの変更

```
# mipagentconfig change adv le0 HomeAgent yes
# mipagentconfig change adv le0 ForeignAgent no
# mipagentconfig change adv le0 PrefixFlags no
# mipagentconfig change adv le0 RegLifetime 300
# mipagentconfig change adv le0 AdvFrequency 4
# mipagentconfig change adv le0 ReverseTunnel yes
```

▼ GlobalSecurityParameters セクションを変更する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. コマンド行で、**GlobalSecurityParameters** セクション内の変更したい各ラベルに対して次のコマンドを入力します。

   ```
   # mipagentconfig change <label> <value>
   ```

たとえば、ホームエージェントおよび外来エージェント認証を有効にしたい場合は、次のコマンドを使用して変更します。

```
# mipagentconfig change HA-FAauth yes
```

例 24–3 では、構成ファイルの GlobalSecurityParameters セクション内のその他のパラメータを変更する方法を示しています。

例 **24–3** GlobalSecurityParameters セクションのパラメータの変更

```
# mipagentconfig change MaxClockSkew 200
# mipagentconfig change MN-FAauth yes
```

例 **24–3** GlobalSecurityParameters セクションのパラメータの変更　(続き)

```
# mipagentconfig change Challenge yes
# mipagentconfig change KeyDistribution files
```

▼ Pool セクションを変更する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. コマンド行で、**Pool** セクション内の変更したい各ラベルに対して次のコマンドを入力します。

   ```
   # mipagentconfig change Pool Pool-identifier <label> <value>
   ```

たとえば、Pool 10 の基底アドレスを 192.168.1.1 に、サイズを 100 に変更したい場合、次のコマンドを使用して変更します。

例 **24–4** Pool セクションのパラメータの変更

```
# mipagentconfig change Pool 10 BaseAddress 192.168.1.1
# mipagentconfig change Pool 10 Size 100
```

▼ SPI セクションを変更する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. コマンド行で、**SPI** セクション内の変更したい各ラベルに対して次のコマンドを入力します。

   ```
   # mipagentconfig change SPI SPI-identifier <label> <value>
   ```

たとえば、SPI 257 のキーを 5af2aee39ff0b332 に変更したい場合、次のコマンドを使用して変更します。

```
# mipagentconfig change SPI 257 Key 5af2aee39ff0b332
```

例 24–5 では、構成ファイルの SPI セクション内の ReplayMethod ラベルを変更する方法を示しています。

例 **24–5** SPI セクションのパラメータの変更

```
# mipagentconfig change SPI 257 ReplayMethod timestamps
```

▼ Address セクションを変更する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。

2. コマンド行で、**Address** セクション内の変更したい各ラベルに対して次のコマンドを入力します。

   ```
   # mipagentconfig change addr [NAI | IPaddr | node-default] <label> <value>
   ```

   3 つの構成方法 (NAI、IP アドレス、デフォルトノード) については、454 ページの「Address セクション」を参照してください。

たとえば、IP アドレス 10.1.1.1 の SPI を 258 に変更したい場合は、次のコマンドを使用して変更します。

```
# mipagentconfig change addr 10.1.1.1 SPI 258
```

例 24–6 では、サンプル構成ファイルの Address セクションに指定されたその他のパラメータを変更する方法を示しています。

例 **24–6** Address セクションのパラメータの変更

```
# mipagentconfig change addr 10.1.1.1 Type agent
# mipagentconfig change addr 10.1.1.1 SPI 259
# mipagentconfig change addr mobilenode@abc.com Type node
# mipagentconfig change addr mobilenode@abc.com SPI 258
# mipagentconfig change addr mobilenode@abc.com Pool 2
# mipagentconfig change addr node-default SPI 259
# mipagentconfig change addr node-default Pool 3
# mipagentconfig change addr 10.68.30.36 Type agent
# mipagentconfig change addr 10.68.30.36 SPI 260
# mipagentconfig change IPsecRequest apply {auth_algs md5 sa shared}
```

▼ 構成ファイルのパラメータを追加または削除する方法

1. モバイル **IP** を有効にしたいシステムでスーパーユーザーになります。
2. コマンド行で、指定したセクションに対して追加または削除したい各ラベルについてコマンドを入力します。

   General セクション

   ```
   # mipagentconfig [add | delete] <label> <value>
   ```

   Advertisements セクション

   ```
   # mipagentconfig [add | delete] adv device-name <label> <value>
   ```

注 – 次のコマンドを入力してインタフェースを追加できます。

```
# mipagentconfig add adv device-name
```

この場合、デフォルト値は (外来エージェントおよびホームエージェントに対する) インタフェースに割り当てられます。

GlobalSecurityParameters セクション

```
# mipagentconfig [add | delete] <label> <value>
```

Pool セクション

```
# mipagentconfig [add | delete] Pool Pool-identifier <label> <value>
```

SPI セクション

```
# mipagentconfig [add | delete] SPI SPI-identifier <label> <value>
```

Address セクション

```
# mipagentconfig [add | delete] addr [NAI | IPaddr | node-default] \
<label> <value>
```

注 – 同じ内容の Advertisements 、Pool 、SPI 、および Address セクションは作成できないので注意してください。

たとえば、基底アドレスが 192.167.1.1 でサイズが 100 の新しいアドレスプール Pool 11 を作成したい場合、次のコマンドを使用します。

例 **24–7** 新しいプールおよびパラメータの追加

```
# mipagentconfig add Pool 11 BaseAddress 192.167.1.1
# mipagentconfig add Pool 11 size 100
```

また、特定のセキュリティパラメータを削除したい場合もあります。例 24–8 では、SPI 257 を削除する方法を示しています。

例 **24–8** SPI の削除

```
# mipagentconfig delete SPI 257
```

▼ 構成ファイルの現在のパラメータ設定を表示する方法

mipagentconfig get コマンドを使用して、パラメータ宛先に関連付けられている現在の設定を表示できます。

1. モバイル **IP** を有効にしているシステムでスーパーユーザーになります。
2. コマンド行で、設定値を表示したい各パラメータについて次のコマンドを入力します。

   ```
   # mipagentconfig get [<parameter> | <label>]
   ```

たとえば、le0 デバイスに対する通知設定を表示する場合、次のコマンドを使用します。

```
# mipagentconfig get adv le0
```

このコマンドで表示される出力例を次に示します。

```
[Advertisements le0]
   HomeAgent = yes
   ForeignAgent = yes
```

例 24–9 は、その他のパラメータ宛先に mipagentconfig get コマンドを使用した結果です。

例 **24–9** mipagentconfig get コマンドを使用する

```
# mipagentconfig get MaxClockSkew
      [GlobalSecurityParameters]
         MaxClockSkew=300

# mipagentconfig get HA-FAauth
      [GlobalSecurityParameters]
         HA-FAauth=no

# mipagentconfig get MN-FAauth
      [GlobalSecurityParameters]
         MN-FAauth=no

# mipagentconfig get Challenge
      [GlobalSecurityParameters]
         Challenge=no

# mipagentconfig get Pool 10
      [Pool 10]
         BaseAddress=192.168.1.1
         Size=100

# mipagentconfig get SPI 257
      [SPI 257]
         Key=11111111111111111111111111111111
         ReplayMethod=none

# mipagentconfig get SPI 258
      [SPI 258]
         Key=15111111111111111111111111111111
         ReplayMethod=none

# mipagentconfig get addr 10.1.1.1
      [Address 10.1.1.1]
         SPI=258
         Type=agent

# mipagentconfig get addr 192.168.1.200
      [Address 192.168.1.200]
         SPI=257
         Type=node

# mipagentconfig get addr 10.1.1.1
```

例 **24–9** mipagentconfig get コマンドを使用する　(続き)

```
[Address 10.1.1.1]
   Type=agent
   SPI=258
   IPsecRequest = apply {auth_algs md5 sa shared}
   IPsecReply = permit {auth_algs md5}
   IPsecTunnel = apply {encr_algs 3des sa shared}
```

モビリティエージェント状態の表示

mipagentstat コマンドを使用して外来エージェントのビジターリストおよびホームエージェントの結合テーブルを表示できます。459 ページの「モバイル IP モビリティエージェントの状態」では、mipagentstat コマンドについて説明しています。詳細は、mipagentstat(1M) のマニュアルページでも説明しています。

▼ モビリティエージェント状態を表示する方法

1. モバイル **IP** を有効にしているシステムでスーパーユーザーになります。
2. コマンド行から次のコマンドを入力します。

   ```
   # mipagentstat <option>
   ```

   次のオプションを使用できます。

-f	外来エージェントのビジターリストに稼動中のモバイルノードの一覧を表示する
-h	ホームエージェントの結合テーブルに稼動中のモバイルノードの一覧を表示する
-p	エージェントのモビリティエージェントピアに関するセキュリティアソシエーションの一覧を表示する

たとえば、外来エージェントに登録された全モバイルノードのビジターリストを表示するには、次のコマンドを使用します。

```
# mipagentstat -f
```

このコマンドで表示される出力例を次に示します。

```
Mobile Node     Home Agent     Time (s)     Time (s)  Flags
                               Granted      Remaining
--------------- -------------- ------------ --------- -----
```

```
foobar.xyz.com  ha1.xyz.com    600          125        .....T.
10.1.5.23       10.1.5.1       1000         10         .....T.
```

外来エージェントのセキュリティアソシエーションを表示するには、次のコマンドを使用します。

```
# mipagentstat -p
```

このコマンドで表示される出力例を次に示します。

```
Foreign                  ..... Security Association(s).....
Agent                    Requests Replies  FTunnel  RTunnel
----------------------   -------- -------- -------- --------
forn-agent.eng.sun.com   AH       AH       ESP      ESP
```

ホームエージェントのセキュリティアソシエーションを表示するには、次のコマンドを使用します。

```
# mipagentstat -fp
```

このコマンドで表示される出力例を次に示します。

```
Home                     ..... Security Association(s) .....
Agent                    Requests Replies  FTunnel  RTunnel
----------------------   -------- -------- -------- --------
home-agent.eng.sun.com   AH       AH       ESP      ESP
ha1.xyz.com              AH,ESP   AH       AH,ESP   AH,ESP
```

外来エージェントでのモビリティ経路指定の表示

`netstat` コマンドを使用して、順方向および逆方向トンネルによって作成される発信元固有の経路指定に関する追加情報を表示できます。このコマンドの詳細については、 `netstat`(1M) マニュアルページを参照してください。

▼ 外来エージェントでモビリティ経路指定を表示する方法

1. モバイル **IP** を有効にしているシステムでスーパーユーザーになります。
2. コマンド行から次のコマンドを入力します。

   ```
   # netstat -rn
   ```

次の例は、逆方向トンネルを使用する外来エージェントの経路指定を示します。

```
Routing Table:   IPv4 Source-Specific
Destination      In If     Source      Gateway Flags  Use  Out If
--------------  ------- ------------ --------- -----  ---- -------
10.6.32.11      ip.tun1      --      10.6.32.97  UH      0 hme1
    --          hme1    10.6.32.11       --      U       0 ip.tun1
```

最初の行は、宛先 IP アドレス 10.6.32.11 と着信インタフェース ip.tun1 がパケットを転送するインタフェースとして hme1 を選択していることを表します。次の行は、インタフェース hme1 から発信する任意のパケットと発信元アドレス 10.6.32.11 が ip.tun1 に転送されることを表しています。 これは、逆方向トンネル経路指定の一例です。

第 25 章

モバイル IP のファイルおよびコマンド (リファレンス)

この章では、モバイル IP の Solaris 実装に提供される構成要素について説明します。モバイル IP を使用するには、最初に以下の節で説明されるパラメータとコマンドを使用して、モバイル IP 構成ファイルを構成する必要があります。

この章では、以下の内容について説明します。

Solaris モバイル IP 実装の概要

モビリティエージェントソフトウェアにはホームエージェントと外来エージェントの機能が組み込まれています。Solaris モバイル IP ソフトウェアではクライアントモバイルノードを提供していません。エージェント機能だけが提供されています。モビリティサポートのある各ネットワークは、このソフトウェアを実行している静的な (非モバイル) ホストを 1 つ以上持たなければなりません。次に示す RFC 機能がモバイル IP の Solaris 実装でサポートされています。

RFC 1918　専用インターネットのアドレスの割り当て

RFC 2002　(エージェントのみ) IP モビリティサポート

RFC 2003　IP 内 IP カプセル化

RFC 2794　IPv4 用モバイル IP ネットワークアクセス識別子拡張

RFC 3012　モバイル IP チャレンジ/レスポンス拡張

RFC 3024　　モバイル IP 用逆トンネリング

基本モバイル IP プロトコル (RFC 2002) は、スケーラブルな鍵配布の問題を取り扱わず、鍵の配布として扱っています。Solaris モバイル IP ソフトウェアは、構成ファイルに指定された、手動で構成された鍵のみを使用します。

次に示す IETF ドラフトにある機能も、モバイル IP の Solaris 実装でサポートされています。

- **draft-ietf-mobileip-rfc2002–bis-03.txt** – 更新された、IPv4 の IP モビリティサポート
- **draft-ietf-mobileip-vendor-ext-09.txt** – モバイル IP ベンダー/組織固有の拡張

次の RFC 機能は、モバイル IP の Solaris 実装でサポートされていません。

RFC 1700　　一般経路指定カプセル化

RFC 1701　　一般経路指定カプセル化

RFC 2004　　IP 内最小カプセル化

次の機能は、モバイル IP の Solaris 実装でサポートされていません。

- ホームエージェントによる、マルチキャストトラフィックまたはブロードキャストトラフィックの外部ネットワークにアクセスしているモバイルノードの外来エージェントへの転送
- 逆方向トンネルを経由するブロードキャストデータグラムマルチキャストデータグラムの経路指定
- 専用気付アドレス、または専用ホームエージェントアドレス

詳細については、`mipagent(1M)` マニュアルページを参照してください。

モバイル IP 構成ファイル

`mipagent` コマンドは、起動時に `/etc/inet/mipagent.conf` 構成ファイルから構成情報を読み取ります。 モバイル IP は `/etc/inet/mipagent.conf` 構成ファイルを使用してモバイル IP モビリティエージェントを初期化します。構成および配置されると、モビリティエージェントは定期的なルーター通知を発行し、ルーター発見要請メッセージおよびモバイル IP 登録メッセージに応答します。

ファイルの属性については、`mipagent.conf(4)` のマニュアルページ、ファイルの使用法については、`mipagent(1M)` のマニュアルページを参照してください。

構成ファイルの形式

モバイル IP 構成ファイルはセクションにより構成されています。各セクションは固有の名前を持っていて、角括弧で囲まれています。各セクションには 1 つ以上のラベルが付いています。ラベルに値を設定するには次の形式を用います。

```
[Section_name]
      Label-name = Value-assigned
```

