Solaris のシステム管理 (第 3 巻)

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part Number 806-2719–10
2000 年 3 月

原典: *System Administration Guide, Volume 3*

Part No: 806-0916-10

Revision A

Adobe PostScript

目次

はじめに

『*Solaris* のシステム管理 *(*第 *3* 巻*)*』は、Solaris™ システム管理に関する重要な情報を提供する、3 巻構成のマニュアルの第 3 巻です。このマニュアルでは、SunOS™ 5.8 オペレーティングシステムがすでにインストールされており、使用する予定のネットワークソフトウェアが設定済みであることを前提としています。SunOS 5.8 オペレーティングシステムは、Solaris 製品ファミリの 1 つで、Solaris 共通デスクトップ環境 (CDE) を含む数多くの機能を備えています。SunOS 5.8 オペレーティングシステムは、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 IA の 2 種類のハードウェア (プラットフォーム) 上で稼働します。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。章、節、注意、箇条書きの項目、図、表、例、コード例において特に明記されないかぎり、本書の情報は SPARC と IA の両方のプラットフォーム、さらに 64 ビットと 32 ビットの両方のアドレス空間に適用されます。

対象読者

本書は、Solaris 8 リリースを稼働するシステムの管理者を対象としています。本書を活用するためには、1 - 2 年程度の UNIX システムの管理経験が必要です。UNIX システムの管理トレーニングコースへの参加も役立つでしょう。

『*Solaris* のシステム管理』全 3 巻の概要

『Solaris のシステム管理』 全 3 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 *(*第 *1* 巻*)*』

- 「ユーザーアカウントとグループの管理」
- 「サーバーとクライアントサポートの管理」
- 「システムのシャットダウンとブート」
- 「取り外し可能な媒体の管理」
- 「ソフトウェアの管理」
- 「デバイスの管理」
- 「ディスクの管理」
- 「ファイルシステムの管理」
- 「データのバックアップと復元」

『*Solaris* のシステム管理 *(*第 *2* 巻*)*』

- 「印刷サービスの管理」
- 「リモートシステムの利用」
- 「端末とモデムの管理」
- 「システムセキュリティの管理」
- 「システム資源の管理」
- 「システム性能の管理」
- 「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 *(*第 *3* 巻*)*』

- 「ネットワークサービストピック」
- 「IP アドレス管理トピック」

- 「モデム関連ネットワークサービスのトピック」
- 「リモートファイルシステムへのアクセスについてのトピック」
- 「メールサービスについてのトピック」
- 「ネットワークサービスの監視についてのトピック」

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。

- 『*Solaris* ネーミングの管理』
- 『*Solaris* ネーミングの設定と構成』
- 『*Solaris* のシステム管理 *(*第 *1* 巻*)*』
- 『*Solaris* のシステム管理 *(*第 *2* 巻*)*』
- 『*UNIX Communications*』 Anderson、Bart、Bryan Costales、Harry Henderson 著、Howard W.Sams & Company、1987
- 『*Firewalls and Internet Security*』 Cheswick、Williams R.、Steven M.Bellovin 著、Addison Wesley、1994
- 『*sendmail*、*Second Edition*』 Costales、Bryan 著、O'Reilly & Associates,Inc.、1997
- 『*A Directory of Electronic Mail Addressing and Networks*』 Frey、Donnalyn、Rick Adams 著、O'Reilly & Associates,Inc.、1993
- 『*The Whole Internet User's Guide and Catalog*』 Krol、Ed. 著、O'Reilly & Associates,Inc.、1993
- 『*Managing UUUP and Usernet*』 O'Reilly、Tim、Grace Todino 著、O'Reilly & Associates,Inc.、1992
- 『*TCP/IP Illstrated, Volume 1*』 Stevens、W.Rechard 著、Addison Wesley、1994

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、http://www1.fatbrain.com/documentation/sun の Sun Documentation Center をご覧ください。

Sun のオンラインマニュアル

http://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。

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。

表 P–1 表記上の規則 続く

字体または記号	意味	例
「 」	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'`

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

- C シェルプロンプト

```
system% command y|n [filename]
```

- Bourne シェルおよび Korn シェルのプロンプト

```
system$ command y|n [filename]
```

- スーパーユーザーのプロンプト

```
system# 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用しています。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用し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、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

第 1 章

ネットワークサービストピック

第 2 章	ネットワークサービスの概要

第 2 章

ネットワークサービスの概要

この章では、本リリースのネットワークサービスにおける変更事項を紹介し、さらに、ネットワーク管理者の役割についても言及します。この章の内容は、新たにネットワーク管理者になった方にとっては、ネットワーク管理者が行うべき作業の概略を理解するのに役立ちます。またこの章では、本書を読み進める上で知っていなければならないネットワークの基本概念についても説明します。すでに経験を積んでいるネットワーク管理者の方は、最初の節を読んで内容を確認すれば、以降の節は飛ばしてもかまいません。この章で説明する各トピックは次のとおりです。

Solaris 8 リリースの新機能

このリリースに含まれる新しい機能を表 2–1 に示します。

表 2–1 Solaris 8 リリースの新機能

テクノロジー	説明	参照先
DHCP	ダイナミックホスト構成プロトコル (DHCP) を使用すると、システム管理者が事前に設定を行わなくても、ホストは IP アドレスとその他のシステム設定情報を取得できる。Solaris 8 リリースでは、DHCP サーバーとデータベースを設定、管理するための Java™ ベースのグラフィカルインタフェースが追加されている	第 8 章
IPsec	IP セキュリティアーキテクチャ (IPSec) は、IP データグラムの保護機能を提供する。保護機能には、機密性、強力なデータ完全性、部分シーケンス整全性 (繰り返し攻撃に対する保護)、データの認証がある。IPSec は IP プロセシング中に実行され、インターネットアプリケーションとは独立して適用できる	第 18 章
IPv6	IPv6 は、インターネットプロトコル (IP) の新しいバージョンで、現行バージョンの IPv4 の機能を大幅に拡張している。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させることなく IPv6 ネットワークを展開できる。さらに、IPv6 は新しいインターネット機能を提供するプラットフォームとしても機能する	第 14 章
LLC2	Class II Logical Link Control (論理リンク制御) (llc2) ドライバは、サポートされている通信アダプタのいずれかによって制御される物理 LAN ネットワークと、Solaris オペレーティング環境下で実行されるネットワークソフトウェア (NetBIOS、SNA、OSI など) とのインタフェースをとる	45ページの「新バージョンの Logical Link Control ドライバ」
NFS™ ロギング	NFS ロギングにより、NFS サーバーにトランザクションログを記録する機能が追加される	605ページの「NFS サーバーログ」
Sendmail	Solaris メールサービスは、`sendmail` バージョン 8.9.3 に基づいている	731ページの「`sendmail` の新しい機能」
NCA	Solaris Network Cache and Accelerator により、Web サーバーの性能が向上している	46ページの「Solaris Network Cache and Accelerator (NCA)」

新バージョンの Logical Link Control ドライバ

Class II Logical Link Control (論理リンク制御) (llc2)ドライバ は、サポートされている通信アダプタのいずれかによって制御される物理 LAN ネットワークと、Solaris オペレーティング環境下で実行されるネットワークソフトウェア (NetBIOS、SNA、OSI など) とのインタフェースをとります。ネットワークソフトウェアに対するドライバとしての役割を果たす llc2 ドライバは、カーネルに常駐し、標準の UNIX のストリーム関数を介してアクセスされます。

このバージョンの llc2 ドライバは、適切な Solaris MAC レイヤードライバを介してアクセスした場合、Ethernet、トークンリング、FDDI アダプタ用のコネクションレス型とコネクション型の両方の llc2 オペレーションをサポートします。llc2 ドライバへのデータリンクプロバイダインタフェース (DLPI) を使用すると、複数の異なるプロトコルスタック (NetBIOS や SNA) が、1 つまたは複数のローカルエリアネットワーク上で同時に動作可能になります。

デフォルトで llc2 ドライバを起動するには、`/etc/llc2/llc2_start.default` ファイルの名前を `/etc/llc2/llc2_start` に変更します。これにより、`/etc/init.d/llc2/rc2.d/S40llc2` スクリプトが、各 ppa インタフェース用の構成ファイルを `/etc/llc2/default/llc2.*` 内に構築し、各インタフェース上で llc2 が起動できるようになります。構成ファイルの動作確認をするには、手動で `/usr/lib/llc2/llc2_autoconfig` を実行します。

llc2 ドライバについての詳細は、IEEE 規格 802.2 および `llc2(7)` のマニュアルページを参照してください。

`llc2` ファイルには、基本となる MAC レイヤードライバへの適切なリンクを確立するために、LLC2 で必要となる情報と、そのリンク用の LLC (Logical Link Control) Class II ステーションコンポーネントストラクチャを構築するためのパラメータ情報が格納されています。`llc2` 構成ファイルについての詳細は、`llc2(4)` のマニュアルページを参照してください。

`llc2_autoconfig` ユーティリティは、LLC2 構成ファイル (`/etc/llc2/default/llc2.*`) を生成するために使用されます。`/etc/llc2_default/` に構成ファイルが存在しない場合、このユーティリティは、システム内の使用可能なインタフェースをすべて検出し、インタフェースのためのデフォルトの構成ファイルを生成します。

`/etc/llc2_default/` に構成ファイルが存在する場合、`llc2_autoconfig` ユーティリティは、ファイル内に定義されているインタフェースが今も存在するかどう

かをチェックします。インタフェースがシステム内にない場合、ユーティリティは、ファイルの llc2_on を 0 に設定します。その後で、システム内に新しいインタフェースが存在するかどうかチェックします。新しいインタフェースがあれば、そのための構成ファイルを生成します。llc2_autoconfig ユーティリティについての詳細は、llc2_autoconfig(1) のマニュアルページを参照してください。

llc2_config ユーティリティは、LLC2 サブシステムを起動または終了したり、LLC2 インタフェースパラメータを設定するために使用されます。llc2_config ユーティリティについての詳細は、llc2_config(1) のマニュアルページを参照してください。

ドライバ、アダプタ、ネットワークをテストするには、llc2_loop ループバック診断コマンドを使用します。llc2_loopコマンドについての詳細は、llc2_loop(1M) のマニュアルページを参照してください。

llc2_stats コマンドを使用すると、LLC2 ドライバのホストベース、Class 2 Logical Link Control コンポーネントからの統計情報を取り出すことができます。統計情報は、ステーション、サービスアクセスポイント (SAP)、接続コンポーネントについて保持されています。llc2_stats コマンドについての詳細は、llc2_stats(1) のマニュアルページを参照してください。

Solaris Network Cache and Accelerator (NCA)

Solaris Network Cache and Accelerator (NCA) は、HTTP 要求時にアクセスされた Web ページのカーネル内部キャッシュを維持することによって、Web サーバーの性能を向上させます。NCA は、自身で要求を処理するか、あるいは処理のために Web サーバーに要求を渡すことにより、カーネル内で HTTP を完璧にサポートします。

この製品は、専用の Web サーバー上で実行するようにします。NCA を実行しているサーバー上で別の大規模なプロセスを実行すると、障害が生じる可能性があります。

NCA 機能には次の 2 つのコンポーネントが必要です。

- カーネルモジュール: ncakmod
- Web サーバー: httpd

ncakmod は、Solaris door ライブラリ (door_create(3DOOR) を参照) を介して、Web サーバーである httpd と通信します。Solaris door ライブラリは、同一ホスト上のプロセス間、および、カーネルとユーザープロセスの間で、高速かつ信頼性の高い同期式 RPC メカニズムを提供します。

ncakmod と httpd の間の通信に使用されるプロトコルは、Solaris door のリモートプロシージャコール (RPC) インタフェースを使用した、同期式の要求-応答型プロトコルです。door の RPC は、NCA のカーネル内で呼ばれ、同期がとられます。データは双方向に転送されます。つまり、各 door の RPC 内で、NCA から http サーバーと、http サーバーから NCA の両方向に転送されます。ncakmod は HTTP 要求を httpd に渡し、httpd は door インタフェースを介した要求に応答を返します。これによって、acceptx と sendfile を使用した場合と同様の機能が提供されます。

NCA ロギングは、HTTP 要求データをバイナリ形式でディスクに書き込みます。NCA 機能では、CLF (共通ログ形式) ログファイル形式も使用できます。

表 2–2 Solaris NCA の管理作業

作業	説明	参照先
NCA を有効にする	Web サーバー上の Web ページのカーネル内部キャッシュを有効にする手順を実行する	47ページの「NCA を有効にする方法」
NCAを無効にする	Web サーバー上の Web ページのカーネル内部キャッシュを無効にする手順を実行する	49ページの「NCA を無効にする方法」
NCA ロギングの変更	NCA ロギングプロセスを有効または無効にする手順を実行する	50ページの「NCA ロギングを有効または無効にする方法」

▼ NCA を有効にする方法

1. スーパーユーザーになります。

2. インタフェースを登録します。

   /etc/nca/nca.if ファイルに各物理インタフェースの名前を登録します (詳細は、nca.if(4) のマニュアルページを参照)。