449 ページの「構成ファイルのセクションとラベル」では、セクション名、ラベル、および可能な設定値を説明しています。

構成ファイルの例

Solaris のデフォルトのインストールでは、次の構成ファイルのサンプルが /etc/inet ディレクトリにあります。

- mipagent.conf-sample – 外来エージェントおよびホームエージェントの両機能を提供するモバイル IP エージェント用のサンプル構成ファイル
- mipagent.conf.fa-sample – 外来エージェント機能のみを提供するモバイル IP エージェント用のサンプル構成ファイル
- mipagent.conf.ha-sample – ホームエージェント機能のみを提供するモバイル IP エージェント用のサンプル構成ファイル

これらのサンプル構成ファイルには、モバイルノードアドレスおよびセキュリティ設定の例が記載されています。モバイル IP を実装する前に、mipagent.conf という構成ファイルを作成して /etc/inet ディレクトリに格納しなければなりません。このファイルには、ユーザーのモバイル IP 実装の要件を満たす値を指定します。サンプル構成ファイルの 1 つを選択し、ユーザーのアドレスおよびセキュリティ設定で変更して、/etc/inet/mipagent.conf にコピーすることもできます。

実行方法については、429 ページの「モバイル IP 構成ファイルを作成する方法」を参照してください。

mipagent.conf-sample ファイル

次に mipagent.conf-sample ファイルに指定されたセクション名、ラベル、および設定値を示します。449 ページの「構成ファイルのセクションとラベル」では、構文、セクション、ラベル、および設定値について説明しています。

```
[General]
   Version = 1.0    # version number for the configuration file. (required)

[Advertisements hme0]
   HomeAgent = yes
   ForeignAgent = yes
```

```
   PrefixFlags = yes
   AdvertiseOnBcast = yes
   RegLifetime = 200
   AdvLifetime = 200
   AdvFrequency = 5
   ReverseTunnel = no
   ReverseTunnelRequired = no

[GlobalSecurityParameters]
   MaxClockSkew = 300
   HA-FAauth = yes
   MN-FAauth = yes
   Challenge = no
   KeyDistribution = files

[Pool 1]
   BaseAddress = 10.68.30.7
   Size = 4

[SPI 257]
   ReplayMethod = none
   Key = 11111111111111111111111111111111

[SPI 258]
   ReplayMethod = none
   Key = 15111111111111111111111111111111

[Address 10.1.1.1]
   Type = node
   SPI = 258

[Address mobilenode@sun.com]
   Type = node
   SPI = 257
   Pool = 1

[Address Node-Default]
   Type = node
   SPI = 258
   Pool = 1

[Address 10.68.30.36]
   Type = agent
   SPI = 257
[Address 10.68.30.36]
   Type = agent
   SPI = 257
   IPsecRequest = apply {auth_algs md5 sa shared}
   IPsecReply = permit {auth_algs md5}
   IPsecTunnel =  apply {encr_algs 3des sa shared}
```

例 – mipagent.conf.fa-sample ファイル

次に mipagent.conf.fa-sample ファイルに指定されたセクション名、ラベル、および設定値を示します。449 ページの「構成ファイルのセクションとラベル」では、構文、セクション、ラベル、および設定値について説明しています。mipagent.conf.fa-sample ファイルは、外来エージェント機能のみを提供する構成を示しています。サンプルファイルには Pool セクションがありません。Pool はホームエージェントのみが利用するからです。その他の点では、このファイルは mipagent.conf-sample ファイルと同じです。

```
[General]
   Version = 1.0    # version number for the configuration file. (required)

[Advertisements hme0]
   HomeAgent = no
   ForeignAgent = yes
   PrefixFlags = yes
   AdvertiseOnBcast = yes
   RegLifetime = 200
   AdvLifetime = 200
   AdvFrequency = 5
   ReverseTunnel = yes
   ReverseTunnelRequired = no

[GlobalSecurityParameters]
   MaxClockSkew = 300
   HA-FAauth = yes
   MN-FAauth = yes
   Challenge = no
   KeyDistribution = files

[SPI 257]
   ReplayMethod = none
   Key = 11111111111111111111111111111111

[SPI 258]
   ReplayMethod = none
   Key = 15111111111111111111111111111111

[Address 10.1.1.1]
   Type = node
   SPI = 258

[Address 10.68.30.36]
   Type = agent
   SPI = 257

[Address 10.68.30.36]
   Type = agent
   SPI = 257
   IPsecRequest = apply {auth_algs md5 sa shared}
   IPsecReply = permit {auth_algs md5}
   IPsecTunnel = apply {encr_algs 3des sa shared}
```

例 – mipagent.conf.ha-sample ファイル

次に mipagent.conf.ha-sample ファイルに指定されたセクション名、ラベル、および設定値を示します。449 ページの「構成ファイルのセクションとラベル」では、構文、セクション、ラベル、および設定値について説明しています。

mipagent.conf.ha-sample ファイルは、ホームエージェント機能のみを提供する構成を示しています。その他の点では、このファイルは mipagent.conf-sample ファイルと同じです。

```
[General]
   Version = 1.0    # version number for the configuration file. (required)

[Advertisements hme0]
   HomeAgent = yes
   ForeignAgent = no
   PrefixFlags = yes
   AdvertiseOnBcast = yes
   RegLifetime = 200
   AdvLifetime = 200
   AdvFrequency = 5
   ReverseTunnel = yes
   ReverseTunnelRequired = no

[GlobalSecurityParameters]
   MaxClockSkew = 300
   HA-FAauth = yes
   MN-FAauth = yes
   Challenge = no
   KeyDistribution = files

[Pool 1]
   BaseAddress = 10.68.30.7
   Size = 4

[SPI 257]
   ReplayMethod = none
   Key = 11111111111111111111111111111111

[SPI 258]
   ReplayMethod = none
   Key = 15111111111111111111111111111111

[Address 10.1.1.1]
   Type = node
   SPI = 258

[Address mobilenode@sun.com]
   Type = node
   SPI = 257
   Pool = 1

[Address Node-Default]
   Type = node
```

```
   SPI = 258
   Pool = 1

[Address 10.68.30.36]
    Type = agent
    SPI = 257
    IPsecRequest = apply {auth_algs md5 sa shared}
    IPsecReply = permit {auth_algs md5}
    IPsecTunnel = apply {encr_algs 3des sa shared}
```

構成ファイルのセクションとラベル

モバイル IP 構成ファイルには、次のセクションがあります。

- General (必須)
- Advertisements (必須)
- GlobalSecurityParameters (省略可能)
- Pool (省略可能)
- SPI (省略可能)
- Address (省略可能)

General および GlobalSecurityParameters セクションは、モバイル IP エージェントの動作に関する情報を含み、構成ファイル内に 1 つだけ指定できます。

General セクション

General セクションは、1 つのラベル、つまり構成ファイルのバージョン番号だけが含まれます。General セクションの構文は次のとおりです。General セクションの構文は次のとおりです。

```
[General]
     Version = 1.0
```

Advertisements セクション

Advertisements セクションには、HomeAgent、ForeignAgent などのラベルが含まれます。モバイル IP サービスを提供するローカルホストの各インタフェースには、それぞれ異なる Advertisements セクションを指定しなければなりません。Advertisements セクションの構文は次のとおりです。

```
[Advertisements Interface-name]
     HomeAgent = <yes/no>
     ForeignAgent = <yes/no>
     .
     .
```

通常、システムは 1 つのインタフェース (le0、hme0 など) を持ち、ホームエージェントおよび外来エージェントの両方の動作をサポートします。たとえば hme0 の場合、yes が HomeAgent および ForeignAgent の両ラベルに次のように指定されます。

```
[Advertisements hme0]
     HomeAgent = yes
     ForeignAgent = yes
     .
     .
```

動的インタフェースによる通知の場合、デバイス ID 部分に * を使用します。たとえば、 *Interface-name* ppp* は、mipagent の開始後に構成されるすべての ppp インタフェースを含むことを意味します。動的インタフェースタイプの advertisement セクションにあるすべての属性は、同じ状態にします。

表 25–1 で、Advertisements セクションに指定可能なラベルと設定値について説明します。

表 25–1 Advertisements セクションのラベルと設定値

ラベル	値	説明
HomeAgent	yes または no	mipagent がホームエージェント機能を提供するかどうかを指定する
ForeignAgent	yes または no	mipagent が外来エージェント機能を提供するかどうかを指定する
PrefixFlags	yes または no	通知が任意の接頭辞拡張を含むかどうかを指定する
AdvertiseOnBcast	yes または no	設定値が yes の場合、通知は 224.0.0.1 ではなく 255.255.255.255 に送信される
RegLifetime	n	登録要求で受け付けた、秒単位の最長有効期間
AdvLifetime	n	通知がそれ以上ない場合に現在の通知が有効と考えられる、秒単位の最大時間
AdvFrequency	n	2 つの連続した通知間の、秒単位の時間

表 **25–1** Advertisements セクションのラベルと設定値　(続き)

ラベル	値	説明
ReverseTunnel	yes、no、FA、HA、both のいずれか	mipagent が逆方向トンネル機能を要求するかどうかを指定する。 設定値が yes の場合、外来エージェントとホームエージェントの両方が逆方向トンネリングをサポートする。設定値が no の場合、インタフェースは逆方向トンネリングをサポートしない 設定値が FA の場合、外来エージェントが逆方向トンネリングをサポートする。設定値が HA の場合、ホームエージェントが逆方向トンネリングをサポートする。設定値が both の場合、外来エージェントとホームエージェントの両方が逆方向トンネリングをサポートする。
ReverseTunnelRequired	yes、no、FA、HA のいずれか	mipagent が逆方向トンネル機能を要求するかどうかを指定する。したがって、モバイルノードが逆方向トンネルを登録中に要求すべきかどうかを指定する 設定値が yes の場合、外来エージェントとホームエージェントの両方が逆方向トンネルを要求する。設定値が no の場合、インタフェースは逆方向トンネルを要求しない 設定値が FA の場合、外来エージェントが逆方向トンネリングを要求する。設定値が HA の場合、ホームエージェントが逆方向トンネリングを要求する
AdvInitCount	n	要請しない通知の初期値を指定する。デフォルト値は 1。この値は、AdvLimitUnsolicited が yes の場合に有効
AdvLimitUnsolicited	yes または no	モビリティインタフェースによる、要請されない通知の制限値を有効または無効にする

GlobalSecurityParameters セクション

GlobalSecurityParameters セクションには、maxClockSkew、HA-FAauth、MN-FAauth、Challenge、および KeyDistribution ラベルが含まれます。このセクションではセキュリティパラメータを定義します。GlobalSecurityParameters セクションの構文は次のとおりです。

```
[GlobalSecurityParameters]
     MaxClockSkew = n
     HA-FAauth = <yes/no>
```

```
MN-FAauth = <yes/no>
Challenge = <yes/no>
KeyDistribution = files
```

モバイル IP プロトコルは、タイムスタンプをメッセージ内に含めることで、メッセージの再実行に対する保護を提供します。クロックが異なる場合、ホームエージェントは現在時間とともにエラーをモバイルノードに返します。モバイルノードはその現在時間を使って再登録できます。モバイルノードはその現在時間を使って再登録できます。MaxClockSkew ラベルを使用して、ホームエージェントとモバイルノードのクロック間で異なる最大秒数を構成することができます。デフォルト値は 300 秒です。

HA-FAauth および MN-FAauth ラベルは、それぞれホームと外来間、およびモバイルと外来間の認証に関する条件を有効または無効にします。デフォルトは無効です。外来エージェントが通知内に指定されたモバイルノードへ呼び出しを発行するようにするためには、challenge ラベルを使用します。このラベルは再実行に対する保護のために使用します。デフォルト値は無効です。

表 25–2 で、GlobalSecurityParameters セクションに指定可能なラベルと設定値について説明します。

表 **25–2** GlobalSecurityParameters セクションのラベルと設定値

ラベル	値	説明
MaxClockSkew	n	mipagent が自分のローカル時間と登録要求に示された時間の差として受け入れる秒数
HA-FAauth	yes または no	HA-FA 認証拡張が、登録要求と応答に存在する必要があるかを指定する
MN-FAauth	yes または no	MN-FA 認証拡張が、登録要求と応答に存在する必要があるかどうかを指定する
Challenge	yes または no	外来エージェントが自分のモビリティ通知内に呼び出しを含むかどうかを指定する
KeyDistribution	files	常に files に設定

Pool セクション

モバイルノードには、ホームエージェントによって動的アドレスを割り当てることができます。動的アドレスの割り当ては、DHCP とは独立に mipagent が行います。ユーザーは、ホームアドレスを要求することによってモバイルノードが使用できるアドレスプールを作成できます。アドレスプールは、構成ファイルの Pool セクションを使って構成されます。

Pool セクションには、BaseAddress および Size ラベルが含まれます。Pool セクションの構文は次のとおりです。

```
[Pool Pool-identifier]
     BaseAddress = IP-address
     Size = size
```

注 – Pool 識別子を使用している場合、モバイルノードの Address セクションにも存在していなければなりません。

Pool セクションを使用してモバイルノードに割り当て可能なアドレスプールを定義します。BaseAddress ラベルは、プール内の最初の IP アドレスを設定するのに使用します。Size は、プール内の使用可能なアドレス数を指定するのに使用します。

たとえば、IP アドレスの 192.168.1.1 から 192.168.1.100 が Pool 10 に予約されている場合、Pool セクションには次の項目を指定します。

```
[Pool 10]
     BaseAddress = 192.168.1.1
     Size = 100
```

注 – アドレスの範囲にブロードキャストアドレスは含まないでください。たとえば、BaseAddress = 192.168.1.200、Size = 60 のように割り当てないでください。このアドレス範囲にはブロードキャストアドレスの 192.168.1.255 が含まれているからです。

表 25–3 で、Pool セクションに指定可能なラベルと設定値について説明します。

表 **25–3** Pool セクションのラベルと設定値

ラベル	値	説明
BaseAddress	n.n.n.n	アドレスプール内の最初のアドレス
Size	n	プール内のアドレス数

SPI セクション

モバイル IP プロトコルはメッセージ認証を要求するので、セキュリティパラメータインデックス (SPI) を使用してセキュリティコンテキストを特定しなければなりません。セキュリティコンテキストは SPI セクションに定義します。定義したセキュリティコンテキストそれぞれに異なる SPI セクションを指定しなければなりません。ID 番号がセキュリティコンテキストを特定します。モバイル IP プロトコルは、最初の 256 SPI を予約しています。したがって、256 より大きい SPI 値を使用してください。SPI セクションには、共有された秘密情報や再実行保護などのセキュリティに関連した情報が含まれています。

SPI セクションにはまた、ReplayMethod および Key ラベルがあります。このセクションではセキュリティコンテキストを定義します。SPI セクションの構文は次のとおりです。

```
[SPI SPI-identifier]
     ReplayMethod = <none/timestamps>
     Key = key
```

2 つの通信中のピアは、同じ SPI 識別子を共有しなければなりません。ユーザーはそれらを同じ鍵と再実行メソッドで構成しなければなりません。鍵は 16 進数の文字列で指定します。最大長は 16 バイトです。たとえば、鍵の長さが 16 バイトで 16 進数値の `0` から `f` を含んでいる場合、鍵は次のようになります。

```
Key = 0102030405060708090a0b0c0d0e0f10
```

鍵は、偶数の桁 (1 バイト 2 桁の表示法に対応) を持たなければなりません。

表 25–4 で、`SPI` セクションに指定可能なラベルと設定値について説明します。

表 25–4 `SPI` セクションのラベルと設定値

ラベル	値	説明
`ReplayMethod`	`none` または `timestamps`	SPI 用の再実行認証の種類を指定する
`Key`	`x`	16 進表示の認証キー

`Address` セクション

モバイル IP の Solaris 実装では、3 つの方法の 1 つを使ってモバイルノードを構成できます。各方法は `Address` セクションで構成されます。最初の方法は、従来のモバイル IP プロトコルに従い、各モバイルノードがホームアドレスを持つことを要求します。第 2 の方法では、モバイルノードをネットワークアクセス識別子 (NAI) を使って特定することが可能になります。最後の方法では、ユーザーは「デフォルト」のモバイルノードを構成できます。このデフォルトモバイルノードは、適当な SPI 値および関連する鍵情報を持っているどのモバイルノードでも利用できます。

モバイルノード

モバイルノード用の `Address` セクションには、アドレスタイプと SPI 識別子を定義した `Type` および `SPI` ラベルが含まれます。`Address` セクションの構文は次のとおりです。

```
[Address address]
     Type = node
     SPI = SPI-identifier
```

サポートされた各モバイルノードに対して `Address` セクションをホームエージェントの構成ファイル内に指定しなければなりません。

モバイル IP メッセージ認証が外来エージェントおよびホームエージェント間で必要な場合は、エージェントが通信する必要のある各ピアに対して `Address` セクションを指定しなければなりません。

構成した SPI 値は、構成ファイルに存在する SPI セクションを示さなければなりません。

また、モバイルノード用の専用アドレスを構成することもできます。

表 25–8 で、デフォルトモバイルノード用の Address セクションに指定可能なラベルと設定値について説明します。

表 **25–5** Address セクションのラベルと設定値 — モバイルノード

ラベル	値	説明
Type	node	この項目がモバイルノード用であることを指定する
SPI	n	関連する項目用の SPI 値を指定する

モビリティエージェント

モビリティエージェント用の Address セクションには、アドレスタイプと SPI 識別子を定義した Type および SPI ラベルが含まれます。この節では、IPsec 要求、応答、トンネルラベルについても説明します。 Address セクションの構文は次のとおりです。

```
[Address address]
     Type = agent
     SPI = SPI-identifier
     IPsecRequest = action {properties} [: action {properties}]
     IPsecReply = action {properties} [: action {properties}]
     IPsecTunnel = action {properties} [: action {properties}]
```

サポートされた各モビリティエージェントに対して Address セクションをホームエージェントの構成ファイル内に指定しなければなりません。

モバイル IP メッセージ認証が外来エージェントおよびホームエージェント間で必要な場合は、エージェントが通信する必要のある各ピアに対して Address セクションを指定しなければなりません。

構成した SPI 値は、構成ファイルに存在する SPI セクションを示さなければなりません。

次の表で、モビリティエージェント用の Address セクションに指定可能なラベルと設定値について説明します。

表 **25–6** Address セクションのラベルと設定値 — モビリティエージェント

ラベル	値	説明
Type	agent	この項目がモビリティエージェント用であることを指定する

表 **25–6** Address セクションのラベルと設定値 — モビリティエージェント　(続き)

ラベル	値	説明
SPI	n	関連する項目用の SPI 値を指定する
IPsecRequest	apply または permit (次の注を参照)	このモビリティエージェントのピアとの間の登録要求に対して呼び出す IPsec プロパティ
IPsecReply	apply または permit (次の注を参照)	このモビリティエージェントのピアとの間の登録要求に対して呼び出す IPsec プロパティ
IPsecTunnel	apply または permit (次の注を参照)	このモビリティエージェントとの間のトンネルトラフィックに対して呼び出す IPsec プロパティ

注 – apply の値は、出力データグラムに対応します。permit の値は、入力データグラムに対応します。したがって、IPsecRequest apply の値と IPsecReply permit の値は、登録データグラムを送受信するのに外来エージェントで使用されます。また、IPsecRequest permit の値と IPsecReply apply の値は、登録データグラムを送受信するのにホームエージェントでも使用されます。

自分の *NAI* で識別されるモバイルノード

自分の NAI で識別されるモバイルノード用の Address セクションには、Type、SPI、および Pool ラベルが含まれます。NAI パラメータがあるため、NAI によるモバイルノードの識別が可能になります。NAI パラメータを使用した Address セクションの構文は次のとおりです。

```
[Address NAI]
     Type = Node
     SPI = SPI-identifier
     Pool = Pool-identifier
```

プールを利用するには、NAI 経由でモバイルノードを特定します。Address セクションでは、ホームアドレスの場合と異なり NAI を構成できます。NAI には、user@domain の形式を使用します。ホームアドレスをモバイルノードに割り当てるためにどのアドレスプールを使用するかを指定するには、Pool ラベルを使用します。

表 25–7 で、自分の NAI で識別されるモバイルノード用の Address セクションに指定可能なラベルと設定値について説明します。

表 **25–7** Address セクションのラベルと設定値 — 自分の NAI で識別されるモバイルノード

ラベル	値	説明
Type	node	この項目がモバイルノード用であることを指定する
SPI	n	関連する項目用の SPI 値を指定する
Pool	n	モバイルノードに割り当てるアドレスのプールを割り当てる

図 25–1 に示すように、NAI で識別されたモバイルノードを指定した Address セクションに定義された SPI および Pool ラベルに対して、ユーザーは対応する SPI および Pool セクションを持たなければなりません。

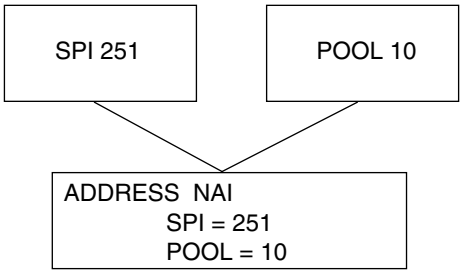


図 **25–1** 自分の NAI で識別されたモバイルノードを指定した Address セクションに対応する SPI および Pool

デフォルトのモバイルノード

デフォルトのモバイルノード用の Address セクションには、Type、SPI、および Pool ラベルが含まれます。Node-Default パラメータがあるため、(このセクションで定義された) 正しい SPI を持っている場合は、すべてのモバイルノードがサービスを受けられるようになります。Node-Default パラメータを使用した Address セクションの構文は次のとおりです。

```
[Address Node-Default]
     Type = Node
     SPI = SPI-identifier
     Pool = Pool-identifier
```

Node-Default パラメータがあるため、構成ファイルのサイズを縮小することが可能になります。その他の方法では、各モバイルノードには独自のセクションが必要です。ただし、Node-Default はセキュリティに影響します。何かの理由でモバイルノードが信用できなくなった場合、すべての信頼のおけるモバイルノードに関するセキュリティ情報を更新する必要があります。この作業は手間がかかります。しかし、セキュリティがあまり重要でないネットワークでは Node-Default を利用できます。

表 25–8 で、デフォルトモバイルノード用の Address セクションに指定可能なラベルと設定値について説明します。

表 **25–8** Address セクションのラベルと設定値 — デフォルトモバイルノード

ラベル	値	説明
Type	node	この項目がモバイルノード用であることを指定する
SPI	n	関連する項目用の SPI 値を指定する
Pool	n	モバイルノードに割り当てるアドレスのプールを割り当てる

図 25–2 に示すように、デフォルトモバイルノードを指定した Address セクションに定義された SPI および Pool ラベルに対して、対応する SPI および Pool セクションを持たなければなりません。

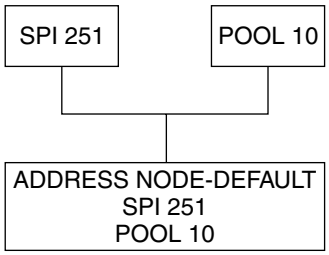


図 **25–2** デフォルトモバイルノードを指定した Address セクションに対応する SPI および Pool セクション

モビリティ IP エージェントの構成

mipagentconfig コマンドを使用してモビリティエージェントを構成できます。また、/etc/inet/mipagent.conf 構成ファイル内のどのような パラメータも作成または変更できます。特に、設定値の変更や、モビリティクライアント、プール、および SPI の追加および削除ができます。mipagentconfig コマンドは、次の形式になります。

```
# mipagentconfig <command> <parameter> <value>
```

表 25–9 に、/etc/inet/mipagent.conf 構成ファイルにパラメータを作成または変更するために、mipagentconfig で利用できるコマンドを示します。

表 **25–9** mipagentconfig コマンド

コマンド名	説明
add	通知パラメータ、セキュリティパラメータ、SPI、およびアドレスを構成ファイルに追加するために使用する
change	構成ファイル内の通知パラメータ、セキュリティパラメータ、SPI、およびアドレスを変更するために使用する
delete	構成ファイル内の通知パラメータ、セキュリティパラメータ、SPI、およびアドレスを削除するために使用する
get	構成ファイル内の現在の設定を表示するのに使用する

コマンドパラメータおよび許容できる設定値については、mipagentconfig(1M) マニュアルページを参照してください。432 ページの「モバイル IP 構成ファイルの変更」では、mipagentconfig コマンドの利用方法について説明しています。

モバイル IP モビリティエージェントの状態

mipagentstat コマンドを使用して、外来エージェントのビジターリストおよびホームエージェントの結合テーブルを表示できます。また、エージェントのモビリティエージェントのピアに関連するセキュリティを表示することもできます。外来エージェントのビジターリストを表示するには、mipagentstat コマンドの -f オプションを使用します。ホームエージェントの結合テーブルを表示するには、mipagentstat コマンドの -h オプションを使用します。エージェントのモビリティエージェントのピアに関連するセキュリティを表示するには、mipagentstat コマンドの -p オプションを使用します。次の例では、これらのオプションを使用した場合の出力例を示します。

例 **25–1** 外来エージェントのビジターリスト

```
Mobile Node     Home Agent     Time (s)     Time (s)  Flags
                               Granted      Remaining
--------------- -------------- ------------ --------- -----
foobar.xyz.com  ha1.xyz.com    600          125       .....T.
10.1.5.23       10.1.5.1       1000         10        .....T.
```

例 **25–2** ホームエージェントの結合テーブル

```
Mobile Node     Home Agent     Time (s)     Time (s)  Flags
                               Granted      Remaining
--------------- -------------- ------------ --------- -----
foobar.xyz.com  fa1.tuv.com    600          125       .....T.
```

例 **25–2** ホームエージェントの結合テーブル　(続き)

```
10.1.5.23       123.2.5.12     1000          10         .....T.
```

例 **25–3** モビリティエージェントのピアのセキュリティアソシエーションテーブル

```
Foreign                    ..... Security Association(s).....
Agent                      Requests Replies  FTunnel  RTunnel
-----------------------    -------- -------- -------- --------
forn-agent.eng.sun.com     AH       AH       ESP      ESP

Home                       ..... Security Association(s) .....
Agent                      Requests Replies  FTunnel  RTunnel
-----------------------    -------- -------- -------- --------
home-agent.eng.sun.com     AH       AH       ESP      ESP
ha1.xyz.com                AH,ESP   AH       AH,ESP   AH,ESP
```

コマンドのオプションの詳細については、mipagentstat(1M) マニュアルページを参照してください。 439 ページの「モビリティエージェント状態の表示」では、mipagentstat コマンドを使用する手順を説明しています。

モバイル IP の状態情報

mipagent デーモンは、シャットダウン時に状態情報を /var/inet/mipagent_state に格納します。これは、mipagent がホームエージェントとしてサービスを提供している場合です。この状態情報には、ホームエージェントとしてサポートされているモバイルノードのリスト、それらのノードの現在の気付アドレス、および残りの有効期間が含まれます。また、モビリティエージェントのピアに関するセキュリティアソシエーション構成も含まれます。mipagent プログラムを (保守のために) 終了して再起動すると、モビリティエージェントの内部状態をできるだけ再現するために mipagent_state が使用されます。このようにして、モバイルノードが他のネットワークにいる場合でも、サービスの中断を最小限に抑えます。mipagent_state が存在していれば、mipagent が起動または再起動されるたびに mipagent.conf の直後に読み込まれます。

モバイル IP 用の netstat 拡張

モバイル IP の転送先経路指定を特定するために、モバイル IP 用の拡張が netstat (1M) コマンドに追加されています。 netstat(1M) コマンドを使用して、「Source-Specific」と呼ばれる新しい経路指定テーブルを表示できます。詳細については、netstat(1M) マニュアルページを参照してください。

次の例は、-nr フラグを使用した場合の netstat コマンドの出力を示します。

例 **25–4** netstat コマンドの出力

```
Routing Table:   IPv4 Source-Specific
Destination      In If     Source      Gateway Flags  Use  Out If
--------------  ------- ------------ --------- -----  ---- -------
10.6.32.11      ip.tun1      --      10.6.32.97  UH       0 hme1
    --          hme1    10.6.32.11       --      U        0 ip.tun1
```

この例は、逆方向トンネルを使用する外来エージェントの経路指定を示します。最初の行は、宛先 IP アドレス 10.6.32.11 と着信インタフェース ip.tun1 がパケットを転送するインタフェースとして hme1 を選択していることを表します。次の行は、インタフェース hme1 から発信する任意のパケットと発信元アドレス 10.6.32.11 が ip.tun1 に転送されることを表しています。

モバイル IP 用の snoop 拡張

リンク上のモバイル IP トラフィックを特定するために、モバイル IP 拡張が snoop (1M) コマンドに追加されました。詳細については、snoop(1M) マニュアルページを参照してください。

例 25–5 は、モバイルノードの mip-mn2 上で実行中の snoop の出力を示します。

例 **25–5** snoop コマンドの出力

```
mip-mn2# snoop
Using device /dev/hme (promiscuous mode)
  mip-fa2 -> 224.0.0.1    ICMP Router advertisement (Lifetime 200s [1]:
{mip-fa2-80 2147483648}), (Mobility Agent Extension), (Prefix Lengths),
(Padding)
  mip-mn2 -> mip-fa2   Mobile IP reg rqst
  mip-fa2 -> mip-mn2   Mobile IP reg reply (OK code 0)
```

この例は、モバイルノードが外来エージェントの mip-fa2 から定期的に送信されたモビリティエージェント通知の 1 つを受信したことを示しています。その後、mip-mn2 が登録要求を mip-fa2 に送信し、その応答として登録応答を受信しています。登録応答は、モバイルノードが自分のホームエージェントに正常に登録されたことを示しています。

snoop(1M) コマンドは、IPsec 用の拡張もサポートしています。そのため、登録とトンネルパケットを保護する方法を表示できます。

第 26 章

IP ネットワークマルチパス (トピック)

第 27 章

IP ネットワークマルチパス (概要)