```
# cat /etc/nca/nca.if
hme0
```

(続く)

```
hme1
```

インタフェースごとに、対応する hostname.*interface-name* ファイル と、hostname.*interface-name* の内容と一致するエントリが /etc/hosts ファイル内になければなりません。すべてのインタフェースで NCA 機能を使用可能にするには、nca.if ファイル内でアスタリスク (*) を指定します。

3. ncakmod カーネルモジュールを有効にします。

/etc/nca/ncakmod.conf 内の status エントリを enabled に変更します。

```
# cat /etc/nca/ncakmod.conf
#
# Copyright (c) 1998-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)ncakmod.conf       1.1     99/08/06 SMI"
#
# NCA Kernel Module Configuration File
#
status=enabled
httpd_door_path=/var/run/nca_httpd_1.door
```

詳細は、ncakmod.conf(4) のマニュアルページを参照してください。

4. NCA ロギングを有効にします。

/etc/nca/ncalogd.conf 内の status エントリを enabled に変更します。

```
# cat /etc/nca/ncalogd.conf
#
# Copyright (c) 1998-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)ncalogd.conf       1.1     99/08/06 SMI"
#
# NCA Log Daemon Configuration File
#
status=enabled
logd_path_name="/var/nca/log"
```

(続く)

```
logd_file_size=1000000
```

logd_path_name エントリに示されているパスを変更すると、ログファイルの格納場所を変更できます。詳細は、ncalogd.conf(4) のマニュアルページを参照してください。

5. **IA の場合のみ:** 仮想メモリーサイズを増やします。

   eeprom コマンドを使用して、システムの kernelbase を設定します。

```
# eeprom kernelbase=0x900000000
# eeprom kernelbase
kernelbase=0x900000000
```

2 行目の eeprom コマンドを実行すると、パラメータが設定済みかどうかを確認できます。

注 - NCA を有効にすると、ユーザープロセスで使用可能な仮想メモリ容量が 3G バイト未満に削減されます。このため、システムは ABI に準拠しなくなります。この場合、システムの起動時に、システムがABI に準拠していないことを知らせる警告メッセージが表示されます。ほとんどのプログラムは、実際には 3G バイトの仮想アドレス空間を必要としません。3G バイトの仮想アドレス空間を必要とするプログラムは、NCA が有効に設定されていないシステムで実行する必要があります。

---

6. サーバーを再起動します。

▼ NCA を無効にする方法

1. スーパーユーザーになります。

2. ncakmod カーネルモジュールを無効にします。

/etc/nca/ncakmod.conf 内の status エントリを disabled に変更します。

```
# cat /etc/nca/ncakmod.conf
#
# Copyright (c) 1998-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)ncakmod.conf       1.1     99/08/06 SMI"
#
# NCA Kernel Module Configuration File
#
status=disabled
httpd_door_path=/var/run/nca_httpd_1.door
```

詳細は、ncakmod.conf(4) のマニュアルページを参照してください。

3. **NCA** ロギングを無効にします。

/etc/nca/ncalogd.conf 内の status エントリを disabled に変更します。

```
# cat /etc/nca/ncalogd.conf
#
# Copyright (c) 1998-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)ncalogd.conf       1.1     99/08/06 SMI"
#
# NCA Log Daemon Configuration File
#
status=disabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
```

詳細は、ncaklogd.conf(4) のマニュアルページを参照してください。

4. サーバーを再起動します

▼ NCA ロギングを有効または無効にする方法

NCA が有効になっていると、必要に応じて NCA のログ処理のオンまたはオフを切り替えることができます (詳細は、47ページの「NCA を有効にする方法」を参照)。

1. スーパーユーザーになります。

2. **NCA** のログ処理のオンとオフを切り替えます。

   ロギングを恒久的に無効にする場合は、/etc/nca/ncalogd.conf 内の status を disabled に変更し、システムを再起動する必要があります。詳細は、ncalogd.cof(4) のマニュアルページを参照してください。

   a. ロギングを停止するには、次のコマンドを入力します。

```
# /etc/init.d/ncalogd stop
```

   b. ロギングを開始するには、次のコマンドを入力します。

```
# /etc/init.d/ncalogd start
```

NCA ファイル

NCA 機能をサポートするにはいくつかのファイルが必要です。ほとんどのファイルは ASCII 形式ですが、バイナリ形式のファイルもあります。表 2–3 に、必要なファイルを一覧します。

表 2–3 NCA ファイル

ファイル名	機能
/etc/hostname.*	サーバー上に構成されるすべての物理インタフェースの一覧が記述されている
/etc/hosts	サーバーに割り当てられるすべてのホスト名の一覧が記述されている。NCA が機能するには、このファイルの各種エントリが、/etc/hostname.* ファイル内のエントリと一致していなければならない
/etc/init.d/ncalogd	サーバーの起動時に NCA 起動スクリプトが実行される
/etc/nca/nca.if	NCA が実行されるインタフェースの一覧が記述されている (nca.if(4) のマニュアルページを参照)
/etc/nca/ncakmod.conf	NCA 用の構成パラメータの一覧が記述されている (ncakmod.conf(4) のマニュアルページを参照)
/etc/nca/ncalogd.conf	NCA ロギング用の構成パラメータの一覧が記述されている (ncalogd.conf(4) のマニュアルページを参照)

表 2–3　NCA ファイル　続く

ファイル名	機能
/usr/bin/ncab2clf	ログファイル内のデータを共通ログファイル形式 (CLF) に変換するためのコマンド (ncab2clf(1) のマニュアルページを参照)
/var/nca/log	ログファイルデータを保持します。ファイルはバイナリ形式なので、編集しないようにする

Perl5

汎用プログラミング言語である、Practical Extraction and Report Language (Perl) 5.005_03 は、一般にフリーソフトウェアとして入手可能なツールですが、Solaris 8 リリースには付属しています。Perl は、プロセス、ファイルおよびテキストの処理機能に優れ、グラフィックス、ネットワーク、World Wide Web (WWW) プログラミングなどの複雑なシステム管理作業を行うための標準開発ツールとして広く使用されています。

Perl5 には、動的にロード可能なモジュールフレームワークが含まれています。このモジュールフレームワークを使用すると、特定の作業に新しい機能を追加することができます。多くのモジュールが、http://www.cpan.org の Comprehensive Perl Archive Network (CPAN) から自由に入手できます。Solaris Perl のインストール時にシステムに組み込まれるコアモジュールには、CGI、NDBM_File、Getopt などがあります。これらのモジュールは、/usr/per15/5.00503 ディレクトリに置かれます。site_perl ディレクトリは初期設定では空の状態で、ローカルにインストールされた Perl5 モジュールを格納するためのディレクトリです。

詳細は、perldoc コマンドを次のように実行して、/usr/perl5/pod ディレクトリ内の Perl pod (ポータブルドキュメンテーション) を調べてください。