同じ IP リンク (たとえば Ethernet) に複数のネットワークインタフェースカードを接続していれば、IP ネットワークマルチパスによって負荷分散と障害経路の迂回がサポートされます。

この章では、以下の内容について説明します。

はじめに

IP ネットワークマルチパスには次の機能があります。

- ネットワークアダプタの単一点障害の回避
- 単位時間当たりのデータの流量の向上

ネットワークアダプタに障害が発生した場合、同じ IP リンクに別のアダプタが接続されていれば、すべてのネットワークアクセスは、障害の発生したアダプタからこのアダプタに自動的に切り替えられます。このプロセスにより、ネットワークへのアクセ

スは中断することなく継続されます。また、同じ IP リンクに複数のネットワークアダプタが接続されている場合、トラフィックを複数のネットワークアダプタに分散させることにより、トラフィックのスループットが向上します。

注 – RFC 2460 など、IP 関連の他の文書では、「IP リンク」の代わりに「リンク」という用語が使用されています。このマニュアルでは、IEEE 802 との混同を避けるため「IP リンク」を使用します。IEEE 802 では、「リンク」は Ethernet NIC から Ethernet スイッチへの 1 本のワイヤを意味します。

IP リンクの説明は、用語集または表 27–1 を参照してください。

IP ネットワークマルチパスの機能

Solaris の IP ネットワークマルチパスには、次の機能があります。

- 障害検出 – ネットワークアダプタの障害を自動的に検出し、ネットワークアクセスを別のネットワークアダプタに自動的に切り替えます (障害経路の迂回)。ただし、別のネットワークアダプタが事前に構成されていなければなりません。詳細については、468 ページの「物理インタフェース障害の検出」を参照してください。
- 回復検出 – 障害の発生したネットワークアダプタが回復したことを検出し、別のネットワークアダプタで行われていたネットワークアクセスを、自動的に元に戻します (回復した経路への復帰)。ただし、回復した経路への復帰が事前に有効になっていなければなりません。詳細については、470 ページの「物理インタフェースの回復検出」を参照してください。
- 送信負荷分散 – 送信ネットワークパケットをパケットの順序を変えずに複数のネットワークアダプタに分散し、単位時間当たりのデータの流量を向上させます。ただし、負荷分散が行われるのは、データが複数の接続を経由して複数の標識に送信される場合だけです。

通信障害

通信の障害は次の場合に起こります。

1. NIC の送受信パスがパケット送信を停止した。
2. NIC からリンクへの接続が切れた
3. Ethernet スイッチ上のポートがパケットを送受信しない。
4. グループ内の物理インタフェースがシステムの起動時に存在しない。

5. 相手方のホストが応答しないか、パケットを転送するルーターが応答しない

Solaris の IP ネットワークマルチパスでは、上記 1 ～ 4 の通信障害に対処します。

IP ネットワークマルチパスの構成要素

表 27–1 に、IP ネットワークマルチパスの構成要素を示します。

表 27–1 IP ネットワークマルチパスの構成要素

コンポーネント	説明
IP リンク	リンク層でノード間の通信に使用される通信設備や通信媒体。リンク層とは IPv4 および IPv6 のすぐ下の層で、例としては、Ethernet (単一の、またはブリッジされた) または ATM ネットワークがある。IP リンクには、1 つまたは複数の IPv4 サブネット番号 (ネットワーク接頭子) が割り当てられる。同じサブネット番号 (ネットワーク接頭子) を複数の IP リンクに割り当てることはできない。ATM LANE では、IP リンクは 1 つのエミュレートされた LAN である。ARP を使用する場合、ARP プロトコルの有効範囲は単一の IP リンクである
ネットワークインタフェースカード (NIC)	リンクとのインタフェースになる、内部ネットワークアダプタまたは独立したネットワークアダプタカード。
物理インタフェース	リンクに対するノードの接続。この接続は通常、デバイスドライバとネットワークアダプタとして実装される。ネットワークアダプタによっては、qfe のように複数の接続点を持つ場合もある。このマニュアルでは、「ネットワークアダプタ」は「単一接続点」を示す。"
物理インタフェースグループ	同じリンクに接続されている、システムの物理インタフェース群。グループ内のすべての物理インタフェースには、識別のための空文字列でない同じ名前が割り当てられる
物理インタフェースグループ名	グループを識別する、物理インタフェースに割り当てられる名前。この名前の有効範囲は 1 つのシステム。同じグループ名を共有する複数の物理インタフェースは、物理インタフェースグループを構成する
障害検出	NIC や NIC から第 3 層の装置への経路が動作しなくなったことを検出する処理
回復検出	障害の発生後、NIC や NIC から第 3 層の装置への経路が、正しく動作し始めたことを検出する処理

表 27–1 IP ネットワークマルチパスの構成要素　(続き)

コンポーネント	説明
障害経路の迂回または回復復帰	ネットワークアクセスを障害が検出されたインタフェースから正常な物理インタフェースに切り替える処理。ネットワークアクセスには、IPv4 のユニキャスト、マルチキャスト、およびブロードキャストと、IPv6 のユニキャストとマルチキャストが含まれる
回復した経路への復帰または回復復帰	ネットワークアクセスを、回復が検出されたインタフェースに戻す処理
待機インタフェース	グループ内の他の物理インタフェースに障害が発生するまでデータの伝送には使用されない物理インタフェース

Solaris ネットワークマルチパス

Solaris ネットワークマルチパスは、次の構成要素で実装されています。

- マルチパスデーモン – `in.mpathd`(1M)
- `ip`(7P)

`in.mpathd` デーモンは障害を検出し、障害経路の迂回や回復した経路への復帰に対するさまざまな方針を実装します。`in.mpathd` は障害や回復を検出すると、ioctl を発行して障害経路の迂回や回復した経路への復帰を指示します。IP はこの ioctl に従い、ネットワークアクセスの障害経路の迂回を透過的かつ自動的に行います。

注意 – ある NIC グループに対して IP ネットワークマルチパスを使用している場合は、同じ NIC グループに対して Alternate Pathing (代替パス) を使用しないでください。同様に、代替パスを使用している場合は、IP ネットワークマルチパスを使用しないでください。NIC グループが異なる場合は、代替パスと IP ネットワークマルチパスを同時に使用できます。

物理インタフェース障害の検出

`in.mpathd` デーモンは、2 つの方法でインタフェース障害および回復を検出します。最初の方法では、インタフェースを通して ICMP エコー検査信号を送受信します。2 番目の方法では、インタフェースで RUNNING フラグを監視します。ネットワークインタフェースカードのいくつかのモデルのリンク状態は、RUNNING フラグによって反映されます。その結果、リンク障害が発生すると、すぐに検出されます。上記のいずれかの方法で障害が検出された場合、インタフェースで障害が発生したものと見なされます。また、前の両方の方法でインタフェースを回復した場合に限り、インタフェースが回復したものと見なされます。

in.mpathd デーモンは、リンクに接続されている標識 (他のシステムやルータなど) に対し、グループに属するすべてのインタフェースを通して ICMP エコー検査信号を送信し、障害や回復を検出します。デーモンは、マルチパスグループにインタフェースが追加され、検査用 IP アドレスが割り当てられていると、マルチパスグループのすべてのインタフェースを通して検査信号を送信し、障害を検出します。検査用 IP アドレスやグループの構成を行う手順については、486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

検査信号を送信する標識は in.mpathd が動的に決定するため、ユーザーは標識を指定できません。リンクに接続されているルーターは、検査信号の宛先となる標識として選択されます。リンクにルーターが接続されていない場合は、リンク上の任意のホストが選択されます。ホスト選択にあたっては、すべてのホストを意味するマルチキャストアドレス (IPv4 では 224.0.0.1、IPv6 では ff02::1) にマルチキャストパケットが送信されます。検査信号は、ICMP エコーパケットに応答する最初のいくつかのホストに送信されます。ICMP エコーパケットに応答するルーターやホストを発見できない場合には、in.mpathd は障害を検出できません。

グループの各 NIC が正常に機能するかどうかを確認するために、in.mpathd は、マルチパスグループのすべてのインタフェースを通してすべての標識に個別に検査信号を送信します。連続する 5 つの検査信号に対して応答がない場合、in.mpathd はそのインタフェースに障害があるものとみなします。検査信号を発信する頻度は、障害検出時間に依存します。障害検出時間のデフォルト値は 10 秒です。障害検出時間の変更方法については、in.mpathd(1M) のマニュアルページを参照してください。障害検出時間が 10 秒の場合、検査信号を発信する頻度はおよそ 2 秒に 1 度になります。

障害検出時間が適用されるのは、障害を検出する ICMP エコー検査信号方法だけです。リンク障害の結果、インタフェースの RUNNING フラグを消去すると、in.mpathd デーモンはフラグ状態の変更に対してただちに応答します。

in.mpathd は障害を検出すると、障害経路の迂回が行われ、すべてのネットワークアクセスが障害のあるインタフェースから同じグループの別の正常なインタフェースに移されます。待機インタフェースが構成されている場合、in.mpathd は、IP アドレス、ブロードキャスト、マルチキャストメンバーシップの移動先に待機インタフェースを選択します。待機インタフェースが構成されていない場合は、最小の IP アドレスをもつインタフェースを選択します。

同じグループ内の物理インタフェースがシステムの起動時に存在しない場合、障害の検出方法は特殊です。起動スクリプトの /etc/init.d/network が、これらの障害を検出します。これらの障害が検出された場合は、次のようなエラーメッセージを表示します。

```
moving addresses from failed IPv4 interfaces: hme0 (moved to hme1)
moving addresses from failed IPv6 interfaces: hme0 (moved to hme1)
```

注 – このような特殊な障害検出では、ホスト名ファイルに指定された静的 IP アドレスだけが、同じマルチパスグループの異なる物理インタフェースに移動します。

このような障害は、回復復帰によって自動的に回復することはできません。IP ネットワークマルチパス の RCM DR ポスト接続機能は、NIC の DR 接続を自動化します。NIC のDR を接続すると、インタフェースが結合および構成されます。リブートする前にインタフェースを削除した場合、IP マルチパスのリブート対応機能により、IP アドレスが回復されます。 その IP アドレスは、交換した NIC に転送されます。 その後、交換した NIC は元の IP マルチパスインタフェースグループに追加されます。詳細については、496 ページの「システムの起動時に存在しない物理インタフェースを回復するには」を参照してください。

物理インタフェースの回復検出

`in.mpathd` デーモンが連続した 10 個の検査パケットの応答を受信し、RUNNING フラグがインタフェースに設定されると、そのデーモンはインタフェースが回復したものと見なします。

あるインタフェースが正常でない場合、そのインタフェースのすべてのアドレスがグループ内の別の正常なインタフェースに移されます。`in.mpathd` は回復を検出するための検査信号の送出にアドレスを必要とするので、障害経路の迂回の際に移すことができない検査用 IP アドレスを構成する必要があります。この検査用 IP アドレスに関してはネットワークアクセスの障害経路の迂回は行われないため、この検査用 IP アドレスを通常のアプリケーションで使用しないようにしてください。設定手順については、486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。`in.mpathd` が回復を検出した場合、すべてのネットワークアクセスを回復したインタフェースに回復した経路への復帰を行います。

468 ページの「物理インタフェース障害の検出」 で説明されているように、自動回復復帰は、システムの起動時に存在しない物理インタフェースに対してはサポートされていません。496 ページの「システムの起動時に存在しない物理インタフェースを回復するには」を参照してください。

グループ障害

グループ障害とは、すべてのネットワークインタフェースカードで同時に障害が発生することです。`in.mpathd` はリンク障害が発生したものと見なし、障害経路の迂回を行いません。これは、すべての標識で同時に障害が発生した場合も同様です。この場合 `in.mpathd` は、現在のすべての標識選択を取り消し、新しく標識を見つけます(468 ページの「物理インタフェース障害の検出」を参照)。

注 – グループ障害は、以前のマニュアルではリンク障害と表記しています。

複数の物理インタフェースで構成されたマルチパスグループの管理

この節では、IP ネットワークマルチパスを有効にする方法について説明します。IP ネットワークマルチパス機能を使用するには、同じ IP リンクに複数の物理インタフェースで接続されていなければなりません。たとえば、同じ Ethernet スイッチや同じ IP サブネットなどに、同じマルチパスグループとして構成された複数の物理インタフェースを接続します。物理インタフェースが 1 つだけの場合は、478 ページの「1 つの物理インタフェースで構成されたマルチパスグループの管理」を参照してください。

マルチパスグループは、空文字列でない名前で識別されます。たとえば、math-link、bio-link、chem-link などは有効な名前です。名前は通常、グループがどこに接続されているかを表しています。マルチパスグループのいずれか 1 つのネットワークアダプタに障害が発生すると、障害が発生したアダプタのすべてのネットワークアクセスが、同じグループの正常なアダプタに迂回されます。ネットワークアクセスの障害経路の迂回の対象には、IPv4 のユニキャスト、ブロードキャスト、およびマルチキャストと、IPv6 のユニキャストおよびマルチキャストが含まれます。IP ネットワークマルチパスが正常に動作するには、同じマルチパスグループに属するネットワークアダプタで次の条件が満たされている必要があります。

1. マルチパスグループのすべてのネットワークアダプタに対し、同じ STREAMS モジュール群をプッシュおよび構成する必要があります。
2. 1 つのネットワークアダプタで IPv4 を結合するのであれば、マルチパスグループのすべてのネットワークアダプタで IPv4 を結合する必要があります。
3. 1 つのネットワークアダプタで IPv6 を結合するのであれば、マルチパスグループのすべてのネットワークアダプタで IPv6 を結合する必要があります。
4. Ethernet の場合は、システムにあるすべての Ethernet ネットワークアダプタに固有の MAC アドレスが必要です。SPARC プラットフォームの場合は、openboot PROM の local-mac-address? を true に設定します。x86 プラットフォームでは、何も設定する必要はありません。
5. マルチパスグループのすべてのネットワークアダプタは、同じ IP リンクに接続されていなければなりません。
6. マルチパスグループに異なる種類のインタフェースが含まれていてはなりません。グループ化するインタフェースは、`/usr/include/net/if_types.h` に定義されているのと同じタイプのインタフェースでなければなりません。たとえば、Ethernet とトークンリングを一緒にしたり、トークンバスと ATM (非同期転送モード) を一緒にしたりすることはできません。
7. ATM で IP ネットワークマルチパスを使用する場合は、ATM を LAN エミュレーションで構成する必要があります (従来の IP インスタンス間のマルチパスの使用は、現在はサポートされていません)。

注 – 4 つ目の条件は、マルチパスグループに属するインタフェースだけでなく、システムのすべてのインタフェースに適用されます。

工場出荷時に一意な MAC アドレスが設定されていないアダプタは、各アダプタの MAC アドレスを手動で構成して対処することができます。また、起動スクリプトファイル中に `ifconfig ether` コマンドを使用して構成します。

注 – 手動で構成された MAC アドレスは、システムをリブートすると保持されません。MAC アドレスは、一意なものを選択する必要があります。アダプタの MAC アドレスが一意でない場合は、IP ネットワークマルチパスは予測できない動作をする可能性があります。

物理インタフェースのグループ化

グループの構成には、`ifconfig` コマンドを使用します。このコマンドの `group` パラメータでグループ名を指定し、インタフェースの IPv4 と IPv6 に両方をそのグループに追加します。`group` オプションは次のように使用します。

`ifconfig` *interface-name*`group` *group-name*

注 – グループ名には空白文字を使用しないでください。`ifconfig` ステータスディスプレイは、スペースを表示しません。そのため、一方にスペースを含む 2 つの似たグループ名を作成した場合、ステータスディスプレイでは同じように見えてしまうことになります。実際には、別のグループ名です。このため、混乱を招くことがあります。

特定のグループに IPv4 インタフェースを追加すると、同じグループに IPv6 インタフェースが自動的に追加されます。さらに、同じコマンドを使って、同じサブネットに接続された 2 つ目のインタフェースを同じグループに入れることができます。486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

マルチパスグループからインタフェースを削除するには、group オプションで空文字列を指定します。492 ページの「グループからインタフェースを削除するには」を参照してください。

別のマルチパスグループに属するインタフェースを新しいグループに入れる場合、既存のグループからそのインタフェースを削除する必要はありません。インタフェースを新しいグループに追加すると、そのインタフェースは現在のグループから自動的に削除されます。493 ページの「インタフェースを既存のグループから別のグループに移動するには」を参照してください。

1 つのマルチパスグループに構成できるネットワークアダプタの数に制限はありません。group オプションを論理インタフェースに指定することはできません。たとえば、hme0 は使用できますが、hme0:1 は使用できません。

マルチパスグループのすべてのインタフェースは、同じ IP リンクに接続されていなければなりません。これは、インタフェースに障害が発生すると、障害経路の迂回処理によって、すべての IP アドレスが障害の発生したインタフェースからグループ内の正常なインタフェースに移されるからです。正常なインタフェースに切り替えられたアドレスにルーターがパケットのルーティングを引き続き行うためには、その正常なインタフェースが同じ IP リンクに接続されていなければなりません。

検査用 IP アドレスの構成

マルチパスグループのすべての物理インタフェースを構成する際には、検査用 IP アドレスを指定する必要があります。検査用 IP アドレスは、障害や回復の検出に必要です。検査用 IP アドレスが指定されていないと、その物理インタフェースは障害経路の迂回には使用されません。in.mpathd だけが検査用 IP アドレスを使用します。通常のアプリケーションでは、このアドレスを使用しないようにしてください。インタフェースに障害が発生しても、このアドレスに関しては障害経路の迂回は行われません。IPv4 では、検査用 IP アドレスを構成する際には、通常のアプリケーションが検査用 IP アドレスを使用しないように設定してください。486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

ここでは、次のインターネットプロトコルに対する検査用 IP アドレス構成の概念について説明します。

- IPv4
- IPv6

IPv4 検査用アドレス

in.mpathd マルチパスデーモンは、障害や回復を検出するための検査用 IP アドレスを必要とします。この IP アドレスは、ルーティング可能なアドレスでなければなりません。つまり、このアドレスのネットワークアドレス (ネットワーク接頭子) がリンク内のすべてのルーターから認識可能でなければなりません。検査用 IP アドレスの構成には、ifconfig コマンドの -failover オプションを使用します。検査用 IP アドレスを設定する構文は次の通りです。

```
# ifconfig interface-name addif ip-address <other-parameters> -failover up
```

<other-parameters> には、実際の構成に応じたパラメータを指定します。詳細は、ifconfig(1M) のマニュアルページを参照してください。IPv4 検査用アドレスの設定手順については、486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

たとえば、アドレスが 19.16.85.21、ネットマスクおよびブロードキャストアドレスがデフォルト値で、かつ検査用に使用できる論理インタフェースを新規に作成するには、次のように指定します。

```
# ifconfig hme0 addif 19.16.85.21 netmask + broadcast + -failover up
```

注 – この検査用 IP アドレスをアプリケーションから使用されないようにするため IPv4 検査用アドレスを deprecated と指定する必要があります。486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

アドレスの障害経路の迂回属性を有効にするには、failover (ダッシュ (-) をつけない) を指定します。

注 – マルチパスグループのすべての検査用 IP アドレスには、同じネットワークアドレスを使用してください。つまり、すべての検査用 IP アドレスは 1 つの IP サブネットに属していなければなりません。

IPv6 検査用 IP アドレス

リンクローカルアドレスが物理インタフェースに結び付けられているので、IPv6 検査用 IP アドレスを構成するには、リンクローカルアドレス自体を使用します。したがって、IPv6 では、別個の IP アドレスは必要ありません。IPv6 の場合、-failover オプションの構文は次の通りです。

```
# ifconfig interface-name inet6 -failover
```

IPv6 検査用 IP アドレスの設定手順については、486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

マルチパスグループですべてのグループのインタフェースに IPv4 と IPv6 の両方が使用される場合には、別個の IPv4 検査用アドレスは必要ありません。in.mpathd デーモンは、IPv6 リンクローカルアドレスを使ってインタフェースを調べることができます。IPv6 リンクローカルアドレスは、IPv6 を結合すると自動的に作成されます。

アドレスの障害経路の迂回属性を有効にするには、failover (ダッシュ (-) をつけない) を指定します。

注 – 有効な IPv6 検査用 IP アドレスは、リンクローカルアドレスだけです。

アプリケーションによる検査用 IP アドレス使用の防止

検査用 IP アドレスを構成したら、このアドレスが通常のアプリケーションで使用されないようにする必要があります。検査用 IP アドレスに対して障害経路の迂回処理が行われないため、アプリケーションから検査用 IP アドレスを使用できるようにすると、検査用 IP アドレスを使用したアプリケーションは障害迂回の処理時に異常終了します。検査用 IP アドレスが通常のアプリケーションに使用されるのを防ぐには、`ifconfig` コマンドを使って検査用 IP アドレスを`deprecated` と指定します。このオプションは次の構文により指定します。

`ifconfig` *interface-name* `deprecated`

アドレスを `deprecated` と指定すると、このアドレスをアプリケーションが明示的に指定しない限り、IP はこのアドレスを通信のソースアドレスとして選択しません。このようなアドレスに明示的に指定するのは、`in.mpathd` だけです。486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

ネームサービス (DNS、NIS、NIS+) にはリンクローカルアドレスは登録されませんので、アプリケーションはリンクローカルアドレスを通信に使用しません。したがって、IPv6 検査用 IP アドレスを `deprecated` と指定する必要はありません。

注 – IPv6 リンクローカルアドレスは `deprecated` と指定しないでください。

アドレスの `deprecated` 属性を無効にするには、`-deprecated` オプションを使用します。

注 – IPv4 検査用アドレスは、ネームサービスデータベース (DNS、NIS、または NIS+) に入れないでください。IPv6 では、検査用 IP アドレスとしてリンクローカルアドレスが使用されますが、このアドレスは通常、ネームサービスデータベースに入れられません。

自動的に構成された IPv6 アドレスは、システムをリブートすると保持されません。リブートするときに IP アドレスを保持する必要がある場合には、アプリケーションで静的 IP アドレスを使用します。

`hostname` ファイルによるグループと検査用 IP アドレスの構成

`/etc/hostname.` *interface* ファイルをマルチパスグループと検査用 IP アドレスの構成に使用できます。`/etc/hostname.` *interface* ファイルを使ってマルチパスグループを構成するには、次の構文に従ってファイルに 1 行追加します。

interface-address <parameters> group *group-name* up \
addif *logical-interface-address* <parameters> up

たとえば、次の構成に基づいて test グループを作成します。

- 物理インタフェース hme0 のアドレスが 19.16.85.19
- 論理インタフェースのアドレスが 19.16.85.21
- deprecated と -failover を指定
- ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。

この場合、/etc/hostname.hme0 ファイルに次の行を追加します。

```
19.16.85.19 netmask + broadcast + group test up \
addif 19.16.85.21 deprecated -failover netmask + broadcast + up
```

IPv4 hostname ファイルの構成手順については、486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してください。

IPv6 の設定では、次の構文に従って /etc/hostname6.*interface* ファイルに 1 行追加します。

<パラメータ> group *group-name* up

たとえば、IPv6 検査用 IP アドレスを使って hme0 に対し test グループを作成するには、/etc/hostname6.hme0 ファイルに次の行を追加します。

```
-failover group test up
addif 1080::56:a00:20ff:feb9:19fa up
```

IPv6 hostname6 ファイルの構成手順については、486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」を参照してくだい。

待機インタフェースの構成

マルチパスグループには、待機インタフェースを構成できます。マルチパスグループには、待機インタフェースを構成できます。名前が示すように、このインタフェースは待機中とみなされ、グループの他のインタフェースに障害が発生しない限り使用されません。待機インタフェースが障害迂回 IP アドレスとして機能していない場合には、IFF_INACTIVE フラグが設定されます。その結果、アクティブなインタフェースに障害が発生すると、障害経路の迂回に待機インタフェースが必ず選択されます。待機インタフェースが選択された後、IFF_INACTIVE フラグがそのインタフェースで消去されます。その後、アクティブとなった待機インタフェースは他のアクティブなインタフェースと同様に処理されます。よって、一部の障害では、待機インタフェースが選択されないことがあります。この場合、待機インタフェースではなく、いくつかの IP アドレスにアクティブなインタフェースが選択されます。

待機インタフェースは、通常のデータパケットの送信には使用されません。したがって、待機インタフェースでのデータの流量は限られています。待機インタフェースが正常であるかどうかを判定するための検査信号の送信に使用するため、待機インタフェースには検査用 IP アドレスが必要です。待機インタフェースに検査用 IP アド

レスが指定されていないと、グループの別のインタフェースに障害が発生しても、この待機インタフェースは障害経路の迂回先にはなりません。次の場合には、待機インタフェースにデータが流れることがあります。

- ネットワーク上の別ホストが待機インタフェースのアドレスを使ってこのホストと通信すると、着信パケットにはその待機インタフェースが使用されます。
- 待機インタフェースのアドレスをアプリケーションが (`bind` または `IP_ADD_MEMBERSHIP` を使って) 指定すると、継続的にこの待機インタフェースにデータが流れることがあります。

このように待機インタフェースは、アプリケーションによって明示的に選択されない限り、通常 (検査目的以外には) 選択されません。グループのインタフェースに障害が発生すると、すべてのネットワークアクセスは待機インタフェースに迂回されます。待機インタフェースを構成するには、次のように `ifconfig` コマンドの `standby` オプションを使用します。

```
# ifconfig interface-name standby group group-name
```

この手順については、489 ページの「インタフェースの 1 つが待機インタフェースであるマルチパスグループを構成するには」を参照してください。

待機インタフェースに検査用 IP アドレスが設定されていると、`in.mpathd` デーモンは、待機インタフェースを通して検査信号を送信します。待機インタフェースには、検査用 IP アドレスだけを設定してください。待機インタフェースに他のアドレスを追加しても、追加は失敗します。検査用 IP アドレス以外のアドレスをもつインタフェースを待機インタフェースに指定すると、これらのアドレスは自動的にグループの他のインタフェースに移され、検査用 IP アドレスがある場合は待機インタフェースには検査用 IP アドレスだけが残ります。待機インタフェースに検査用 IP アドレス以外のアドレスを設定しないことを推奨します。

検査用 IP アドレスを指定するには、`ifconfig` コマンドの `standby` や `up` オプションの前に `deprecated` と `-failover` オプションを指定します。

待機インタフェースに検査用 IP アドレスを設定するには、次の構文を使用します。

```
# ifconfig interface-name plumb ip-address
   <other-parameters> deprecated -failover standby up
```

`<other-parameters>` には、実際の構成に応じたパラメータを指定します。詳細は、`ifconfig`(1M) のマニュアルページを参照してください。

注 – 待機インタフェースに検査用 IP アドレスが設定されていないと、待機インタフェースは障害経路の迂回には使用されません。

たとえば、次の構成に基づいて検査用 IP アドレスを作成します。

- 物理インタフェース `hme2` を待機インタフェースにする。
- アドレスは `19.16.85.22`
- `deprecated` と `-failover` を指定

- ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。

この場合、コマンド行に次のように入力します。

```
# ifconfig hme2 plumb 19.16.85.22 netmask + broadcast + deprecated -failover standby up
```

注 – インタフェースは、アドレスに対して障害経路の迂回が行われないように設定されたあとにだけ、待機インタフェースとして設定されます。

この手順については、489 ページの「インタフェースの 1 つが待機インタフェースであるマルチパスグループを構成するには」を参照してください。

待機状態を解除するには、次の構文を使用します。

```
# ifconfig interface-name -standby
```

1 つの物理インタフェースで構成されたマルチパスグループの管理

マルチパスグループにネットワークアダプタが 1 つしかない場合でも、この NIC の障害を検出するようにネットワークアダプタを構成することができます。

グループに NIC が 1 つしかなければ障害経路の迂回は行われないため、グループの物理インタフェースごとに別個の検査用 IP アドレスを設定する必要はありません。検査用のアドレスを障害経路の迂回が行われないアドレスとして (IFF_NOFAILOVER) 設定すれば、デーモンはそのインタフェースを使って検査信号を送信します。複数の物理インタフェースがある場合とは異なり、1 つの物理インタフェースを deprecated と指定する必要はありません。

インタフェースの IPv4 アドレスに対して障害経路の迂回が行われないように (NOFAILOVER) 設定するには、次の構文を使用します。

```
# ifconfig interface-name -failover group group-name
```

IPv6 の場合は、次の構文を使用します。

```
# ifconfig interface-name inet6 -failover group group-name
```

デーモンが障害を検出すると、インタフェースの状態がそれに応じて変更され、コンソールにログが出力されます。

注 – 障害が検査信号の標識に発生したのか、NIC に発生したのかを検証する方法はありません。これは、検査に使用できる物理インタフェースが標識に 1 つしかないためです。サブネットのデフォルトルーターが 1 つしかない場合に、グループの物理インタフェースが 1 つだけであれば、マルチパスを無効にしてください。IPv4 と IPv6 のデフォルトルーターが別個に存在する場合 (または、複数のデフォルトルーターが存在する場合) は、検査信号の対象は 2 つ以上あります。 したがって、マルチパスを有効にしても問題ありません。

マルチパスグループからのネットワークアダプタの削除

`ifconfig` コマンドの `group` オプションに空文字列を指定すると、インタフェースが既存のグループから削除されます 。492 ページの「グループからインタフェースを削除するには」を参照してください。 グループからインタフェースを削除する場合は、慎重に行う必要があります。マルチパスグループの他のインタフェースに障害が発生している場合、障害経路の迂回が行われていることがあります。たとえば、`hme0` に障害が発生し、すべてのアドレスが、同じグループに属する `hme1` に移されたとします。このグループから `hme1` を削除すると、`in.mpathd` はこれらの障害経路の迂回が行われたアドレスをグループ内の他のインタフェースに戻します。正常に動作しているインタフェースがグループ内になければ障害経路の迂回が行われず、すべてのネットワークアクセスは維持できません。