```
% cd /usr/perl5/pod
% /usr/perl5/bin/perldoc perlfaq1.pod
```

ネットワーク管理者の責任

一般に、ネットワーク管理者の作業には次の分野があります。

- ネットワークの設計と計画
- ネットワークの設定
- ネットワークの保守
- ネットワークの拡張

各作業分野は、ネットワークのライフサイクルの中の各段階に対応しています。ネットワーク管理者は、これらのすべての段階に責任を持つ場合もあり、また、ネットワークの保守など特定の分野だけを専門的に受け持つ場合もあります。

ネットワークの設計

ネットワーク管理の最初の作業として、まずネットワークの設計という作業がありますが、一般にこれはネットワーク管理の初心者が行う作業ではありません。ネットワークの設計では、組織のニーズを最大限に満たすようなネットワークの種類を選定する必要があります。大規模の組織では、熟練したネットワーク設計者、つまりネットワークのソフトウェアとハードウェアの両方を熟知している経験豊富なネットワーク管理者が、この作業を担当します。

ネットワーク設計に関連する各種の要素については、第 5 章で説明します。

ネットワークの設定

新しいネットワークの設計が終わったら、次にネットワークの設定と構成という作業を行います。この段階では、ネットワークの物理的な部分を形成するハードウェアをインストールし、ファイルまたはデータベース、ホスト、ルーター、ネットワーク構成サーバーを構成します。

この作業は、ネットワーク管理者の主な責任のうちの 1 つです。組織が非常に大規模ですでに十分なネットワーク構造が整っている場合を除いて、必須な作業の 1 つです。

ネットワークの設定に関連する作業については、第 6 章で説明しています。

ネットワークの保守

ネットワーク管理作業の第 3 の段階には、管理者の責任のもっとも大きい部分を占める、次のような日常的な作業が含まれます。

- ネットワークへの新規マシンの追加
- ネットワークセキュリティの管理
- NFS、ネームサービス、電子メールなどのネットワークサービスの管理
- ネットワーク上の問題に関する障害追跡

111ページの「ネットワーククライアントの構成」では、既存のネットワーク上での新規ホストの設定方法について説明します。120ページの「一般的な障害追跡方法」では、ネットワーク上の問題を解決するためのヒントを示します。ネットワークサービスについての詳細は、第 29 章、第 33 章、『*Solaris* ネーミングの管理』、『*NIS+* への移行』を参照してください。セキュリティに関連する管理作業については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』を参照してください。

ネットワークの拡張

ネットワークが安定し問題なく動作する期間が長くなるにつれて、ネットワークの機能とサービスの拡張を望む組織の要求が大きくなってきます。始めのうちは、新しいホストを追加することによってネットワーク人口を増やし、共有ソフトウェアを追加することによってネットワークサービスを拡張することができます。しかし最終的には、単一のネットワークではこれ以上効率的に運営できないような限界点に達することになります。そのようになったとき、ネットワーク管理作業の第 4 の段階である拡張作業にとりかかります。

ネットワークの拡張については、以下のように選択肢がいくつかあります。

- 新規のネットワークを設定し、ルーターとして機能するマシンを使用してそのネットワークを既存のネットワークに接続して、インターネットワークを作る。
- 家庭やリモートオフィスにあるマシンを構成し、それらのマシンが電話回線を介してネットワークに接続できるようにする。
- ネットワークをインターネットに接続して、ネットワークのユーザーが、世界中の他のシステムから情報を検索できるようにする。
- UUCP 通信の構成を行い、リモートマシンとの間でファイルや電子メールをやりとりできるようにする。

114ページの「ルーターの構成」では、インタネットワークの設定手順について説明します。443ページの「PPP によるネットワークの拡張」では、可搬コンピュータのためのネットワーク接続の設定方法について説明します。第 25 章では、UUCP を使用して、使用しているマシンと他の UUCP システムとの間で情報交換を行う方法について説明します。

TCP/IP とは

ネットワーク通信プロトコルは、ネットワークの中でソフトウェアとハードウェアがどのように対話するかを規定した正規の規則です。ネットワークが正しく機能するためには、情報が目的の宛先に明瞭な形式で伝送される必要があります。ネットワークが機能するためには異なる種類のネットワーク用のソフトウェアとハードウェアが相互に対話できる必要があることから、通信プロトコルという概念が開発されました。

Solaris オペレーティングシステムには、組織でのネットワーク管理に必要なソフトウェアが含まれています。このネットワークソフトウェアは、総称的に TCP/IP (Transmission Control Protocol/Internet Protocol) と呼ばれる通信プロトコル群を実装しています。TCP/IP は、多くの主要な国際標準化機構によって標準として認定されており、世界中で使用されています。TCP/IP は複数の規格を集まりなので、多くの異種コンピュータで実行することができます。またこれを用いることによって、Solaris オペレーティングシステムを実行する異機種のシステムが混在したネットワークを容易に設定することができます。

TCP/IP は、多くの異なる種類のコンピュータ、オペレーティングシステム、ネットワークに対して、サービスを提供します。TCP/IP は、Ethernet、FDDI、トークンリングなどのローカルエリアネットワークや、T1 (デジタル専用線)、X.25、ATM などの広域ネットワークに、適用することができます。

TCP/IP を使用することで、複数のローカルエリアネットワークから成る 1 つのネットワークを構築できます。また、TCP/IP を使用すれば、事実上どのようなポイントツーポイントデジタル回線を使用しても、広域ネットワークを構築することができます。

TCP/IP とそのプロトコル群については、第 4 章で詳しく説明します。

Solaris ネットワークを形成するハードウェアの種類

ローカルエリアネットワーク (LAN) という用語は、たとえば 1 つのビル内または 2 つの隣接するビル間のように、比較的狭い空間に限定されている単一のコンピュータネットワークを指します。ローカルエリアネットワークには、ハードウェアとソフトウェアの両方の構成要素があります。ハードウェアの観点から見ると、基本的な Solaris LAN は、ローカルエリアのなんらかのネットワークメディアに接続された複数のコンピュータで構成されます。

ローカルエリアネットワークメディア

コンピュータネットワーク用に使用するケーブル配線や電気配線をネットワークメディアと言います。図 2–1 は、Ethernet メディアを介して相互に接続されている 4 つのコンピュータを示しています。Solaris LAN 環境で最もよく使用されているローカルエリアネットワークメディアは、イーサーネットメディアです。Solaris LAN で使用できるその他のローカルエリアネットワークメディアには、FDDI とトークンリングがあります。

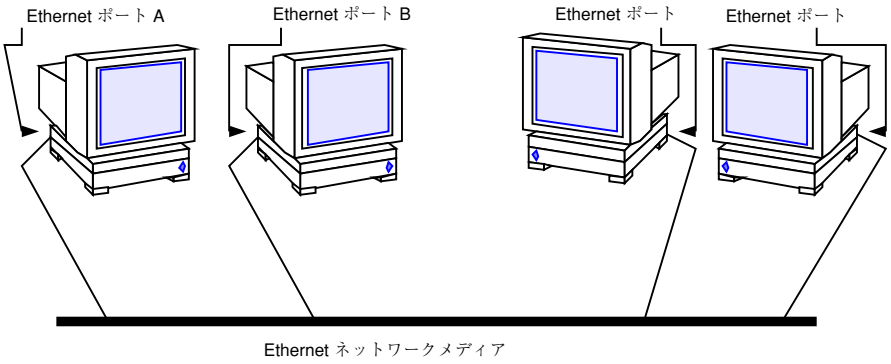


図 2–1　Solaris ローカルエリアネットワーク

コンピュータとそのコネクタ

TCP/IP ネットワーク上のコンピュータは、ネットワークメディアに接続するために 2 種類のコネクタを使用します。それは、シリアルポートと、ネットワークインタフェース上のポートです。

シリアルポート

どのコンピュータにも、少なくとも 2 つのシリアルポートがあり、コンピュータにプリンタやモデムを接続するためのコネクタとして使用されます。シリアルポートは CPU ボードに装備されている場合もありますが、新たに購入しなければならない場合もあります。システムにモデムを接続して PPP 接続や UUCP 接続を確立するときは、これらのポートを使用します。PPP と UUCP はネットワークメディアとして電話回線を使用することができるので、事実上の広域ネットワークサービスを提供します。

ネットワークインタフェース

ネットワークへの接続ができるようにするためにコンピュータに内蔵されているハードウェアを、ネットワークインタフェースと言います。多くのコンピュータにはネットワークインタフェースが始めからインストールされていますが、そうでない場合は、別にネットワークインタフェースを購入する必要があります。

LAN メディアの種類別に、それぞれ異なるネットワークインタフェースが定められています。たとえば、Ethernet をネットワークメディアとして使用したいのであれば、ネットワーク内の各ホストに Ethernet インタフェースをインストールしておく必要があります。Ethernet ケーブルを接続するために使用するボード上のコネクタを、Ethernet コネクタと言います。たとえば FDDI を使用しようと計画しているのであれば、予定している各ホストに FDDI ネットワークインタフェースが装備されている必要があります (その他のネットワークメディアの場合も同様です)。

本書では、ホストのデフォルトのネットワークインタフェースを一次ネットワークインタフェースと呼びます。

注 - ネットワークハードウェアのインストールについては、本書では取り扱いません。シリアルポートの構成方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』を、ネットワークメディアのインストールの手順については、ネットワークメディア付属しているマニュアルを参照してください。

ネットワークソフトウェアが情報を転送する仕組み

ネットワークソフトウェアの設定は複雑な作業です。そこで、まず設定しようとしているネットワークソフトウェアがどのようにして情報を転送するかを理解しておくことが重要です。

図 2–2 に、ネットワーク通信に関係のある基本的な要素を示します。

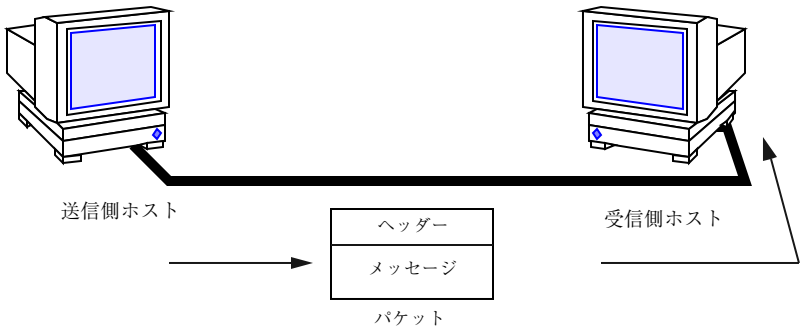


図 2–2　ネットワーク上での情報の転送

この図では、あるコンピュータがネットワークメディアを介して、同じメディアに接続している別のコンピュータにパケットを送信しています。

情報が転送される仕組み：パケット

ネットワークを介して転送する情報の基本単位をパケットと言います。パケットの構成は通常の手紙によく似ています。

どのパケットにもヘッダーがあり、これは手紙の封筒に当たります。ヘッダーには、受取先と送信元のアドレスに加えて、パケットがプロトコル群の各層を移送されるときにそのパケットをどのように扱うかを指示する情報が含まれています。

パケットのメッセージ部は手紙の本文に相当します。パケットに含めることのできるデータのバイト数には制限があり、これは使用しているネットワークメディアによって異なります。したがって、電子メールメッセージなどのような代表的な通信は、いくつかのパケットフラグメントに分割されることがあります。

情報を送受信する主体：ホスト

経験を積んだ Solaris ユーザーなら、もちろん「ホスト」という言葉はよくご存じのことでしょう。この言葉は、しばしば「コンピュータ」または「マシン」の同義語として使用されます。TCP/IP の視点から見れば、ネットワーク上に存在するエンティティは、ルーターとホストの 2 つだけです。

ルーターは、ネットワークから別のネットワークへとパケットを転送するマシンです。これを行うには、ルーターは少なくとも 2 つの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。ネットワークインタフェースが 1 つしかないマシン

は、パケットを転送できません。このようなマシンはホストとみなされます。ネットワーク管理者がネットワーク上に設定するマシンのほとんどはホストです。

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ターとしては機能しないマシンもあります。このようなマシンをマルチホームホストと呼びます。マルチホームホストは、持っているネットワークインタフェースを使用して複数のネットワークに直接的に接続されます。ただし、1 つのネットワークから別のネットワークへとパケットを転送することはしません。

あるホストが通信を開始したとき、それを送信側ホスト、送信、送信元、などと呼びます。たとえば、あるホストのユーザーが `rlogin` を入力するか、または他のユーザーに電子メールメッセージを送ると、そのホストは通信を開始します。通信の宛先となるホストを、受信側ホスト、受信側、受信先などと呼びます。たとえば、`rlogin` への引数として指定されたリモートホストは、そのログイン要求の受信先です。

各ホストは、ネットワーク上の他の対等ホストに自身を識別させるための次の 3 つの特性を備えています。

- ホスト名
- インターネットアドレス (本書では IP アドレスと呼んでいます)
- ハードウェアアドレス

ホスト名

ホスト名 は、ローカルマシンの名前と所属組織の名前を組み合わせたものです。多くの組織では、ユーザーが各自のマシンのホスト名を選定します。`sendmail` や `rlogin` などのプログラムは、ネットワーク上のリモートマシンを指定するときにホスト名を使用します。ホスト名については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』でより詳しく説明しています。

マシンのホスト名は、一次ネットワークインタフェースの名前にもなります。この概念は、ネットワークデータベースを設定したりルーターを構成したりするときに重要な意味を持ちます。

ネットワークを設定するときは、そのネットワークに関与するすべてのマシンのホスト名を入手する必要があります。ネットワークデータベースを設定するときに、必要となります。詳細は、89ページの「ネットワーク上のエンティティへの名前付け」を参照してください。

IP アドレス

「IP アドレス」は、TCP/IP ネットワーク上で各マシンが持っている 2 種類のアドレスの 1 つで、そのマシンをネットワーク上の他の対等ホストに識別させるためのものです。このアドレスには、特定のホストがネットワーク上のどこに位置しているかを、対等ホストに知らせる役割もあります。ネットワーク上のマシンに Solaris オペレーティング環境をインストールしたことがある場合は、インストール時に IP アドレスを指定したことを覚えていることでしょう。IP アドレス指定は TCP/IP の重要な要素の 1 つであり、これについては、87ページの「IPv4 アドレス指定スキーマの設計」で詳しく説明します。

ハードウェアアドレス

ネットワーク上の各ホストは一意のハードウェアアドレスを持っており、これもホストを他の対等ホストに識別させるために使用されます。ハードウェアアドレスは、製造元でマシンの CPU またはネットワークインタフェースに物理的に割り当ててあります。ハードウェアアドレスはどれも一意なものです。

本書では、ハードウェアアドレスを「Ethernet アドレス」という言葉で表しています。Ethernet は、Solaris オペレーティング環境のネットワーク上で最も一般的に使用されているネットワークメディアなので、本書では、Solaris ホストのハードウェアアドレスが Ethernet アドレスであるものと想定して、説明を進めます。FDDI など他のネットワークメディアを使用している場合は、そのメディアに付属しているマニュアルの中のハードウェアアドレス指定に関する部分を参照してください。

ローカルエリアネットワークの境界を越える - 広域ネットワーク

ネットワークをある程度の期間運用していくうちに、他の企業、専門研究機関、所属の LAN 上にない他の組織からの情報にアクセスする必要が出てくる場合があります。このような情報にアクセスするには、広域ネットワーク (WAN) を介して通信することが必要になります。WAN は地理的に広い範囲を対象とするもので、デジタル専用線または電話回線、X.25、ISDN サービスなどのネットワークメディアを使用します。

WAN の代表的な例としてインターネットがあります。インターネットは、TCP/IP が開発された最初の目的となっていた各 WAN の後に続いて開発された、世界規模の公共ネットワークです。WAN のその他の例としては企業ネットワークがありま

す。これは、ある 1 つの企業内の各事業所同士を、1 つの国の全域、や 1 つの大陸の全域にわたるようなネットワークによって結ぶものです。つまり、1 つの組織が独自の WAN を構築することが可能です。

ネットワーク管理者としては、ローカルネットワークのユーザーが WAN にアクセスできるようにする必要があります。TCP/IP と UNIX のコミュニティでは、最もよく使用されている公共ネットワークはインターネットです。インターネットに直接接続する方法については、本書では説明しません。これについては、役立つ書籍がコンピュータ関係の書店にたくさんそろっています。

セキュリティ

LAN を WAN に接続することには、セキュリティに関するある程度のリスクが伴います。自分のネットワークを無許可のアクセスから保護したり、データと資源へのアクセスを制御したりすることが必要になります。セキュリティの概要については、『*Solaris のシステム管理 (第 1 巻)*』に説明されています。詳細な説明は、William R. Cheswick および Steven M. Bellovin 共著の『*Firewalls and Internet Security*』(Addison Wesly, 1994) に記載されています。

米国の `majordomo@greatcircle.com` に、`subscribe firewalls` という文字列を入れたメールを送ることで、セキュリティについての情報を入手することもできます。ダイジェスト版の方をご希望の場合は、テキスト中に `firewalls_digest` という文字列を入れてください。

TCP セッションにおけるラージウィンドウのサポート

TCP セッションのラージウィンドウは、RFC1323 に記述されたサポートを提供します。このサポートは、一般的な上限値の 65,535 バイトより大きなウィンドウを使用することで、ATM や衛星ネットワークなどの広帯域または遅延ネットワークの性能を改善するように設計されています。

サポートするデータ量の増大が顕著なのは、65,535 バイトから約 1G バイトに上限値が拡張された TCP セッションです。

TCP セッションのラージウィンドウでは、多数の TCP 構成パラメータがサポートされます。これらのパラメータにより、システム管理者は拡張された送受信ウィンドウサイズと RFC1323 タイムスタンプオプションを使用できます。その際に、アプリケーションを修正する必要はありません。システム全体か特定のホストやネットワークに対して、パラメータを変更できます。このことが特に有効なのは、使用

するバッファーサイズの拡張機能を持たない ftp や rcp などの標準的なネットワークユーティリティを使用する場合です。

TCP ラージウィンドウのパラメータ

構成パラメータは、TCP デバイスを示す /dev/tcp に関連付けられ、ndd(1M) による検査と変更が可能です。通常、これらのパラメータは、システムのブート時に init(1M) が実行するシェルスクリプトの 1 つに設定されます (新規スクリプトの追加方法については、init.d(4) のマニュアルページを参照してください)。

使用可能なパラメータとそれぞれの意味は下記のとおりです。

tcp_xmit_hiwat
接続の送信バッファースペースにデフォルト値 (8K) を指定します。

tcp_recv_hiwat
接続の受信バッファースペース (受信データ用に割り当てられたバッファースペースの量。公示されている受信ウィンドウの最大サイズ) にデフォルト値 (8K) を指定します。

tcp_wscale_always
パラメータがゼロ以外であれば、リモートシステムへの接続時にウィンドウスケールオプションが必ず送信されます。パラメータがゼロであれば、64K より大きな受信ウィンドウをユーザーが要求した場合に (限って) 送信されます。デフォルトはゼロです。

このパラメータの値にかかわらず、ウィンドウスケールオプションが必ず接続肯定応答に含まれるのは、接続システムがそのオプションを使用した場合です。

tcp_tstamp_always
パラメータがゼロ以外であれば、リモートシステムへの接続時にタイムスタンプオプションが必ず送信されます。デフォルトはゼロです。

このパラメータの値にかかわらず、タイムスタンプオプションが必ず接続肯定応答 (および以降の全パケット) に含まれるのは、接続システムがそのオプションを使用した場合です。

tcp_tstamp_if_wscale

パラメータがゼロ以外であれば、リモートシステムへの接続時にタイムスタンプオプションが送信されるのは、64K より大きな受信ウィンドウをユーザーが要求した場合 (つまり、ゼロ以外のスケールを指定したウィンドウスケールオプションを使用している場合) です。デフォルトはゼロです。

tcp_max_buf

SO_SNDBUF または SO_RCVBUF オプション付きでユーザーが指定できるバッファーサイズの最大値を指定します。この値より大きなバッファーの使用を試みると、EINVAL を返して失敗します。デフォルトは 256K です。アプリケーションに必要な最大バッファーサイズよりもずっと大きな値をパラメータに指定するのはお勧めできません。障害や悪影響の原因となっているアプリケーションが、カーネルメモリを不当に大きく消費しかねないからです。

tcp_host_param

このパラメータは、IP アドレス、ネットワーク、サブネットワーク、および指定されたホストとの接続に使用される特定の TCP パラメータのデフォルト値をテーブルにしたものです。テーブルを表示するには、以下のように ndd コマンドを使用します。

```
example# ndd /dev/tcp tcp_host_param
Hash HSP     Address         Subnet Mask     Send       Receive    TStamp
027 fc31eea4 129.154.000.000 255.255.255.000 0000008192 0000008192      0
131 fc308244 129.154.152.000 000.000.000.000 0000032000 0000032000      0
133 fc30bd64 129.154.152.006 000.000.000.000 0000128000 0000128000      1
```

テーブルの各要素は、ホスト、ネットワーク (サブネットマスクのオプション付き)、サブネットのどれかに加えて、デフォルトの送信バッファースペースと受信バッファースペース、タイムスタンプを使用するかどうかを示すフラグを表示します。

テーブル内で指定されているデフォルト値は、アクティブな接続とパッシブな接続 (connect() と listen()) の両方に使用できます。ホストアドレス全体、サブネット、ネットワークの順で、検出された最適な一致が使用されます。サブネット

の認識が有効に動作するためには、サブネットのネットワークにサブネットマスクを指定するエントリがなければなりません。

上のテーブルの例が示す内容は、以下のとおりです。

- ホスト 129.154.152.6 との接続では、128,000 バイトの送受信バッファーサイズと、タイムスタンプを使用します。
- サブネット 129.154.152 にある他のホストと接続するための送受信バッファーサイズは、32,000 バイトです。
- ネットワーク 129.154 にある他のホストと接続するための送受信バッファーサイズは、8,192 バイトです。

テーブルの要素を追加または削除するには、以下のように ndd を使用します。

```
ndd -set /dev/tcp tcp_host_param '<command>'
```

<command> には次のいずれかを指定します。

```
<ipaddr> [ mask <ipmask>] [ sendspace <integer> ]
    [ recvspace <integer> ] [ timestamp { 0 | 1 } ]
```

または

```
<ipaddr> delete
```

たとえば、上のテーブルを作成するには、次のように指定します。

```
# ndd -set /dev/tcp tcp_host_param '129.154.0.0 mask 255.255.255.0
        sendspace 8192 recvspace 8192'
# ndd -set /dev/tcp tcp_host_param '129.154.152.0 sendspace 32000
        recvspace 32000'
# ndd -set /dev/tcp tcp_host_param '129.154.152.6 sendspace 128000
        recvspace 128000 timestamp 1'
```

注 - 上記の例では、コマンドが 2 行に分割されていますが、コマンドはすべて、1 行に入力してください。

削除するには、次のように指定します。