同様に、インタフェースがグループに属しており、結合解除する必要がある場合は、まずグループからインタフェースを削除する必要があります。その後、インタフェースに構成されたすべての IP アドレスを確実に維持します。これは、グループから削除されるインタフェースの構成を `in.mpathd` デーモンが再現しようとするからです。インタフェースの使用を中止する場合は、その前に構成が再現されていなければなりません。障害経路の迂回の前後でインタフェースの構成がどのように変化するかについては、480 ページの「マルチパスデーモン」を参照してください。

ネットワークアダプタの切り離し

動的再構成 (DR: Dynamic Reconfiguration) では、IP ネットワークマルチパスを使用して、IP を使用中のユーザーに影響を及ぼすことなく特定のネットワークデバイスを切り離すことができます。NIC が動的再構成によって切り離される (オフラインになる) 前に、その NIC のすべての障害迂回 IP アドレスは、同じ IP ネットワークマルチパスグループ内の他の NIC へ自動的に経路迂回処理されます。検査用 IP アドレスは無効にされ、NIC は結合解除されます。

また、IP マルチパスのリブート対応機能により、存在しないカードに対応する /etc/hostname.* ファイル内の IP アドレスは、自動的に同じ IP ネットワークマルチパスグループ内の代替インタフェースに移されます。ただし、元のインタフェースが後でそのシステムに戻されたとしても、これらの IP アドレスは自動的にそのインタフェースへは戻りません。

マルチパスデーモン

in.mpathd マルチパスデーモンは、グループに属するすべてのインタフェースから検査信号を送信することによって障害や回復を検出します。in.mpathd マルチパスデーモンも、グループに属する各インタフェースで RUNNING フラグを監視することによって障害や回復を検出します。 グループに属するインタフェースに検査用 IP アドレスがあれば、デーモンは検査信号の送信を開始し、そのインタフェースに障害がないかどうかを判断します。連続する 5 つの検査信号に応答がない、または RUNNING フラグが設定されていないと、そのデーモンはそのインタフェースに障害が発生したと見なします。検査頻度は、障害検出時間によって異なります。デフォルトの障害検出時間は 10 秒です。つまり、検査頻度は 2 秒に 1 回の割合です。ネットワークで同期が発生するのを防ぐため、検査は定期的には実行されません。連続する 5 つの検査信号に応答がないと、in.mpathd は、そのインタフェースに障害が発生したとみなし、障害経路の迂回が行われ、ネットワークアクセスを障害のあるインタフェースからグループの別の正常なインタフェースへ移します。待機インタフェースが構成されている場合は、IP アドレスと、ブロードキャストやマルチキャストメンバーシップの障害経路の迂回用に待機インタフェースが選択されます。待機インタフェースが構成されていない場合は、最小の IP アドレスをもつインタフェースが選択されます。関連情報については、in.mpathd(1M) のマニュアルページを参照してください。

次の 2 つの例は、一般的な構成と、インタフェースに障害が発生したときに構成がどのように変化するかを示しています。hme0 インタフェースに障害が発生すると、すべてのアドレスが hme0 から hme1 に移されます。

例 **27–1** インタフェースに障害が発生する前のインタフェース構成

```
hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
     inet 19.16.85.19 netmask ffffff00 broadcast 19.16.85.255
     groupname test
hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
     index 2 inet 19.16.85.21 netmask ffffff00 broadcast 129.146.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
     inet 19.16.85.20 netmask ffffff00 broadcast 19.16.85.255
     groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
     index 2 inet 19.16.85.22 netmask ffffff00 broadcast 129.146.85.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
     inet6 fe80::a00:20ff:feb9:19fa/10
     groupname test
```

例 **27–1** インタフェースに障害が発生する前のインタフェース構成　*(続き)*

```
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
     inet6 fe80::a00:20ff:feb9:1bfc/10
     groupname test
```

例 **27–2** インタフェースに障害が発生した後のインタフェース構成

```
hme0: flags=19000842<BROADCAST,RUNNING,MULTICAST,IPv4,NOFAILOVER,FAILED> mtu 0 index 2
        inet 0.0.0.0 netmask 0
        groupname test
hme0:1: flags=19040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,FAILED>
        mtu 1500 index 2 inet 19.16.85.21 netmask ffffff00 broadcast 129.146.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 19.16.85.20 netmask ffffff00 broadcast 19.16.85.255
        groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
        index 2 inet 19.16.85.22 netmask ffffff00 broadcast 129.146.85.255
hme1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 6
        inet 19.16.85.19 netmask ffffff00 broadcast 19.16.18.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER,FAILED> mtu 1500 index 2
        inet6 fe80::a00:20ff:feb9:19fa/10
        groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
        inet6 fe80::a00:20ff:feb9:1bfc/10
        groupname test
```

上記の例では、障害が発生したことを示す FAILED フラグが hme0 に設定されています。また、hme1:2 が新しく作成されているのがわかります。hme0 の構成は hme1:2 に引き継がれました。これによって、アドレス 19.16.85.19 は、hme1 からアクセスできるようになります。19.16.85.19 に対応するマルチキャストメンバーシップはこの後もパケットを受信できますが、パケットは hme1 を通して受信されます。アドレス 19.16.85.19 が hme0 から hme1 に障害経路の迂回が行われるとき、hme0 にはダミーアドレス 0.0.0.0 が作成されます。ダミーアドレスは、回復した経路への復帰時に削除されます。ダミーアドレスは、hme0 を引き続きアクセスできる状態に保つために作成されます。hme0 がなければ、hme0:1 は存在できません。

同様に、IPv6 アドレスが hme0 から hme1 へ移されています。IPv6 では、マルチキャストメンバーシップはインタフェースインデックスに関連付けられています。マルチキャストメンバーシップも hme0 から hme1 に移されます。in.ndpd が構成を行うすべてのアドレスも移されます (上記の例には示されていません)。

in.mpathd デーモンは引き続き、障害が発生した NIC の hme0 を通して検査を行います。(デフォルトの障害検出時間 10 秒の間に) 10 回の応答を連続して受け取ると、そのデーモンはインタフェースが回復したと見なし、回復した経路への復帰を行います。回復した経路への復帰が行われると、元の構成が再び確立されます。

障害や回復の検出時にコンソールに出力されるエラーメッセージについては、in.mpathd(1M) のマニュアルページを参照してください。

マルチパス構成ファイル

in.mpathd デーモンは、/etc/default/mpathd 構成ファイルの設定値を使ってマルチパスを構成します。このファイルへの変更は、in.mpathd が起動したときと SIGHUP シグナルを受信したときに有効になります。このファイルには、次のデフォルト設定値と情報が含まれています。

```
#
# Time taken by mpathd to detect a NIC failure in ms. The minimum time
# that can be specified is 100 ms.
#
FAILURE_DETECTION_TIME=10000
#

# Failback is enabled by default. To disable failback turn off this option
#
FAILBACK=yes
#

# By default only interfaces configured as part of multipathing groups
# are tracked. Turn off this option to track all network interfaces
# on the system
#
TRACK_INTERFACES_ONLY_WITH_GROUPS=yes
```

/etc/default/mpathd 構成ファイルの構成手順については、498 ページの「マルチパス構成ファイルを構成するには」を参照してください。

障害検出時間

障害検出時間の設定値は小さくすることができます。ネットワークの負荷が高すぎると、障害検出時間が守られないことがあります。その場合、in.mpathd はメッセージをコンソールに出力します。また、現在の達成可能な時間もコンソールに出力します。応答が正しく戻ってくる場合は、このファイルの障害検出時間に従って検出が行われます。

回復した経路への復帰

障害経路の迂回が行われた後に障害の発生したインタフェースが回復すると、回復した経路への復帰が行われます。ただし、FAILBACK が no に設定されていると、インタフェースの回復した経路への復帰は行われません。

468 ページの「物理インタフェース障害の検出」 で説明されているように、自動回復復帰は、システムの起動時に存在しない物理インタフェースに対してはサポートされていません。496 ページの「システムの起動時に存在しない物理インタフェースを回復するには」を参照してください。

「グループに属するインタフェースのみの追跡」オプション

TRACK_INTERFACES_ONLY_WITH_GROUPS オプションが無効になっていると、in.mpathd はシステムのすべてのインタフェースを追跡します。障害を検出すると、適切なメッセージをコンソールに出力します。このオプションが正しく機能するには、すべてのインタフェース上の Ethernet アドレスが固有のものでなければなりません。

第 28 章

ネットワークマルチパスの管理 (手順)

この章では、インタフェースグループを作成および使用するための手順や、検査用 IP アドレス、hostname ファイル、マルチパス構成ファイルを構成するための手順について説明します。

この章では、以下の内容について説明します。

マルチパスインタフェースグループの構成

この節では、マルチパスインタフェースグループの構成手順とインタフェースを待機インタフェースに指定するための手順を説明します。

472 ページの「物理インタフェースのグループ化」にも関連情報が記述されています。

マルチパスインタフェースグループの構成 (作業マップ)

表 **28–1** マルチパスインタフェースグループの構成 (作業マップ)

タスク	説明	参照先
2 つのインタフェースでマルチパスインタフェースグループを構成	`ifconfig` コマンド、`group` オプション、`-failover` オプション、`deprecated` オプション、および `/etc/hostname.`*interface* ファイルを使用する。	486 ページの「2 つのインタフェースでマルチパスインタフェースグループを構成するには」
インタフェースの 1 つが待機インタフェースであるマルチパスグループを構成	`ifconfig` コマンド、`group` オプション、`standby` オプション、`-failover` オプション、および `/etc/hostname.`*interface* ファイルを使用する。	489 ページの「インタフェースの 1 つが待機インタフェースであるマルチパスグループを構成するには」
物理インタフェースが属するグループを表示	`ifconfig` コマンドとインタフェース名を使用する。	491 ページの「物理インタフェースが属するグループを表示するには」
グループにインタフェースを追加	`ifconfig` コマンドとインタフェース名を使用する。	492 ページの「グループにインタフェースを追加するには」
グループからインタフェースを削除	`ifconfig` コマンドと空文字列を使用して、IP ネットワークマルチパスを無効にする。	492 ページの「グループからインタフェースを削除するには」
インタフェースを既存のグループから別のグループに移動	`ifconfig` コマンドと `group` オプションを使用する。	493 ページの「インタフェースを既存のグループから別のグループに移動するには」

▼ 2 つのインタフェースでマルチパスインタフェースグループを構成するには

1. スーパーユーザーになります。
2. 次のコマンドを使って、個々の物理インタフェースをマルチパスグループに入れます。

   ```
   # ifconfig interface-name group group-name
   ```

   たとえば、`hme0` と `hme1` を `test` グループに入れるには、次のように入力します。

   ```
   # ifconfig hme0 group test
   # ifconfig hme1 group test
   ```

3. すべての物理インタフェースに対し検査用 **IP** アドレスを指定します。
   - IPv4 検査用アドレスの場合は、次のコマンドを使用します。

注 – この手順では、物理インタフェースのアドレスがすでに構成されているものとします。

```
# ifconfig interface-name addif ip-address <parameters> -failover deprecated up
```

たとえば、次の構成に基づいて hme0 の検査用 IP アドレスを設定します。

- アドレスは 19.16.85.21
- ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。
- -failover と deprecated を指定する。

この場合、次のコマンドを入力します。

```
# ifconfig hme0 addif 19.16.85.21 netmask + broadcast + -failover deprecated up
```

構成を確認するには、次のように入力します。

```
# ifconfig hme0:1
    hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
    mtu 1500 index 2 inet 19.16.85.21 netmask ffffff00 broadcast 19.16.85.255
```

注 – この検査用 IP アドレスをアプリケーションが使用しないようにするため IPv4 検査用 IP アドレスを deprecated と指定する必要があります。

次に、下記の構成に基づいて hme1 の検査用 IP アドレスを設定します。

- アドレスは 19.16.85.22
- ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。
- -failover と deprecated を指定

次のコマンドを入力します。

```
# ifconfig hme1 addif 19.16.85.22 netmask + broadcast + -failover deprecated up
```

- IPv6 検査用 IP アドレスの場合は、次のコマンドを使用します。

  ```
  # ifconfig interface-name inet6 -failover
  ```

注 – この時点では IPv4 アドレスを持つ物理インタフェースがすでにマルチパスグループに追加されているため、IPv6 アドレスを持つ物理インタフェースも自動的に同じマルチパスグループに追加されています。IPv6 アドレスを持つ物理インタフェースが最初にマルチパスグループに追加されていることがあります。その場合、IPv4 アドレスを持つ物理インタフェースが自動的に同じマルチパスグループに追加されます。

たとえば、hme0 に IPv6 検査用 IP アドレスを指定するには、次のコマンドを使用します。

```
# ifconfig hme0 inet6 -failover
```

構成を確認するには、次のように入力します。

```
# ifconfig hme0 inet6

 hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500
            index 2 inet6 fe80::a00:20ff:feb9:17fa/10
            groupname test
```

注 – 検査用 IP アドレスをアプリケーションから使用されないようにするために IPv6 検査用 IP アドレスを deprecated と指定する必要はありません。

---

2 つ目のインタフェース hme1 には、次のコマンドを使用します。

```
# ifconfig hme1 inet6 -failover
```

4. (この手順は、リブート後も構成を有効にしたい場合だけ必要です。) リブート後も構成を有効にするには、次の手順を実行します。

   - IPv4 の場合は、/etc/hostname.*interface* ファイルに次の行を追加します。

     ```
     interface-address <parameters> group group-name up \
         addif logical-interface -failover deprecated <parameters> up
     ```

     注 – この検査用 IP アドレスは、次回のリブートで有効になります。構成をその場で有効にするには、手順 1 から 3 を実行する必要があります。

     たとえば、hme0 に対し次の構成に基づいて test グループを作成します。

     - 物理インタフェース hme0 のアドレスが 19.16.85.19
     - 論理インタフェースのアドレスが 19.16.85.21
     - deprecated と -failover を指定する。
     - ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。

     この場合、/etc/hostname.hme0 ファイルに次の行を追加します。

     ```
     19.16.85.19 netmask + broadcast + group test up \
         addif 19.16.85.21 deprecated -failover netmask + broadcast + up
     ```

     同様に、hme1 を同じグループ (test) に入れ、検査用 IP アドレスを指定するには、次のコマンドを入力します。

     ```
     19.16.85.20 netmask + broadcast + group test up \
         addif 19.16.85.22 deprecated -failover netmask + broadcast + up
     ```

   - IPv6 の場合は、/etc/hostname6.*interface* ファイルに次の行を追加します。

```
-failover group group-name up
```

注 – この検査用 IP アドレスは、次回のリブートで有効になります。構成をその場で有効にするには、手順 1 から 3 を実行する必要があります。

たとえば、IPv6 アドレスを持つ hme0 に対して test グループを作成するには、次の行を /etc/hostname6.hme0 ファイルに追加します。

```
-failover group test up
```