```
# ndd -set /dev/tcp tcp_host_param '129.154.152.6 delete'
# ndd -set /dev/tcp tcp_host_param '129.154.152.0 delete'
# ndd -set /dev/tcp tcp_host_param '129.154.0.0 delete'
```

ネットワークとサブネットを指定するには、ホストビットをゼロにしておきます。エントリ追加用の構文は、既存エントリの修正にも使用できます。

tcp_host_param テーブルからの送受信スペースの値が使用されるのは、それらの値がユーザーが設定した (または、tcp_xmit_hiwat と tcp_recv_hiwat から取得した) 値よりも大きい場合に限られます。したがって、スループット向上のためにユーザーが大きな値を指定することが可能で、それらの値が誤って縮小されることはありません。

tcp_host_param テーブルのタイムスタンプ値が 1 の場合、接続を開始したときに選択したホストにタイムスタンプオプションが送信されます。ただし、値が 0 の場合でも、tcp_tstamp_always と tcp_tstamp_if_wscale オプションの設定により、タイムスタンプオプションが送信されることがあります。

TCP 選択確認応答のサポート

TCP 選択確認応答 (TCP SACK) は、RFC 2018 に記述されているサポートを提供し、特に衛星リンクや大陸間リンク上で TCP ラージウィンドウ (RFC 1323) を使用するアプリケーションにおいて、混雑や複数パケットの脱落に関連した問題を解決します。

構成パラメータは、TCP デバイス /dev/tcp に関連付けられており、ndd(1M) を使用してその検査や変更を行うことができます。通常、このパラメータは、システムの起動時に init(1M) によって実行されるシェルスクリプトのいずれかで設定されます (新しいスクリプトの追加方法については、init.d(4) のマニュアルページを参照してください)。

使用可能なパラメータとその意味を以下に示します。

tcp_sack_permitted

SACK を許可するかどうかを示します。デフォルトは 1 です。使用可能なオプションを以下に示します。

0 TCP は SACK 情報の受信や送信を行いません。

1 TCP は SACK_PERMITTED オプションによる接続は開始しません。受信した要求に SACK_PERMITTED が含まれている場合は、TCP は SACK_PERMITTED オプションを使用して応答します。

2 TCP は SACK_PERMITTED オプションを使用して接続の開始と許可を行います。

詳細は、tcp(7P) のマニュアルページを参照してください。

第 3 章

IP アドレス管理トピック

第 4 章	TCP/IP の概要
第 5 章	TCP/IP ネットワークの計画
第 6 章	TCP/IP ネットワークの設定手順と障害追跡手順
第 7 章	TCP/IP の参照情報
第 8 章	DHCP の概要
第 9 章	DHCP の使用計画
第 10 章	DHCP の設定手順
第 11 章	DHCP の管理手順
第 12 章	DHCP の障害追跡
第 13 章	DHCP の背景情報

第 14 章	IPv6 の概要
第 15 章	IPv6 移行計画とそのメカニズム
第 16 章	Solaris IPv6 の実装
第 17 章	IPv6 関連作業の手順
第 18 章	IPsec の概要
第 19 章	IPsec の設定手順

第 4 章

TCP/IP の概要

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 (ネットワークの基本概念の紹介については、第 2 章を参照してください)。TCP/IP の経験のあるネットワーク管理者の場合は、この章では以下の項目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 関係の書籍は、どれもここに示す概念を理解していることを前提として書かれているので、この情報は重要です。

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ニックネーム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表すものとします。

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 5 章で詳しく説明します (DNS についての詳細は、『*Solaris* ネーミングの管理』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が層の形に構造化されているとみなされるのは、国際標準化機構 (ISO) が設計した開放型相互接続 (OSI) 参照モデルの結果です。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ち、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されるものと規定しています。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 参照モデルのプロトコル層は、通常は表 4–1 に示すように、上 (層 7) から下 (層 1) へ並べて表します。

表 4–1　開放型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し、受信されたデータと送信されたデータが同じになるようにする
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 モデルにより定義されている動作は概念的なものであり、特定のネットワークプロトコル群に特有のものではありません。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 モデルの層のいくつかを使用し、その他を合併しています。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

TCP/IP は、いくつかの OSI 層を合併して 1 つの層にしていたり、またまったく使用しない層があったりするため、このモデルに直接対応しているとは言えません。表 4–2 は、Solaris 実装の TCP/IP の層を示しています。最上位の層 (アプリケーション) から最下位の層 (物理ネットワーク) まで並べてあります。

表 4–2 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、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、それぞれ独自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 5 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、81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) が実装されていました。しかし、インターネットの急速な成長によって、アドレス空間の拡張など、機能強化された新しいインターネットプロトコルを開発する必要が生じました。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の検出とその報告を担当するプロトコルです。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 (パケットの到着が速すぎて処理が間に合わない場合)
- 接続障害 (宛先ホストに到達できない場合)
- リダイレクト (送信側ホストに別のルーターを使用するよう指示)

122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。受信側ホストと送信側ホストとの間で接続が達成されているかどうかを検査する手段は提供しません。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にはさまざまのものがあり、そのうちのいくつかは、すでに使用しているでしょう。以下に、この種のプロトコルの例をいくつか挙げます。

- 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。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各種の匿名 FTP サーバーから、さまざまの豊富な資料や情報を入手できます。これらのサーバーは大学その他の研究機関により設定されたもので、ある種のソフトウェア、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 *(*第 *1* 巻*)*』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ものも含むすべての 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関する説明があります。

- 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.telnet 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。これは、ユーザーが、コピーしたいファイルのフルネームを知っていなければならないことを意味します。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、指定したリモートホストで実行したいコマンドを、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 実装の TCP/IP では、NIS+ と DNS の 2 つのネームサービスが使用できます。

- 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。詳細は、『*Solaris* ネーミングの管理』を参照してください。
- ドメインネームシステム - ドメインネームシステム (DNS) は、ホスト名から IP アドレスへの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* ネーミングの管理』を参照してください。`in.named（1M）` のマニュアルページも参照してください。

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 サービスについての詳細は、第 29 章で説明しています。

ネットワーク管理

SNMP (ネットワーク管理用プロトコルの一種。Simple Network Management Protocol) を使用すると、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し、さらに、その他の複雑な統計情報をグラフィカルユーザーインタフェースを持つソフトウェアから得ることができます。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、160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、ネットワークメディアを通り、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。

ユーザーのコマンドがプロトコルスタックを通過していくとき、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 4–1 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、ネットワーク上を転送される情報の基本単位で、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ています。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 4–1 に示すように層によって異なります。

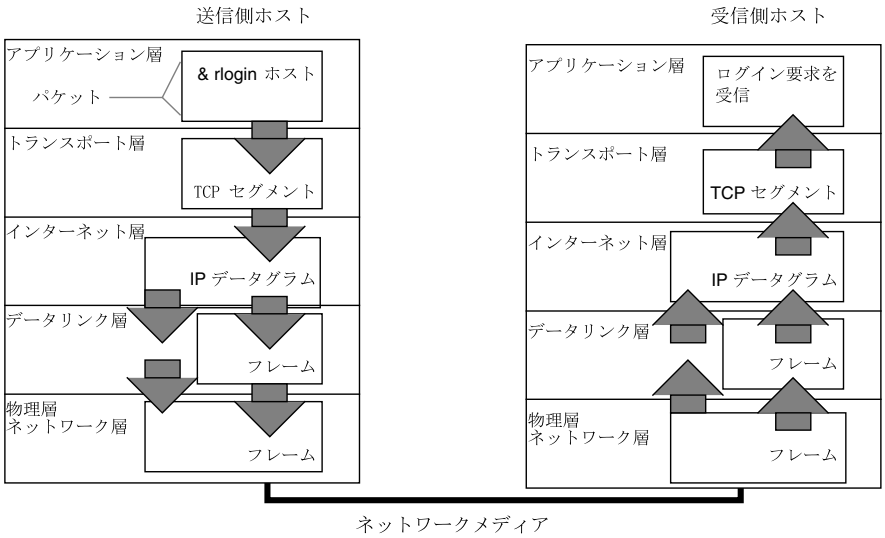


図 4–1　TCP/IP スタックを通過するパケット

この節では、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してから、それを受信側ホストの該当のアプリケーションが受け取るまでの、パケットのライフサイクルを要約して示します。

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。そのコマンドまたはメッセージに関連付けられている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で取り扱えるように、パケットの形式を設定します。

図 4–1 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、このプロトコルが、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 4–1 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、受信側ホストで実行されている対等 TCP プロトコルに SYN と呼ばれるセグメントを送ります。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け取ったメッセージを「UDP パケット」に形式化します。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、検査合計が含まれています。

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 4–1 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 4–1 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ります。
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 と取り除きます。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て、それ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ります。
5. アプリケーション層はメッセージを受け取り、送信側ホストから要求された操作を行います。

TCP/IP 内部トレースのサポート

TCP/IP は、RTS 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。RTS パケットが送信または受信されたときに、その接続上で直前に送信または受信された最大 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、膨大な量の情報が出版されています。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。

市販の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。中でも特にお勧めしたいのは次の書籍です。

- 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、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

1969 年以来、インターネットプロトコル群に携わる開発者たちは、それぞれのプロトコルと関連の主題を、RFC (コメント要求 = Requests for Comments) と呼ばれる文書の形で記述してきました。多くの RFC は特定の TCP/IP プロトコルの仕様であり、そのプロトコルを実装するソフトウェアが従う必要のある規格を記述しています。ほかに、インターネット、そのトポロジ、その運営組織について記述した RFC もあります。さらに、DNS などのような TCP/IP アプリケーションの管理方法を説明する RFC もあります。

RFC がパブリックドメインに公開されるには、IAB (Internet Architecture Board) より承認されることが必要です。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いますが、すべてがそうであるとは限りません。

最近になって、RFC のサブセットとして FYI (For Your Information) 文書が発行されるようになりました。FYI には、インターネット規格を取り扱うような情報は含ま

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。FYI には、たとえば、TCP/IP の入門書や資料の目録、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。

InterNic Directory and Database Service には、RFC の蓄積が維持されています。インターネットに接続している場合は、次のようにしてオンラインで RFC を検索できます。

- `ftp` を用いる場合は、InterNic ディレクトリおよびデータベースサーバー `ds.internic.net` に要求を送ります。要求の形式は次のとおりです。

  `rfc/rfc.`*rfcnum*`.txt` または `rfc/rfc.`*rfcnum*`.ps`

  rfcnum は、入手したい RFC の番号です。たとえば、RFC 1540 を PostScript™ 形式で検索したい場合、`rfc/rfc.1540.ps` という要求を出します。

- 電子メールを用いる場合は、米国の `mailserv@ds.internic.net` に電子メールを送ります。これは自動サーバーなので、要求メッセージのボディーは次の形式になっている必要があります。

  `document-by-name rfc`*rfcnum* または `document-by-name rfc`*rfcnum*`.ps`

- World Wide Web ブラウザを用いる場合は、URL は `http://ds.internic.net/ds/dspg1intdoc.html` を指定します。ホームページは `http://ds.internic.net` です。

RFC のオンラインインデックスを必要とする場合は、`document-by-name rfc-index` という要求を含んだメッセージを、米国の `ds.internic.net` に電子メールで送ってください。

注 - インターネットは急速に成長しているので、上記に示したアドレスは、本書をお読みになる時点には変更されている場合があります。

第 5 章

TCP/IP ネットワークの計画

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。以下の項目について説明します。

ネットワークの設計

ネットワークのライフサイクルの最初の段階は、ネットワークの設計です。この段階では、まず、組織のニーズを満たす最適のネットワークの種類を決定することから始めます。計画段階で行う決定には、たとえば次のように、ネットワークハードウェアに関連したものがいくつかあります。

- 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、すなわちネットワークハードウェアの物理的なレイアウトと接続
- 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については、本書では説明しません。詳細は、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関連要素

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. ネットワークを形成するすべてのマシンの IP アドレスとホスト名を含むリストを作成します。これは、ネットワークデータベースの構築の際に利用できます。
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、上記の要素を念頭に置きながら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 アドレス指定スキーマの設定

サポートを予定しているマシンの数によって、このサイトでのネットワーク設定について、現段階で行ういくつかの決定事項が影響を受けます。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

ネットワーク番号を入手し、IP アドレス指定スキーマを確立することは、ネットワーク管理の計画段階において、最も重要な作業の 1 つです。

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ることのないように注意してください。IPv6 への移行を計画している場合は、第 15 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 5–1 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、*xxx* が 1 バイトを表しています。「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ます。

表 5–1 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	1 ～ 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、常に 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。

表 5–2 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、ホストへの割り当てが可能な各バイトにどの範囲の値を指定できるかを示しています。

表 5–2 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

57ページの「ネットワークインタフェース」で説明しているように、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、それにも一意な IP アドレスが必要です。114ページの「ルーターの構成」で説明したように、第 2 のネットワークインタフェースを追加すると、マシンの機能がホストからルーターに変わります。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 アドレスを指定してしまったら、次に行う作業は、ホストに名前を割り当て、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ことです。これらの名前は、最初にネットワークを設定するときに使用するほか、後日ルーターや PPP を使用してネットワークを拡張するときにも使用します。

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、人間にとっては、マシンに意味のある名前が付いている方が、識別しやすくて便利です。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、ネットワークとそのマシン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。名前が付けられたこれらのエンティティを総称してドメインと呼びます。

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、ネットワーク内の “fred” というマシンが複数の IP アドレスを持っていてもかまいませんが、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。

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。

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、サーバーマシン、クライアントマシン、スタンドアロンマシンのホスト名と 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` データベースと `ipnodes` データベースという 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。これらのデータベースは、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により読み取られます。

ネットワークデータベースの設定は、ネットワーク構成の重要な部分です。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、148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とそれぞれの設定方法については、『*Solaris* ネーミングの設定と構成』 に詳しい説明があります。「名前空間」と「管理ドメイン」の概念に関する説明も出ています。ネームサービスを NIS から NIS+ に変更する場合は、『*NIS+* への移行』を参照してください。これらのマニュアルは、ネットワークでこれらのネームサービスのどれを使用するかを決める際の参考として役立ちます。

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。特に、DNS の使用を予定している場合は、この処置が重要です。

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。

最上位ドメインの例としては次のようなものがあります。

- `.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するか、または既存のネットワークをサブネットに分割します。ネットワーク管理の作業を分化するかどうかを決める点には、次のものがあります。

- ネットワークの規模

  数百台のホストから成る単一のネットワークにおいて、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ものである場合は、1つの管理部門だけで対処できるでしょう。これに対して、マシン数がもっと少ない場合でも、ネットワークが多数のサブネットに分割されていて、しかも地理的に広い範囲に散在しているとすれば、複数の管理部門を設立する方が効率的になります。

- 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。また、このネットワークのマシンがいくつかのサブネットワークに分割され、各サブネットワークがそれぞれ大幅にニーズの異な

るユーザーのグループをサポートしているとします。このような場合は、各サブネットごとに管理部門を設けるとよいでしょう。

管理作業の分化についての詳細は、『*Solaris* ネーミングの管理』で説明しています。

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、管理ドメインの使用を予定している場合は、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、インターネットのユーザーに以下の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。登録については、次の 2 つ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する予定がなくても、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。

サブネット番号は InterNIC が割り当てるものではありません。この番号は、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、145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、米国東部標準時で午前 7 時から午後 7 時までです。

- 電子メール

  次の米国の宛先にネットワーク登録に関する電子メールを送ります。

  `Hostmaster@rs.internic.net`

- Gopher と WAIS インタフェースを使用した匿名 FTP または Telnet 照会

  `rs.internic.net` に接続します (本書では、匿名 FTP と Telnet については説明しませんが、コンピュータ関係の書店にこれらの事項に関する書籍がそろっています)。

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がありますが、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。ルーターを使用するかどうかは、ネットワークの物理的なトポロジによって異なります。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。

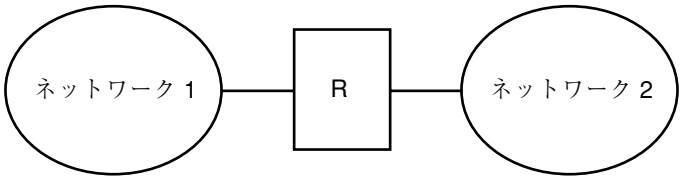
ネットワークトポロジ

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、114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。

複数のネットワークをルーターによって接続することで、より大きなインターネットワークを作ることができます。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 5–1 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、ネットワーク 1 とネットワーク 2 がルーター R で連結されて、より大きなインターネットワークが作られています。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 から成る 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 5–1　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、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、パケットをどこに送るかについての各ルーターでの決定の回数は増加します。

複雑さの度合の増加を示す例として、のような場合が考えられます。この例では、ネットワーク 1 とネットワーク 3 が、ルーター R3 により直接接続されています。このような冗長な方法を使用する目的は、信頼性にあります。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。すべてが同じネットワークプロトコルに従っていれば、ネットワークをいくつでも相互接続して、互いに通信させることができます。

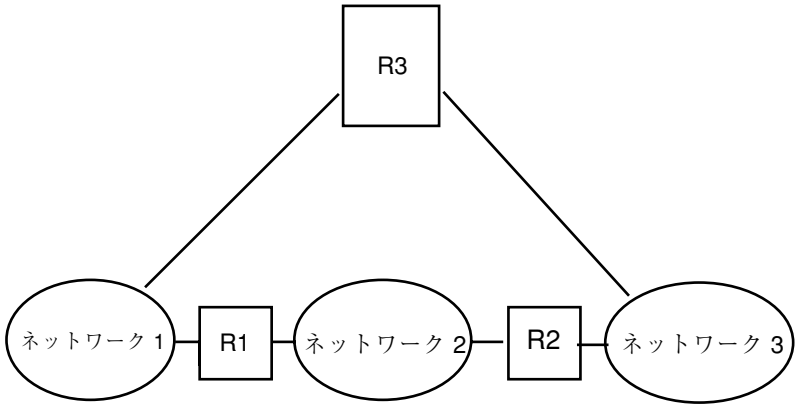


図 5–2　ネットワーク間のパスの追加

ルーターがどのようにパケットを転送するか

ネットワーク上でのルーティングに関する決定は、パケットヘッダーに含まれている受信側 IP アドレスのネットワーク部に基づいて行われます。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、パケットを受け取ると、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、 114ページの「ルーターの構成」を参照してください。

図 5–3 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ています。

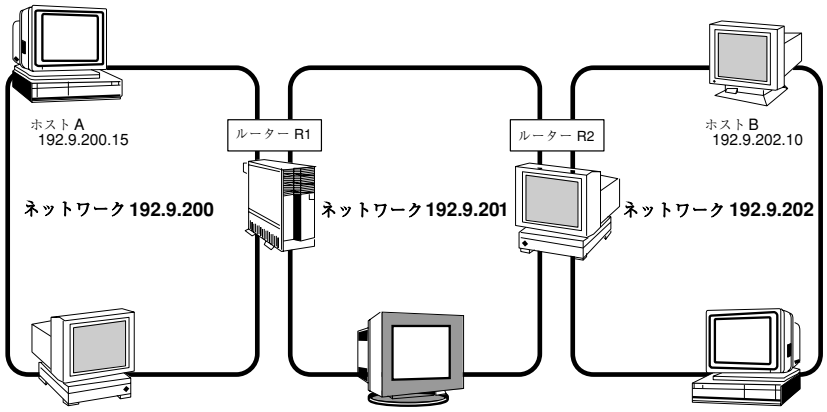


図 5–3　相互接続された 3 つのネットワーク

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 を接続しています。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とすると、その操作は次の手順で行われます。

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「次のリレー」ルーターとして R2 を選択し、パケットを R2 に送ります。
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け取ります。

第 6 章

TCP/IP の管理

ネットワーク管理の段階では、ネットワークを設定します。ここで行う作業は、ネットワークの物理部分を形成するハードウェアの組み立てと、TCP/IP の構成です。この章では TCP/IP の構成方法と、ネットワークを構成した後で TCP/IP の障害追跡を行う方法について説明します。

- 123ページの「ホストでパケットが失われていないか確認する方法」
- 124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」
- 125ページの「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」
- 126ページの「プロトコル別の統計情報の表示方法」
- 128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」
- 128ページの「ルーティングテーブルの状態の表示方法」
- 129ページの「ネットワークの問題を記録する方法」
- 131ページの「システムから全パケットを確認する方法」
- 132ページの「snoop の結果をファイルに取り込む方法」
- 133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」
- 134ページの「traceroute ユーティリティの実行方法」

TCP/IP を構成する前に行う作業マップ

TCP/IP ソフトウェアの構成を行う前に、表 6–1 に示す作業を完了しておく必要があります。

表 6–1 TCP/IP を構成する前に行う作業マップ

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	94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	87ページの「IPv4 アドレス指定スキーマの設計」
設計した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、ハードウェアが動作することを確認する	ハードウェアのマニュアルと 94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	94ページの「ルーターの追加」と 114ページの「ルーターの構成」

表 6–1 TCP/IP を構成する前に行う作業マップ 続く

説明	参照先
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	87ページの「IPv4 アドレス指定スキーマの設計」と 334ページの「IPv6 アドレス指定」
ネットワークに含まれるすべてのマシンに、IP 番号とホスト名を割り当てる	87ページの「IPv4 アドレス指定スキーマの設計」と 334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* ネーミングの管理』
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* ネーミングの管理』
予定している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 8* のインストール (上級編)』

ホスト構成モードの決定

ネットワーク管理者が行う主要な作業の 1 つに、ホストとルーター (必要な場合) で実行できるように TCP/IP を構成する作業があります。これらのマシンは、2 つの情報源から構成情報を入手するように設定できます。それは、ローカルマシン上のファイルと、ネットワーク内の他のマシンにあるファイルです。構成情報には次のものがあります。

- 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の状態を、ローカルファイルモードで稼動していると言います。TCP/IP 構成情報をリモートマシンから入手するマシンの状態を、ネットワーククライアントモードで稼動していると言います。

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、137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。主な必要条件は次のとおりです。

- 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、既知の 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実行されているホストが、各自の IP アドレスと TCP/IP 構成ファイルをサーバーから入手できるようになります。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。
- 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。

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の状態を、ネットワーククライアントモードで「稼動中」であると言います。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確実に同じ構成標準に従ったものとなります。

完全なスタンドアロンシステムからデータレスマシンに至るまで、すべての種類のコンピュータについて、ネットワーククライアントマシンを構成できます。ルーターとサーバーもネットワーククライアントモードで構成できますが、これらのマシンではローカルファイルモードの方がよい選択です。ルーターとサーバーは、できる限り自給自足型にしておかねばなりません。

混合構成

システムは高い柔軟性を備えているため、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ような、どちらか一方に限定する必要はありません。そのよい例がルーターとサーバーで、これらは常に

ローカルモードで構成するのが最適です。ホストについては、必要に応じて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

図 6–1 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おり、それぞれの名前は、ネットワーク 192.9.200 用が timbuktu で、ネットワーク 192.9.201 用が timbuktu-201 です。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。

この章の中のほとんどの例では、図 6–1 に示すネットワークにもとづいて説明しています。

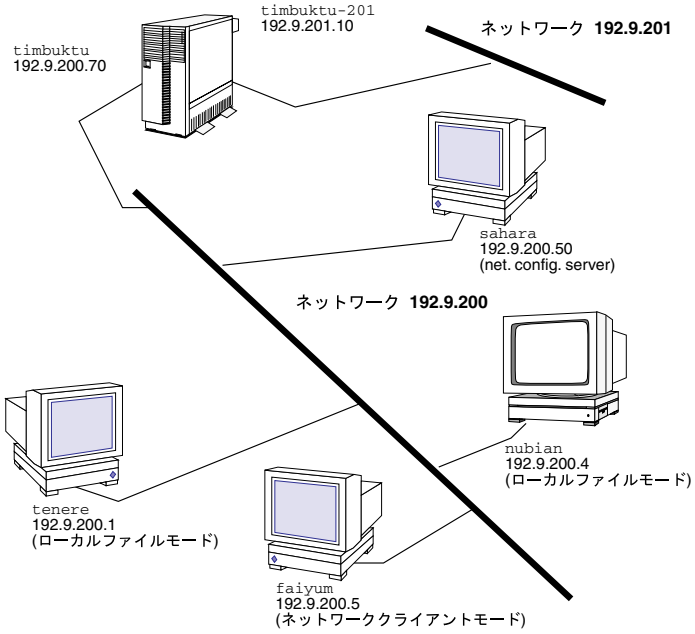


図 6–1　サンプルネットワーク内のホスト

ネットワークにサブネットを追加するための作業マップ

サブネットを使用していないネットワークをサブネット化するためには、表 6–2 に示す作業を行ないます。

表 6–2　ネットワークにサブネットを追加するための作業マップ

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	94ページの「ルーターの追加」、145ページの「サブネット化とは」、および163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	87ページの「IP アドレス指定スキーマの設定」、および 161ページの「IPv4 アドレスの構成部分」
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	144ページの「netmasks データベース」、および 145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	140ページの「hosts データベース」、および 144ページの「ipnodes データベース」
5. すべてのマシンを再起動する	

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。構成情報がどのようにマシンのカーネルに対して使用可能にされるかは、構成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)、それともネットワーク構成サーバーから構成ファイルを入手するか (ネットワーククライアントモード) によって異なります。

ネットワーク構成時に指定するパラメータには、次のものがあります。

- 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。これを指定するのは、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か、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だけである。(これらのプロトコルについての詳細は、160ページの「ルーティングプロトコル」を参照してください)
- 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* ネーミングの管理』を参照してください。

ネットワークを構成するための作業マップ

表 6–3 ネットワークを構成するための作業マップ

作業	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	108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` daemon をオンにし、`intetd.conf`、`hosts`、`ethers`、および `bootparams` ファイルを編集する	110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	111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	112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」

表 6–3　ネットワークを構成するための作業マップ　続く

▼ 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)。138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、368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** アドレスか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 アドレスか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、140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、383ページの「/etc/inet/ipnodes ファイル」を参照してください。

---

6. 完全指定のドメイン名を /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、140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、140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。111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」中の、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
```