同様に、hme1 を同じグループ (test) に入れ、検査用 IP アドレスを指定するには、次の行を /etc/hostname6.hme1 ファイルに追加します。

```
-failover group test up
```

注 – マルチパスグループにさらにインタフェースを追加する場合は、上記の手順 1 から 3 を繰り返します。新しいインタフェースを、動作しているシステム上の既存のグループに追加することができます。ただし、リブート後は変更の内容は失われます。

▼ インタフェースの 1 つが待機インタフェースであるマルチパスグループを構成するには

この手順の例では、hme1 を待機インタフェースとして構成します。

注 – 待機インタフェースのアドレスには、検査用 IP アドレスしか指定できません。

1. **486** ページの「**2** つのインタフェースでマルチパスインタフェースグループを構成するには」の手順 **1** と **2** を実行します。
2. 次の手順を実行し、すべての物理インタフェースに検査用 **IP** アドレスを設定します。
   a. **hme0** のような非待機インタフェースは、**486** ページの「**2** つのインタフェースでマルチパスインタフェースグループを構成するには」の手順 **3** を実行します。
   b. 待機インタフェースは、次のコマンドを使って検査用 **IP** アドレスを設定します。

注 – 待機インタフェースのアドレスには、検査用 IP アドレスしか指定できません。待機インタフェースは、これ以外の IP アドレスを持つことはできません。

```
# ifconfig interface-name plumb ip-address <other-parameters> deprecated -failover
standby up
```

注 – -failover オプションは standby オプションより前に、standby オプションは up より前にそれぞれ指定する必要があります。

<other-parameters> には、実際の構成に応じたパラメータを指定します。詳細は、ifconfig(1M) のマニュアルページを参照してください。

たとえば、次の構成に基づいて検査用 IP アドレスを作成します。

- 物理インタフェース hme1 を待機インタフェースにする。
- アドレスは 19.16.85.22
- deprecated と -failover を指定する。
- ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。

この場合、次のコマンドを入力します。

```
# ifconfig hme1 plumb 19.16.85.22 netmask + broadcast + deprecated -failover standby up
```

結果を確認するには、次のコマンドを入力します。

```
# ifconfig hme1
flags=69040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,STANDBY,INACTIVE>
        mtu 1500 index 4 inet 19.16.85.22 netmask ffffff00 broadcast 19.16.85.255
        groupname test
```

IPv6 の場合、検査用 IP アドレスを作成するには、次のコマンドを使用します。

```
ifconfig hme1 plumb -failover standby up
```

INACTIVE は、このインタフェースが送信パケットには使用されないことを示します。この待機インタフェースに障害経路の迂回が行われると、INACTIVE 状態は取り消されます。

3. **(**この手順は、リブート後も構成を有効にしたい場合だけ必要です。**)** リブート後も構成を有効にするには、次の手順を実行します。

   - IPv4 の場合は、/etc/hostname.*interface* ファイルに次の行を追加します。

     interface-address <parameters> group *group-name* up \
         addif *logical-interface*-failover deprecated <parameters> up

注 – この検査用 IP アドレスは、次回のリブートで有効になります。構成をその場で有効にするには、手順 1 と 2 を実行する必要があります。

たとえば、hme0 に対し次の構成に基づいて test グループを作成します。

- 物理インタフェース hme0 のアドレスが 19.16.85.19
- 論理インタフェースのアドレスが 19.16.85.21
- deprecated と -failover を指定する。
- ネットマスクおよびブロードキャストアドレスをデフォルト値に設定する。

この場合、/etc/hostname.hme0 ファイルに次の行を追加します。

```
19.16.85.19 netmask + broadcast + group test up \
    addif 19.16.85.21 deprecated -failover netmask + broadcast + up
```

同様に、hme1 を同じグループ (test) に入れ、検査用 IP アドレスを指定するには、次のコマンドを入力します。

```
19.16.85.22 netmask + broadcast + deprecated group test -failover standby up
```

- IPv6 の場合は、/etc/hostname6.*interface* ファイルに次の行を追加します。

```
-failover group group-name up
```

注 – この検査用 IP アドレスは、次回のリブートで有効になります。構成をその場で有効にするには、手順 1 と 2 を実行する必要があります。

たとえば、IPv6 アドレスを持つ hme0 に対して test グループを作成するには、次の行を /etc/hostname6.hme0 ファイルに追加します。

```
-failover group test up
```

同様に、hme1 を同じグループ (test) に入れ、検査用 IP アドレスを指定するには、次の行を /etc/hostname6.hme1 ファイルに追加します。

```
-failover group test standby up
```

▼ 物理インタフェースが属するグループを表示するには

1. スーパーユーザーになります。
2. コマンド行から次のコマンドを入力します。

```
# ifconfig interface-name
```

たとえば、hme0 のグループ名を表示するには、次のコマンドを入力します。

```
# ifconfig hme0
    hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
      index 2 inet 19.16.85.19 netmask ffffff00 broadcast 19.16.85.255
      groupname test
```

IPv6 だけのグループ名を表示するには、次のコマンドを入力します。

```
# ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:19fa/10
            groupname test
```

▼ グループにインタフェースを追加するには

1. スーパーユーザーになります。
2. コマンド行から次のコマンドを入力します。

   # ifconfig *interface-name* group *group-name*

たとえば、test グループに hme0 を追加するには、次のコマンドを入力します。

```
# ifconfig hme0 group test
```

▼ グループからインタフェースを削除するには

1. スーパーユーザーになります。
2. コマンド行から次のコマンドを入力します。

   # ifconfig *interface-name* group ""

   引用符 ("") は空文字列を表します。

たとえば、test グループから hme0 を削除するには、次のコマンドを入力します。

```
# ifconfig hme0 group ""
    # ifconfig hme0
    hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
            index 2 inet 19.16.85.19 netmask ffffff00 broadcast 19.16.85.255
    # ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:19fa/10
```

479 ページの「マルチパスグループからのネットワークアダプタの削除」に関連情報が記述されています。

▼ インタフェースを既存のグループから別のグループに移動するには

1. スーパーユーザーになります。
2. コマンド行から次のコマンドを入力します。

```
# ifconfig interface-name group group-name
```

注 – インタフェースを新しいグループに追加すると、そのインタフェースは現在のグループから自動的に削除されます。

たとえば、`test` グループから `hme0` を削除し、`cs-link` グループに追加するには、次のコマンドを入力します。

```
# ifconfig hme0 group cs-link
```

そのインタフェースが現在のグループから削除され、`cs-link` グループに追加されます。

障害が発生した物理インタフェースの交換または物理インタフェースの DR 切り離し/DR 接続

この節の手順は、`ifconfig`(1M) を使用して構成される IP 層だけに適用されます。ATM または他のサービスなど、IP 層よりも上位または下位の層が自動化されていない場合には、手動による特別な手順が必要です。この特別な手順は、事前切り離し時の構成解除および事後接続後の構成を行うために実行します。障害または DR の対処方法については、該当する層およびアプリケーションのマニュアルを参照してください。

障害が発生した物理インタフェースを交換する場合は、まず次の各手順を手作業で行う必要があります。次の手順の例では、インタフェースとして `hme0` と `hme1` を使用します。両インタフェースとも同じマルチパスグループに属し、`hme0` に障害が発生したとします。さらに、論理インタフェース `hme0:1` が検査用 IP アドレスをもっているとします。

注 – 次の手順の例では、障害のあるインタフェースを同じ名前の物理インタフェースで置き換えます (たとえば、hme0 を hme0 で置き換えます)。

▼ 障害が発生した物理インタフェースを取り外すには

注 – 検査用 IP アドレスが /etc/hostname.hme0 ファイルを使用して結合されている場合は、 の手順 1 は省略することができます。

1. 次のコマンドを実行して、検査用 **IP** アドレスの構成情報を入手します。

```
# ifconfig hme0:1

hme0:1:
flags=9040842<BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
inet 129.146.233.250 netmask ffffff00 broadcast 129.146.233.255
```

   この情報は、物理インタフェースを交換する時に、検査用 IP アドレスを再結合するために必要です。

   検査用 IP アドレスの hostname ファイルを使用した構成方法についての詳細は、475 ページの「hostname ファイルによるグループと検査用 IP アドレスの構成」を参照してください。

2. 物理インタフェースの取り外し方については、**cfgadm(1M)** のマニュアルページ、『*Sun Enterprise 6x00, 5x00, 4x00, 3x00* システム *Dynamic Reconfiguration* ユーザー マニュアル』、または『*Sun Enterprise 10000 DR* 構成マニュアル』を参照してください。

▼ 障害が発生した物理インタフェースを交換するには

1. 物理インタフェースの交換方法については、**cfgadm(1M)** のマニュアルページ、『*Su n Enterprise 6x00, 5x00, 4x00, 3x00* システム *Dynamic Reconfiguration* ユーザー マ ニュアル』、または『*Sun Enterprise 10000 DR* 構成 マニュアル』、あるいは『*Sun Fire 880 Dynamic Reconfiguration* ユーザーマニュアル 』を参照してください。

2. 次のコマンドを実行して、検査用 **IP** アドレスを結合し、有効にします。

```
# ifconfig hme0 <test address configuration>
```

注 – この検査用アドレス構成は、/etc/hostname.hme0 ファイルに構成されたものと同じです。494 ページの「障害が発生した物理インタフェースを取り外すには」の手順 1 に従った場合は、検査用 IP の構成は同手順で表示された構成と同じにします。

この構成によって、in.mpathd デーモンが検査を再開します。 検査の結果、in.mpathd は回復を検出します。その後、in.mpathd により、障害経路の迂回が行われた元の IP アドレスが hme1 から回復した経路へ戻されます。

検査用 IP アドレスの構成方法についての詳細は、473 ページの「検査用 IP アドレスの構成」 を参照してください。

注 – 障害が発生した物理インタフェースの回復時における、IP アドレスの障害回路の迂回には、3 分かかります。この所要時間は変わる場合があります。つまり、ネットワークトラフィックに応じて、所要時間は異なります。また、所要時間は in.mpathd によって障害経路を迂回し回復した着信インタフェースの安定性によっても異なります。

システムの起動時に存在しない物理インタフェースの回復

この節の手順は、ifconfig(1M) を使用して構成される IP 層だけに適用されます。ATM または他のサービスなど、IP 層よりも上位または下位の層が自動化されていない場合には、手動による特別な手順が必要です。この特別な手順は、事前切り離し時の構成解除および事後接続後の構成を行うために実行します。障害または DR の対処方法については、該当する層およびアプリケーションのマニュアルを参照してください。

NIC の DR 操作後の回復は、Sun Fire プラットフォーム上の IO ボードの一部であり、NIC が PCI デバイスの場合には自動化されます。よって、NIC が DR 操作の一部として戻される場合には、次の手順を行う必要はありません。Sun Fire x800 および Sun Fire 15000 の詳細については、cfgadm_sbd(1M) のマニュアルページを参照してください。物理インタフェースは、/etc/hostname.*interface* ファイルで指定された構成に回復されます。リブートを行っても構成が保持されるように、インタフェースを構成する方法の詳細については、485 ページの「マルチパスインタフェースグループの構成」を参照してください。

注 – 以前の Sun Fire システム (Exx00) の場合には、DR 切り離しは手動で行う必要があります。ただし、DR 接続は自動的に行われます。

システムの起動時に存在しない物理インタフェースを回復するには、次の手順を行なってください。次の手順では、物理インタフェース `hme0` と `hme1` を例として使用します。なお、次の手順では `hme0` と `hme1` の両方のインタフェースが 1 つのマルチパスグループ内にあり、`hme0` はシステムの起動時に存在しないということを想定しています。

注 – 障害が発生した物理インタフェースの回復時における、IP アドレスの障害回路の迂回には、3 分かかります。この所要時間は変わる場合があります。つまり、ネットワークトラフィックに応じて、所要時間は異なります。また、所要時間は `in.mpathd` によって障害経路を迂回し回復した着信インタフェースの安定性によっても異なります。

▼ システムの起動時に存在しない物理インタフェースを回復するには

1. 障害の発生したネットワーク情報を、コンソールログ中の障害エラーメッセージから入手します。

   `syslog(3C)` マニュアルページを参照してください。エラーメッセージは次のように表示されます。

   ```
   moving addresses from failed IPv4 interfaces:
   hme1 (moved to hme0)
   ```

   または、次のようなエラーメッセージが表示されます。

   ```
   moving addresses from failed IPv4 interfaces:
   hme1 (couldn't move, no alternative interface)
   ```

2. システムに物理インタフェースを接続します。

   物理インタフェースの交換方法については、 `cfgadm(1M)` のマニュアルページ、『*Sun Enterprise 10000 DR* 構成マニュアル』、または『*Sun Enterprise 6x00, 5x00, 4x00, 3x00* システム *Dynamic Reconfiguration* ユーザーマニュアル』を参照してください。

3. 手順**1** のエラーメッセージの内容を参照します。アドレスを移動できなかった場合は手順 **5** へ進みます。アドレスが移動された場合は手順 **4** へ進みます。

4. 次の指示に従って、障害迂回処理の一部として構成された論理インタフェースを結合解除します。

a. **/etc/hostname.<moved_from_interface> (**この場合は **/etc/hostname.hme1)** のファイルの内容を見て、障害迂回処理の一部として構成された論理インタフェースを確認してください。

b. 次のコマンドを入力して、各障害迂回 **IP** アドレスを結合解除します。

```
# ifconfig moved_to_interface removeif moved_ip_address
```

注 – 障害迂回アドレスは、failover パラメータが指定されたアドレス、または -failover パラメータが指定されていないアドレスです。-failover が指定された IP アドレスは、結合解除の必要がありません。

たとえば、/etc/hostname.hme0 ファイルの中に次の行が含まれている場合

```
inet 1.2.3.4 -failover up group one
addif 1.2.3.5 failover up
addif 1.2.3.6 failover up
```

各障害迂回 IP アドレスを結合解除するためには、次のコマンドを入力します。

```
# ifconfig hme0 removeif 1.2.3.5
# ifconfig hme0 removeif 1.2.3.6
```

5. 問題となっている各インタフェース用に次のコマンドを入力して、交換した物理インタフェースの **IPv4** 情報を再構成します。

```
# ifconfig removed_from_NIC <parameters>
```

手順 4 の例を使用して、次のコマンドを入力します。

```
# ifconfig hme1 inet plumb
# ifconfig hme1 inet 1.2.3.4 -failover up group one
# ifconfig hme1 addif 1.2.3.5 failover up
# ifconfig hme1 addif 1.2.3.6 failover up
```

マルチパス構成ファイルの構成

マルチパス構成ファイル /etc/default/mpathd で、必要に応じて以下の 3 つのパラメータを調整できます。

- FAILURE_DETECTION_TIME
- FAILBACK
- TRACK_INTERFACES_ONLY_WITH_GROUPS

これらのパラメータについては、482 ページの「マルチパス構成ファイル」を参照してください。

▼ マルチパス構成ファイルを構成するには

1. スーパーユーザーになります。
2. **/etc/default/mpathd** ファイルに対し、次の手順を実行しパラメータの値を変更します。
   a. **FAILURE_DETECTION_TIME** パラメータの新しい値を入力します。

      ```
      FAILURE_DETECTION_TIME=n
      ```

   b. **FAILBACK** パラメータの新しい値を入力します。

      ```
      FAILBACK=[yes | no]
      ```

   c. **TRACK_INTERFACES_ONLY_WITH_GROUPS** パラメータの新しい値を入力します。

      ```
      TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
      ```

3. コマンド行から次のコマンドを入力します。

   ```
   # pkill -HUP in.mpathd
   ```

用語集

この用語集には、このマニュアルで新たに使用した、『*Sun Global Glossary*』にはない用語の説明だけが記載されています。その他の用語の説明については、『*Sun Global Glossary*』を参照してください。

AES Advanced Encryption Standard。対称 128 ビットブロックのデータ暗号技術。米国政府は、2000 年の 10 月に暗号化標準として Rijndael 方式を採用しています。DES に代わる米国政府の標準として、AES が採用されています。

Blowfish 32 ビットから 448 ビットまでの可変長キーの対称ブロックの暗号化アルゴリズム。その作成者である Bruce Schneier 氏は、鍵を頻繁に変更しないアプリケーションに効果的であると述べている。

DES Data Encryption Standard。1975 年に開発され、1981 年に ANSI X.3.92 として ANSI で標準化された対称鍵の暗号化方式。DES では 56 ビットの鍵を使用する。

Diffie-Hellman プロトコル 公開鍵暗号化としても知られている。1976 年に Diffie 氏と Hellman 氏が開発した非対称暗号鍵協定プロトコル。このプロトコルを使用して、セキュリティ保護されていない媒体で事前に秘密鍵を用意しなくても 2 人のユーザーが秘密鍵を交換できます。Diffie-Hellman は、IKE プロトコルで使用されます。

DSA デジタル署名アルゴリズム。512 ビットから 1024 ビットまでの可変長キーの公開鍵アルゴリズム。この場合、入力に SHA-1 を使用する。

HMAC メッセージ認証を行うためのキー付きハッシュ方法。HMAC は秘密共有鍵と併用して、MD5 、SHA-1 などの繰り返し暗号化のハッシュ関数で使用する。HMAC の暗号の強さは、基底ハッシュ関数のプロパティによって異なる。

IKE インターネットキー交換。IKE は、IPsec セキュリティアソシエーションの認証されたキー情報を自動的に提供する。

IPsec IP データグラムを保護するためのセキュリティアーキテクチャ。

IPv4 インターネットプロトコルバージョン 4。IP とも呼ばれる。このバージョンは 32 ビットのアドレス空間を提供する。

IPv6 インターネットプロトコルバージョン 6。このバージョンは 128 ビットのアドレス空間を提供する。

IP 内 **IP** カプセル化 IPv4 パケット内で IPv4 パケットをトンネリングするためのインターネット標準プロトコル

IP リンク リンク層でノード間の通信に使用される通信設備や通信媒体。リンク層は、IPv4/IPv6 のすぐ下にある。 例としては、Ethernet (単一の、またはブリッジされた) または ATM ネットワークがある。1 つまたは複数の IPv4 サブネット番号またはネットワーク接頭辞が IP リンクに割り当てられる。同じサブネット番号またはネットワーク接頭辞を複数の IP リンクに割り当てることはできない。ATM LANE では、IP リンクは 1 つのエミュレートされた LAN である。ARP を使用する場合、ARP プロトコルの有効範囲は単一の IP リンクである。

MD5 デジタル署名などのメッセージ認証に使用する繰り返し暗号化のハッシュ関数。1991 年に Rivest 氏によって開発された。

MTU 最大転送単位。リンクに転送できるサイズ (オクテット単位)。たとえば、Ethernet の MTU は 1500 オクテット。

PKI Public Key Infrastructure。インターネットトランザクションに関係する各関係者の有効性を確認および承認する、デジタル署名、認証局、他の登録機関のシステム。

RSA デジタル署名と公開鍵暗号化システムを取得するための方法。1978 年に最初に公開され、 Rivest氏、Shamir 氏、Adleman 氏によって開発された。

SADB セキュリティアソシエーションデータベース。データの転送に使用する暗号鍵とアルゴリズムを指定するテーブル。

SHA-1 アルゴリズム セキュリティ保護されたハッシュアルゴリズム。メッセージ要約を作成するために 2^{64} 文字以下の長さを入力するときに操作する。これは DSA への入力となる。

SPI セキュリティパラメータインデックス。受信したパケットの暗号解除に受信側が使用する SADB の行を指定する整数。

Triple-DES Triple-Data Encryption Standard。168 ビットの鍵を提供する対称鍵暗号化方法。

アドレスプール ホームアドレスを必要とするモバイルノードが利用する、ホームネットワーク管理者によって指定された一連のアドレス

移動先ネットワーク モバイルノードのホームネットワーク以外の、現在接続しているネットワーク

エージェント通知 ホームエージェントおよび外来エージェントが、モバイルノードがリンク上に存在することを通知するために定期的に送信するメッセージ

エージェント発見	モバイルノードが移動している場合は、自分の現在の場所および外部ネットワーク上での自分の気付アドレスを決定すること
回復検出	障害の発生後、NIC や NIC から第 3 層の装置への経路が、正しく動作し始めたことを検出する処理
回復した経路への復帰	ネットワークアクセスを、回復が検出されたインタフェースに戻す処理
外部ネットワーク	モバイルノードのホームネットワーク以外のネットワーク
外来エージェント	モバイルノードが移動する外部ネットワーク上のルーターまたはサーバー
鍵管理	セキュリティアソシエーションを管理するための手法。
仮想プライベートネットワーク **(VPN)**	インターネットのような公共ネットワーク内でトンネルを利用する、単独の、安全で論理的なネットワーク。
カプセル化	ヘッダーとペイロードを 1 番目のパケット内に配置し、そのパケットを 2 番目のパケットのペイロード内に配置すること。
カプセル化セキュリティヘッダー	データグラムに対して認証と完全性を提供する拡張ヘッダー。
気付アドレス	モバイルノードの一時的アドレス。モバイルノードを外来ネットワークに接続するとき、トンネル出口として使用する。
逆方向トンネル	モバイルノードの気付アドレスで始まり、ホームエージェントで終わるトンネル
近傍検索	接続されているリンク上にある他のホストをホストが特定できるようにするための IP メカニズム。
近傍通知	近傍要請メッセージに対する応答、またはデータリンク層アドレスの変更を通知するために、ノードが自発的に近傍通知メッセージを送ること。
近傍要請	近傍のリンク層アドレスを決定するために、ノードによって送信される要請。また、キャッシュされたリンク層アドレスによって近傍が到達可能であるかを確認する。
結合テーブル	ホームアドレスを、残りの有効期間と与えられた時間を含む気付アドレスに関連付けるホームエージェント表
公開鍵暗号化	相互に認識している公開鍵とメッセージの受信側だけが認識している非公開鍵の 2 つの鍵を使用する暗号化システム。IKE により、IPsec の公開鍵が提供される。
最小カプセル化	ホームエージェント、外来エージェント、およびモバイルノードによってサポートされる任意の形態の IPv4 内 IPv4 トンネリング。最小カプセル化は、IP 内 IP カプセル化よりも 8 ないし 12 バイト少ないオーバヘッドしか持たない。
サイトローカルアドレス	単一サイト上でアドレスを指定するために使用する。

自動設定	IPv6 において、ホストが自身のインタフェースを自動的に設定すること。
順方向トンネル	ホームエージェントから始まり、モバイルノードの気付アドレスで終わるトンネル
障害経路の迂回	ネットワークアクセスを障害が検出されたインタフェースから正常な物理インタフェースに切り替える処理。ネットワークアクセスには、IPv4 のユニキャスト、マルチキャスト、およびブロードキャストと、IPv6 のユニキャストとマルチキャストが含まれる。
障害検出	障害の発生後、NIC や NIC から第 3 層の装置への経路が、正しく動作し始めたことを検出する処理
ステートフル自動設定	ホストが、インタフェースアドレスや設定情報、およびパラメータをサーバーから取得すること。
ステートレス自動設定	ホストが、ローカルに入手可能な情報と、ルーターが通知した情報を組み合わせて自身のアドレスを生成すること。
セキュリティアソシエーション **(SA)**	1 つのホストから別のホストにセキュリティ属性を指定するアソシエーション
セキュリティパラメータインデックス **(SPI)**	受信したパケットを復号化するために使用する、SADB (セキュリティアソシエーションデータベース) 内の行を特定する整数値。
専用アドレス	インターネット経由で経路指定ができない IP アドレス。
双方向トンネル	双方向にデータグラムを送信するトンネル。
待機	グループ内の他の物理インタフェースに障害が発生するまでデータの伝送には使用されない物理インタフェース
対称鍵暗号化	メッセージの暗号解除に、メッセージの送受信側が単一の共通鍵を使用する暗号化システム。対称鍵は、IPsec での大量データ転送の暗号化に使用する。対称鍵システムの一例として DES がある。
デジタル署名	送信側を一意に識別する、電子的に転送されたメッセージに添付されるデジタルコード。
デュアルスタック	IPv6 への移行に使用する、IPv4 と IPv6 の両機能を併せ持つプロトコルスタックで、スタックの残り部分は同じ。
登録	モバイルノードが、ホームにないときに自分の気付アドレスを自分のホームエージェントおよび外来エージェントに登録すること。
トンネリング	IPv6 パケットを IPv4 パケット内に組み込み、IPv4 ルーターを経由して配送するメカニズム。この用語は IPv6 に限定される。
トンネル	カプセル化される間データグラムが通過するパス。
任意キャストアドレス	(一般的に別のノードに属す) 複数のインタフェースに割り当てられる IP アドレス。任意キャストアドレスに送られたパケットは、そのアド

	レスを持つ、プロトコルに基づき「最も近い」インタフェースに配送される。パケットの経路指定は、経路指定プロトコルの距離測定に応じて決定される。
認証局 **(CA)**	デジタル署名および公開鍵と非公開鍵のペアの作成に使用するデジタル証明書を発行する、公証された第三者機関または企業。CA は、一意の証明書を付与された個人が当該の人物であることを保証する。
認証ヘッダー	IP データグラムに対し認証と完全性を提供する拡張ヘッダー。機密性は提供されない。
ネットワークアクセス識別子 **(NAI)**	user@domain 形式でモバイルノードを一意に特定するために使用する。
ネットワークインタフェースカード **(NIC)**	リンクとのインタフェースになる、内部ネットワークアダプタまたは独立したネットワークアダプタカード。
ノード	ホストまたはルーター。
パケット	通信回線上で、1 単位として送られる情報の集合。ヘッダーとペイロードで構成される。
ハッシュ値	テキストの文字列から生成される数値。ハッシュ関数は、転送されるメッセージが改ざんされないようにするために使用する。1 方向のハッシュ関数の一例としては、MD5 および SHA-1 がある。
汎用経路指定カプセル化 **(GRE)**	ホームエージェント、外来エージェント、およびモバイルノードによってサポートされる任意の形態のトンネリング。他の任意の (または同じ) ネットワーク層プロトコルの配信パケット内で任意のネットワーク層プロトコルのパケットをカプセル化できるようにする。
ビジターリスト	ある外来エージェントに移動しているモバイルノードのリスト
非対称鍵暗号化	メッセージの送受信側で異なる鍵を使用してメッセージの暗号化および暗号解除を行う暗号化システム。非対称鍵を使用して、対称鍵暗号に対するセキュリティ保護されたチャネルを作成する。非対称鍵プロトコルの一例には、Diffie–Hellman がある。対称鍵暗号化と対比。
ファイアウォール	組織内の私的ネットワークまたはイントラネットを、インターネットなどの外部ネットワークからの侵入に対して保護する装置またはソフトウェア。
物理インタフェース	リンクに対するノードの接続。この接続は通常、デバイスドライバとネットワークアダプタとして実装される。ネットワークアダプタによっては、qfe のように複数の接続点を持つ場合もある。このマニュアルでは、「ネットワークアダプタ」は「単一接続点」を示す。"
物理インタフェースグループ	同じリンクに接続されている、システムの物理インタフェース群。グループ内のすべての物理インタフェースには、識別のための空文字列でない同じ名前が割り当てられる。
物理インタフェースグループ名	グループを識別する、物理インタフェースに割り当てられる名前。この名前の有効範囲は 1 つのシステム。同じグループ名を共有する複数の物理インタフェースは、物理インタフェースグループを構成する

ホームアドレス	モバイルノードに長期間割り当てられた IP アドレス。このアドレスは、インターネットあるいは企業ネットワークに接続されたときにも変更されない。
ホームエージェント	モバイルノードのホームネットワーク上のルーターまたはサーバー。
ホームネットワーク	モバイルノードのホームアドレスのネットワーク接頭辞と一致するネットワーク接頭辞を持つネットワーク
ホップ	2 つのホストを分離するルーターの数を判別するための手段。たとえば、始点ホストと終点ホストが 3 つのルーターで分離されている場合、ホストは互いに 4 ホップ離れている、という。
マルチキャストアドレス	特定の方法でインタフェースのグループを特定する IP アドレス。マルチキャストアドレスに送信されるパケットは、グループにあるすべてのインタフェースに配信される。
モバイルノード	自分の IP ホームアドレスを使用して既存の通信をすべて維持しながら、接続点をネットワーク間で変更するホストまたはルーター
モビリティエージェント	ホームエージェントまたは外来エージェント
モビリティ結合	ホームアドレスと気付アドレスとを関連付ける。その関連付けの残りの有効期間も含む。
モビリティセキュリティアソシエーション	認証アルゴリズムのような、ノード間のセキュリティ対策の集合。2 つのノード間で交換されるモバイル IP プロトコルメッセージに適用される。
ユニキャストアドレス	単一のインタフェースを指示する IP アドレス。
リダイレクト	特定の終点に到達するために、ホストに対して最適な最初のホップノードを、ルーターが通知すること。
リンクローカル使用アドレス	自動アドレス設定などのために、単一リンク上でアドレスを指定するために使用する。
ルーター通知	ルーターが、各種のリンクパラメータおよびインターネットパラメータと共に、その存在を定期的にあるいはルーター要請メッセージに応じて通知すること。
ルーター発見	ホストが、接続されているリンク上にあるルーターを特定すること。
ルーター要請	ホストがルーターに対し、次に予定されている時刻ではなく、ただちにルーター通知メッセージを送信するように要求すること。
ローカル使用アドレス	ローカルの経路指定可能な範囲だけを対象とするユニキャストアドレス (サブネット内またはネットワーク内)。 また、ローカルまたはグローバルな一意の範囲を対象とすることもできます。

索引

C

D

E

F

G

H

O

P

Q

R

U

V

W

あ

き

く

け

こ

さ

し

す

せ

そ

た

つ

て

と

な

に

ね

の

は

ゆ

ら

り