   これで、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。

   154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. サーバーをリブートします。

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 8* のインストール (上級編)』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。106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、368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。

   (142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。また、IPv6 を使用している場合は、383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ます。

   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追跡し続けます。

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、in.routed デーモンを実行してルーターを追跡します。

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。ネームサービスの順序を nsswitch.conf の中で指定したように、TCP/IP のサービスは、/etc/inetd.conf ファイルの中で inetd –t フラグを使用して構成できます。

たとえば、inetd を使用して、着信したすべての TCP 接続 (リモートログインと telnet) の IP アドレスをログに記録できます。ログ記録を作成するには、次の手順を実行します。

▼ 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. inetd デーモンを強制終了します。
3. 次のコマンドを入力して、ログ記録をオンにします。

```
# /usr/sbin/inetd -t -s
```

スイッチ –t を指定することで、inetd は TCP 接続トレースを開始します。

inetd(1M) と inetd.conf(4) のマニュアルページを参照してください。

ネームサービスについての詳細は、『*Solaris* ネーミングの管理』と『*Solaris* ネーミングの設定と構成』を参照してください。

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、57ページの「ネットワークインタフェース」 で説明したように、マシンが少なくとも 2 つのネットワークインタフェースを持っていなければならないということです。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、絶えずネットワーク上でのルーターの状態を追跡し、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、108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 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ための作業マップ

表 6–4 ルーターを構成するための作業マップ

作業	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	115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	116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	116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	117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. インストールされているネットワークインタフェースごとに、/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。詳細については、138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、368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からです。IPv6 を使用している場合は、383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。

140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。
   このファイルを空にしておくと、これによりネットワーククライアントに強制的に動的ルーティングプロトコルを選択させることができます。

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. **ホスト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。**

これは、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、467ページの「ルーティングに関する考慮事項」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。その一方で、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、それ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
```

これで、/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 も実行せず、また、ifconfig により "up" として構成されているインタフェースでは、いっさい IP の転送を行いません。これは、/etc/gateway 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) をホストに提供する必要がない場合は、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/jc2.d/S69inet` 起動スクリプトを編集します。

```
/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 ルーター発見をオフにする

ルーターの信頼性の都合により、ホストに RDISC 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) で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 ルーター発見をオフにするための作業マップ

表 6–5　ICMP ルーター発見をオフにするための作業マップ

作業	説明	参照先
ホスト上で ICMP ルーター発見をオフにする	ホストの in.rdisc ファイルの名前を変更する	144ページの「netmasks データベース」
ルーター上で ICMP ルーター発見をオフにする	ルーターの in.rdisc ファイルの名前を変更する	145ページの「サブネット化とは」

▼ ホスト上で ICMP ルーター発見をオフにする方法

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. ホストを再起動します。

▼ ルーター上で ICMP ルーター発見をオフにする方法

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. ルーターを再起動します。

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徴候は、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、またはネットワークインタフェー

スに問題があることが考えられます。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。

1. netstat コマンドを使用してネットワーク情報を表示します。
2. 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。
3. 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。
4. telnet によりローカルホストに接続してみます。
5. 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。そのためには、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]
```

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。詳細は、74ページの「ICMP プロトコル」を参照してください)。

ping コマンドで行う作業マップ

表 6–6 ping コマンドで行う作業マップ

作業	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	148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	149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
```

これは、*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。

- コマンド行で次のように入力します。

```
% 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、124ページの「ifconfig コマンド」と 126ページの「netstat コマンド」を参照してください。

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。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 コマンドで行う作業マップ

表 6–7 ifconfig コマンドで行う作業マップ

作業	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	124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	151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
```

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 コマンドで行う作業マップ

表 6–8 netstat コマンドで行う作業マップ

作業	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	126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	128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	128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、下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、各ネットワークについてマシンが送信し受信したとみなしているパケットの数が分かります。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。これは、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。この原因としては、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考えられます。

逆に、入力パケットカウントが長時間にわたり変化しないとすれば、それは、マシンがパケットをまったく見ていないことを意味します。この原因としては、上記の場合と違って、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、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。詳細は、`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 root(#) 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。これを参照するには、Web ブラウザで `http://ds.internic.net/rfc/rfc1761.txt` にアクセスしてください。

`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ための作業マップ

表 6–9 パケットの内容を表示するための作業マップ

作業	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	131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	132ページの「snoop の結果をファイルに取り込む方法」
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	133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使用します。これを参照するには、Web ブラウザで http://ds.internic.net/rfc/rfc1761.txt にアクセスします。

▼ 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。正確な状態を確認するには、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 の確認には、Web を検索することをお奨めします。

▼ 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に接続されたハブのいずれかを外します。

   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。これを参照するには、Web ブラウザで `http://ds.internic.net/rfc/rfc1761.txt` にアクセスします。

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 応答と、宛先ホストからの応答 PORT_UNREACHABLE (または、ECHO_REPLY) の受信を試みます。traceroute ユーティリティは、ttl を 1 にして探査の送信を開始し、探査が目的のホストに到達するか、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 7 章

TCP/IP ネットワークリファレンス

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/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、148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、383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。108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 であるとす

ると、/etc/hostname.*interface* ファイルの名前は /etc/hostname.smc0 となり、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 6–1 に示したマシン timbuktu について考えてみましょう。このマシンは 2 つのネットワークインタフェースを持っており、ルーターとして動作します。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 一次ネットワークインタフェース用として、/etc/hostname.le0 というファイルを作成し、そのファイルにホスト名 timbuktu を入れます。

第 2 のネットワークインタフェースは le1 で、これはネットワーク 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。したがって、ネットワーク管理者が、このインタフェース用に /etc/hostname.le1 ファイルを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。少なくとも 1 つの /etc/hostname. または /etc/hostname6. がローカルマシンになければなりません。Solaris のインストールプログラムは自動的にこれらのファイルを作成します。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、368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 6–1 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* ネーミングの管理』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 6–1 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 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、このマシンが認識してい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 ファイルを作成します。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 (ループバックアドレス、IPv4 アドレス、ホスト名) が入っています。

たとえば、図 6–1 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 7–1　マシン ahaggar 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 7–1 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使用する予約済みネットワークインタフェースで、これによりプロセス間通信が可能になり、ローカルマシンは自分自身にパケットを送ることができます。124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、114ページの「ルーターの構成」を参照してください)。

例 7–2 は、図 6–1 に示したマシン timbuktu 用の /etc/inet/hosts ファイルです。

例 7–2　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 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。これらのサービスについては、『*Solaris* ネーミングの管理』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、/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

次のコード例に、ローカルファイルモードで実行されるマシンである tenere の /etc/inet/hosts ファイルを示しています。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、192.9.200 ネットワークを 192.9.201 ネットワークに接続するためのネットワークインタフェースの IPv4 アドレスと、インタフェース名 timbuktu-201 も含まれています。

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。

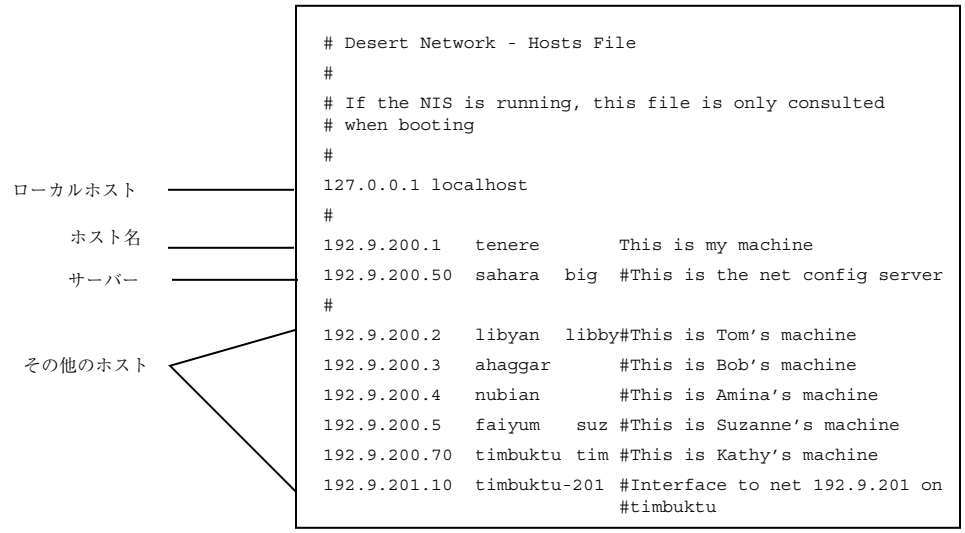


図 7–1　ローカルファイルモードで実行されるマシン用の /etc/inet/hosts ファイル

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 などのネームサービスを使用している場合、ipnodes データベースは、ホスト情報用に指定されたデータベース内に保持されます。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、383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。

87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。

ネットマスクを説明するには、2 進数表現の視点から見るのが最も簡単です。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したいとすれば、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 または NIS+ 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` ファイルが `/etc/inet/netmasks` へのシンボリックリンクとなっています。

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 7–3 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
```

(続く)

```
#
# 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。

例:

```
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。

ネットワークデータベースではありませんが、nsswitch.conf ファイルも、関連の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がとる形式は、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 7–2 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。

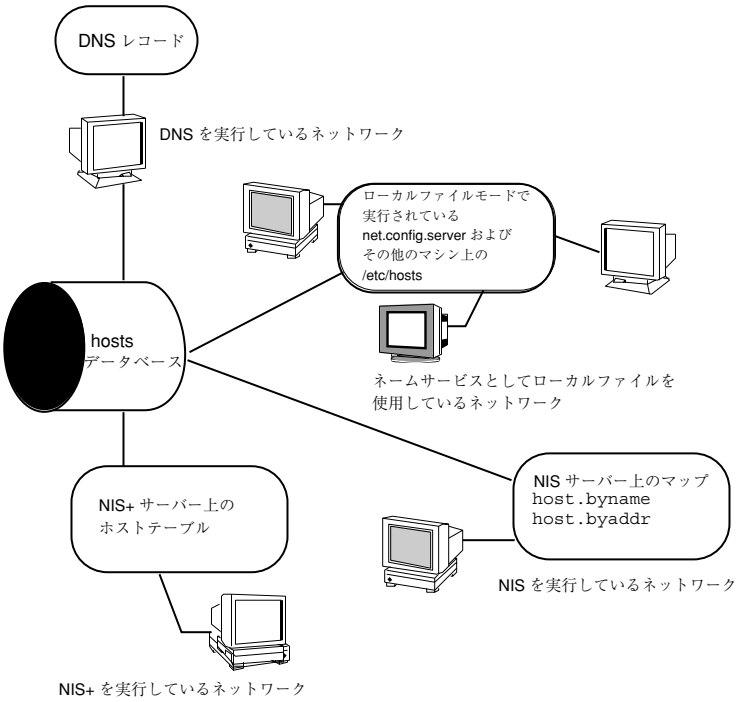


図 7–2　ネームサービスが使用する hosts データベースの形式

表 7–1 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 7–1　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

表 7–1 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　続く

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
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、140ページの「hosts データベース」を、ipnodes データベースについては、383ページの「/etc/inet/ipnodes ファイル」を、netmasks データベースについては、144ページの「netmasks データベース」を、NIS、DNS、NIS+ 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* ネーミングの管理』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 7–4 のようになります。

例 7–4　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;
# 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。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の *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 のように、複数のネームサービスを指定することもできます。

例 7–4 にサーチ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 7–5 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
```

(続く)

```
sendmailvars:  files
```

ネームサービススイッチについての詳細は、『*Solaris* ネーミングの管理』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と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、111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。このデータベースには、すべてのクライアントに一致するワイルドカードエントリを含めることができます。次に例を示します。

例 **7–6**　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 ネットワークの作成が必要になるのは、RARP デーモンを実行する場合、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できる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 7–7　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。このデータベースを更新する必要があるのは、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した場合だけです。

/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 ファイルの管理は大変重要です。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 7–8　/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。このファイルについて管理作業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。詳しい構文は、protocols(4) のマニュアルページに記載されています。例 7–9 に、/etc/inet/protocols ファイルの例を示します。

例 7–9　/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。このデータベースについては、通常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 7–10 /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
```

ブート処理の概要

以下の情報は参考用です。ネットワークのブート処理の概要を示しています。構成時にどのようなことが起こるかを全体的にとらえるのに役立ちます。

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. /etc/rcS.d/S30rootusr.sh 起動スクリプトが、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。これは、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からです。
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作業を実行します。これらの作業には、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 の作業が完了すると、/etc/rc2.d/S71rpc が実行されます。このスクリプトは、NIS、NIS+、DNS のど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 *(*第 *1* 巻*)*』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を経由する「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」) を指定しない場合、`in.routed` は、ルーターで実行したときと同様にフルルーティングテーブルを作成します。
- `S` フラグを指定すると、`in.routed` は、使用可能な各ルーターについてデフォルトのルートを 1 つずつ示す最小核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、ルーター相互間での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` により実現されます。`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についてのデフォルトのルートに入りま

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。115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て、そのネットワーク上のすべてのマシンがそれぞれ一意な 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ことが重要です。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、334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な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 7–3 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。

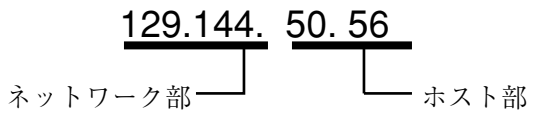


図 7–3　IPv4 アドレスの構成部分

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。これは、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 7–3 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。この番号は、ネットワーク内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、ネットマスクを使用して作成します。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 (詳細は、145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)。

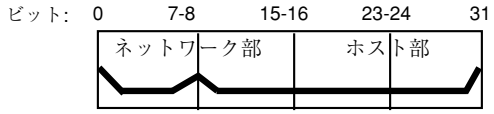
ネットワーククラス

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。それが済んだら、きわめて重要な第 2 のステップ、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移ることができます。

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、図 7–4 に示すように、IPv4 アドレスのホスト部です。



クラス A アドレス

図 7–4　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 7–5 に、クラス B のアドレスを示します。

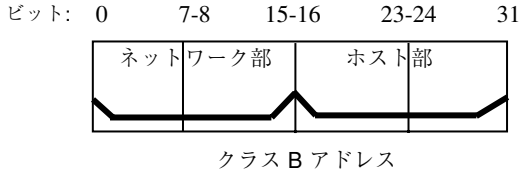


図 7–5　クラス B アドレスのバイト割り当て

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 7–6 に、クラス C アドレスのバイトを示します。

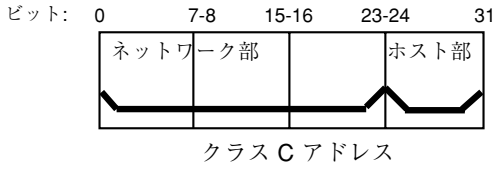


図 7–6　クラス C アドレスのバイト割り当て

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 8 章

DHCP の概要

この章では、ダイナミック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、以下の各項目について説明します。

DHCP について

DHCP は、TCP/IP ネットワーク内のホストシステムが、その起動時に TCP/IP ネットワークに対して自動的に構成されるようにするために開発された標準プロトコルです。DHCP は、クライアント/サーバーメカニズムを採用し、サーバーは、クライアントの設定情報を格納、管理し、クライアントの要求に応じてその設定情報を提供します。設定情報には、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 は、クライアントとサーバー間の通信に BOOTP と同じメッセージ形式を使用しながら、メッセージ内により多くの情報を含められるようにすること

で、BOOTP を拡張しています。追加された情報は、クライアントのためのネットワーク構成データです。

DHCP の主な利点は、リースによる IP アドレス割り当てを管理できる機能にあります。この機能を使用すると、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可能になります。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永続的なアドレスを割り当てた場合に比べ、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なければなりません。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設定しなければなりません。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なければなりません。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。クライアントが別のサブネットに移動する場合、クライアントはその新しいネットワークに適合する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、手動による再設定は必要ありません。
- 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 データ保存内に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なく、DHCP データ保存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、そのように構成すると、要求を送る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対して BOOTP 要求を渡すことができます。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設定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、サブネットを超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。

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。構成作業において、システム管理者は、クライアントがネットワーク上で動作するの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 8–1 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。

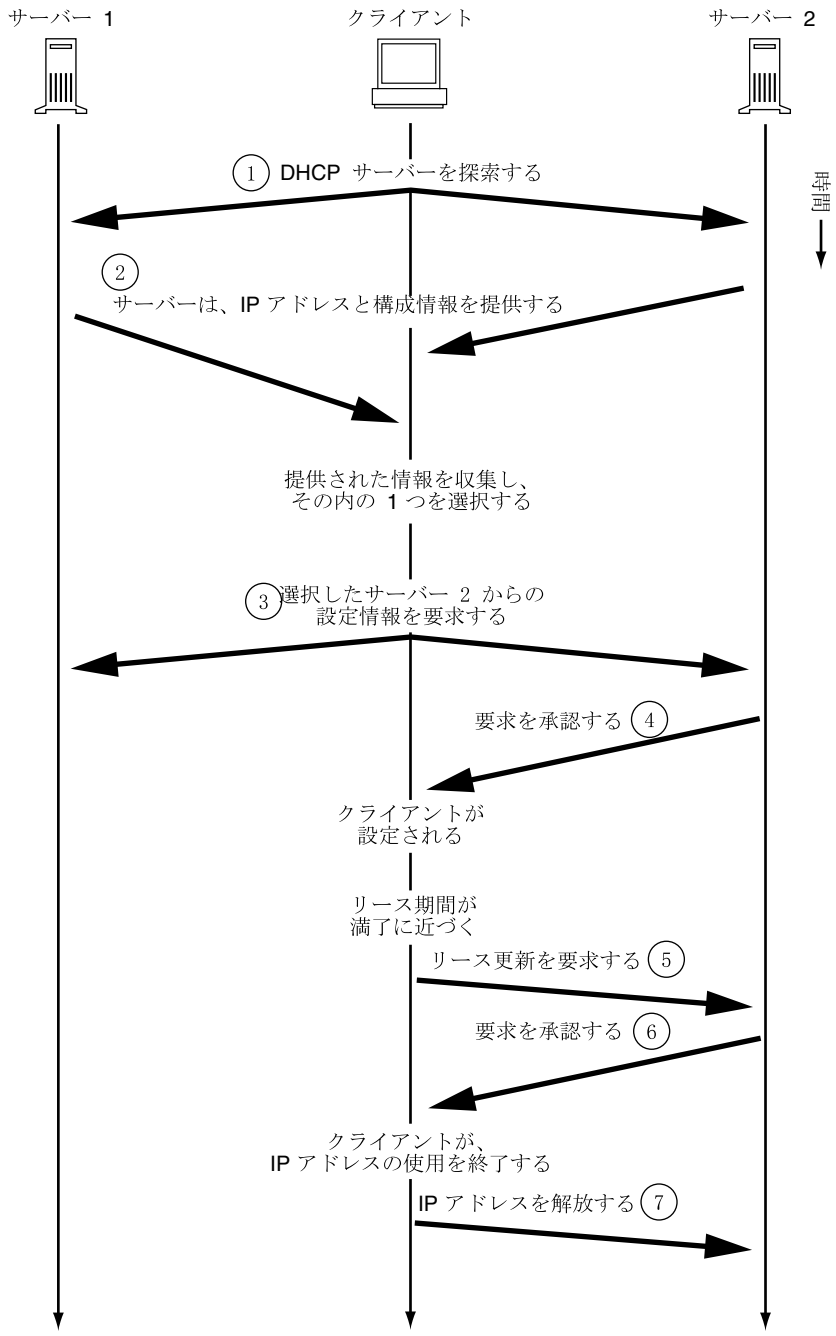


図 8–1　DHCP サービスにおける一連のイベント

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) に探索メッセージをブロードキャストすることで、DHCP サーバーを検出します。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、クライアントの一意の ID が含まれます。Solaris DHCP の実装では、この ID はクライアントの MAC アドレスから導出されます。

   探索メッセージを受け取る DHCP サーバーは、以下の情報を調査し、クライアントのネットワークを決定します。

   - サーバーのネットワークインタフェースの内、どのインタフェースが要求を受け取ったのか。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をキーにして、ネットマスクテーブルを調べます。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決定し、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. クライアントのネットワークを決定したら、DHCP サーバーは、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、クライアントの設定に使用可能なサービス情報を含むオファーメッセージ (offer message) 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. クライアントは最善のオファーを選択し (提供されるサービスの番号とタイプをもとに選択する)、要求をブロードキャストして、最善のオファーを行ったサーバーの IP アドレスを指定します。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、クライアントに対して IP アドレスを割り当て、その情報を DHCP データ記憶領域に格納し、クライアントに承認を送信します。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、他のシステムが IP アドレスを使用していないことを確認するために IP アドレスをテストし、ブート処理を継続し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 アドレスを必要としなくなった場合、IP アドレスを解放する旨を通知するメッセージをサーバーに送信します。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、クライアントに IP アドレスを永続的に、あるいは定義した期間割り当てます。DHCP サーバーはリースメカニズムを使用し、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定します。アドレスは、不要になるとプールに戻され、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。

- クライアントへのネットワーク構成情報の提供 – Solaris DHCP サーバーは、クライアントに IP アドレスを割り当てて、ネットワーク構成情報、たとえば、ホスト名、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、さらに、その他のさまざまな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設定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 サーバーを探索する 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設定された場合、DHCP サービスや BOOTP サービスを提供できなくなります。
- DHCP クライアントに対するネットワークブート機能のサポート – 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のに必要な情報、つまり、IP アドレス、ブートパラメータ、ネットワーク構成情報を DHCP クライアントに提供できます。

DHCP サーバーの管理

スーパーユーザーは、DHCP Manager を使用して、あるいはコマンド行ユーティリティを使用して、DHCP サーバーの起動、終了、および設定を行うことができます。通常 DHCP サーバーは、システムの起動時に自動的に立ち上がり、システムのシャットダウン時に終了するように設定されています。したがって、サーバーの起動と終了を手動で行うことは、ほとんどありません。

DHCP サーバーのデータ記憶領域

DHCP サーバーで使用されるすべてのデータは、2 つのデータリポジトリに格納されます。これら 2 つのデータリポジトリは、DHCP Manager またはコマンド行ユー

ティリティのいずれかを使用して表示および管理できます。2 つのデータリポジトリは、次のとおりです。

- `dhcptab` – クライアントに渡すことが可能な設定情報を含んでいるファイル
- DHCP ネットワークテーブル – テーブル名に指定されたネットワーク上に存在している DHCP および BOOTP クライアントに関する情報を含んでいるテーブル。たとえば、ネットワーク 134.20.0.0 には、`134_20_0_0` という名前のテーブルがある

ローカルディレクトリ上のファイル、または NIS+ データベースに、DHCP データを保存できます。データの保存方法の選択については、190ページの「データ保存方法の選択」を参照してください。

dhcptab ファイル

`dhcptab` ファイルには、クライアントが DHCP サーバーから入手できるすべての情報が入っています。DHCP サーバーは起動するたびにこのファイルを走査します。

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、一般的に使用されている標準オプションの一部を表 8–1 に示します。

表 8–1 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
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、設定情報を形成します。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は dhcptab(4) のマニュアルページに記載されています。DHCP Manager では、オプションとマクロタブに示されているすべての情報は dhcptab ファイルから得られます。オプションについては 177ページの「オプションについて」、マクロについては 178ページの「マクロについて」を参照してください。

dhcptab ファイルはテキストファイルですが、手作業では編集できないことに注意してください。オプションやマクロを作成、削除、変更するためには、dhtadm または DHCP Manager のいずれか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設定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は、dhcp_network(4) に記載されています。DHCP Manager では、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 Manager

DHCP Manager は、DHCP サービスに関連するすべての管理業務を行うためのグラフィカルツールです。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 Manager を下記の場合に使用することができます。

- DHCP サーバーを設定および設定解除する場合
- 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

また、DHCP Manager を使用すると、IP アドレス、ネットワーク構成マクロ、およびネットワーク構成オプションを下記のように管理することもできます。

- IP アドレスの表示、追加、削除、および解放
- ネットワーク構成マクロの表示、追加、変更、および削除
- 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、および削除

DHCP Manager では、このツールを使用して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。表 8–2 に、各ユーティリティとその使用目的を示します。表内のコマンド名をクリックすると、各ユーティリティの使用方法を詳しく説明したマニュアルページが表示されます。

表 8–2 DHCP コマンド行ユーティリティ

DHCP コマンドのマニュアルページ	説明と使用目的
`in.dhcpd(1M)`	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(1M)`	テキスト形式のメニューシステムを提供するシェルスクリプトで、DHCP サーバーの構成に役立つ。`dhcpconfig` は、サーバーマシンのネットワークトポロジファイルから情報を収集し、初期構成に役立つ情報を作成する。`dhcpconfig` は、バックグラウンドで `dhtadm` と `pntadm` ユーティリティを使用して初期 `dhcptab` テーブルとネットワークテーブルを作成する

表 8–2　DHCP コマンド行ユーティリティ　続く

DHCP コマンドのマニュアルページ	説明と使用目的
dhtadm(1M)	DHCP クライアント用の設定オプションとマクロの追加、削除、および変更に使用する。このユーティリティを使用するときは dhcptab ファイルを間接的に編集するため、dhcptab ファイルのフォーマットが正しいことが保証される。dhcptab ファイルは直接編集しないようにする
pntadm(1M)	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を使用すると、IP アドレスとネットワークを DHCP の管理下に (から) 追加 (削除) したり、指定した IP アドレスのネットワーク構成を変更したり、また DHCP の管理下にある IP アドレスとネットワークについての情報を表示したりできる

DHCP サーバーの設定

DHCP サーバーを動作させたいシステム上で DHCP Manager を初めて実行するときは、DHCP サーバーを設定します。DHCP Manager 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促す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 Manager 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
- データの保存方法 : ローカルファイルまたは NIS+ のいずれか
- 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)
- ルーターの探索、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設定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、有用な初期設定を提供します。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 コマンドが情報を入手するために使用するファイルについては、dhcpconfig(1M) を参照してください。初期設定後に設定を変更するときは、システムファイルに変更を加え、dhcpconfig コマンドを再度実行して、変更がコマンドに反映される様にする必要があります。

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。
- 自動または永続的な割り当て – 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設定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 DHCP クライアントにどのような情報を提供するかを決定します。DHCP サーバーを設定するときは、そのネットワークについての基本的な情報を提供しますが、後から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 サーバーアドレスは、DHCP サーバーにより割り当てられた値 (IP アドレスのリスト) を持つ、NISservrs と呼ばれる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。DHCP Manager を使用すると、これらのオプションに値を割り当て、これらのオプションをグループ化するためのマクロを作成することができます。グラフィカルツール以外のツールを使用したい場合は、DHCP 構成テーブル管理用ユーティリティの dhtadm を使用してオプションおよび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、オプションはクライアントに渡されるネットワーク情報です。また、DHCP の説明では、オプションはシンボルまたタグと呼ばれる場合もあります。オプションは数値コードおよびテキストラベルで定義され、値が割り当てられます。

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。標準オプションの一覧表を DHCP Manager のヘルプで見ることができます。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、使用しているネットワークに関連したオプションに値を割り当て、オプションと値のペアをマクロに含めることはできます。

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 3 つのカテゴリのいずれかに分類されるものでなければなりません。

- 拡張 – 使用している DHCP サーバーの実装にはまだ含まれていない、最新の標準 DHCP オプションのため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、ネットワークブートを行う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しなければならない、サーバーの IP アドレスを指定するために使用されます。

第 11 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、そのマクロで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。マクロの中には、特定のタイプのクライアント向けにサーバーが自動的に処理するものがあります。

マクロが自動的に処理されるようにするためには、表 8–3 に示されているカテゴリのいずれかに従ってそのマスクに名前を付ける必要があります。

表 8–3 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、サーバーのマクロの名前が `SUNW.Ultra-1` の場合、SUNW.Ultra-1 マシンであるクライアントはいずれも自動的に `SUNW.Ultra-1` マクロ内の値を受け取る
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 `125.53.224.0` の場合、125.53.224.0 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 `125.53.224.0` マクロ内の値を受け取る
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、サーバーのマクロ名が `08002011DF32` の場合、(Ethernet アドレス 8:0:20:11:DF:32 から導出される) `08002011DF32` の ID を持つクライアントは、`08002011DF32` の名前を持ったマクロ内の値を自動的に受け取る

表 8–3 に示されているカテゴリの 1 つを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る他のマクロに含まれる場合

注 - サーバーを設定する場合、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なマクロ
4. クライアント ID マクロ – 最も特定なマクロ

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、DHCP サーバーからそのネットワーク構成を受け取るように構成されているマシン上の 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。184ページの「複数のネットワークインタフェースを持つ DHCP クライアント」の項では 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に関する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 クライアント上で行う作業はこれだけです。

すでに Solaris 環境を稼動しているマシンで、DHCP を使用してネットワークインタフェースを構成する場合は、221ページの「Solaris DHCP クライアントの設定および設定解除」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が必要とする構成情報を取得します。このため、dhcpagent はシステム起動スクリプトによってブート処理の初期段階で起動されます。ブートは、ネットワーク構成情報が取得されるまで遅延されることになります。

/etc/dhcp.*interface* ファイル (たとえば、Sun Enterprise Ultra™ マシン上の /etc/dhcp.hme0) が存在すると、起動スクリプトは、DHCP が指定されたインタフェース上で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 を起動します。

起動後、dhcpagent はネットワークインタフェースの設定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 コマンドを発行し、167ページの「DHCP の動作」で説明しているように、dhcpagent に DHCP を起動するよう指示します。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 とで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。

DHCP クライアントのネットワーク構成情報の管理方法

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設定、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 は、メモリー内に保持された内部テーブル中に設定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 のテーブルから設定オプションの値を抽出します。それらの値はシステムの設定に使用され、ネットワークの一部を形成します。

エージェントは、設定された時間が経過するまで (通常はリース期間の半分まで) そのまま待機し、DHCP サーバーにリースの延長を要求します。dhcpagent はそのインタフェースの停止、または IP アドレスの変更を検出した場合、ifconfig からの

指示があるまでそのインタフェースの制御は行いません。また、dhcpagent は、そのインタフェースが適切に動作していること、および IP アドレスに変更が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。リースが更新できない場合、dhcpagent はリース期間の満了時にそのインタフェースを停止させます。

DHCP のクライアントの管理

Solaris DHCP クライアントは、通常のシステム動作時には管理を必要としません。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トし、システムの終了時に停止します。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。しかし必要であれば、クライアントマシン上でスーパーユーザーとして ifconfig コマンドを使用して、クライアントの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

ifconfig コマンドを使用すると、次のような操作が行えます。

- DHCP クライアントの起動 – ifconfig *interface* dhcp start コマンドは、IP アドレスと設定オプションの新規セットを取得するために、DHCP クライアントと DHCP サーバーの間の相互動作を開始します。このコマンドは、サーバー上のオプションの変更、たとえば、IP アドレスの追加、サブネットマスクの変更などを行い、変更結果を直ちにクライアントに反映させたい場合に便利です。
- 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、IP アドレスの管理には DHCP を使用しないが、ネットワーク上のホストの設定に DHCP を使用する場合が考えられます。
- リースの延長要求 – ifconfig *interface* dhcp extend コマンドは、dhcpagent がリース期間の更新を要求するようにします。この操作は自動的に実行されますが、リース期間を変更し、クライアントが新しいリース期間をただちに使用するようにしたい場合には、次のリース更新を待つよりも、手動でこのコマンドを実行するようにします。
- 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

放棄するようにします。この操作はリース満了時に自動的に実行されます。リース期間が長すぎ、ネットワークインタフェースを長期間停止させる必要がある場合、あるいは、ネットワークからシステムを切り離す場合は、このコマンドを発行するようにします。

- IP アドレスの停止 – `ifconfig` *interface* `dhcp drop` コマンドは、`dhcpagent` が DCHP サーバーへの通知を行なわず、ネットワークインタフェースを停止するようにします。この操作により、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の上での `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。表示される情報には、IP アドレスとクライアントの結合状態、送信、受信、および拒否された要求数、一次インタフェースかどうかを示すフラグ、リースが獲得された時刻、満了した時刻、およびリースが更新された時刻と、更新予定時刻が含まれます。表示例を以下に示します。

```
# ifconfig hme0 dhcp status
Interface  State         Sent  Recv  Declined  Flags
hme0       BOUND            1     1         0   [PRIMARY]
(Began, Expires, Renew) = (07/15/1999 15:27, 07/17/1999 13:31, 07/16/
1999 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` のための設定可能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。このファイルにはわかりやすい注釈が付けられているので、パラメータの詳細についてはファイル内の注釈をお読みください。

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停止するときは、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、リースは開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場合、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停止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許可しない場合には、クライアントは前述の DHCP プロトコルシークエンスを開始し、新しいネットワーク構成情報を取得します。

複数のネットワークインタフェースを持つ DHCP クライアント

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設定されている場合、クライアントはそれらのインタフェースを設定するために個別のリクエストを発行し、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、それら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。ホスト名、NIS ドメイン名、および時間帯のようなオプションが広域パラメータであり、これらのパラメータは各インタフェースについ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場合は、上記の事項には当てはまりません。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扱うインタフェースについては、/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 9 章

DHCP サービスの使用計画

DHCP サービスは、構築中のネットワークでも、既存のネットワークでも使用することができます。ネットワークの構築を行なっている場合は、DHCP サービスの設定を行う前に第 5 章を参照してください。ネットワークがすでに存在している場合には、この章をお読みください。

この章では、ネットワーク上で DHCP サービスを設定する前に行うべき作業について解説します。DHCP Manager で使用することを想定していますが、コマンド行ユーティリティの `dhcpconfig` を使用しても DHCP サービスの設定を行うことができます。

この章では、次の情報について説明します。

DHCP を使用するためのネットワークの準備

DHCP を使用するためにネットワークを構成する前に、情報を収集し、サーバー (複数も可) の設定方法について決定を行う必要があります。

以下の作業を行います。

- ネットワークトポロジについての詳細な計画を立て、どのサーバーがDHCP サーバーとして最善の候補であり、サーバーがいくつ必要になるのかを決定します。
- システムファイルと netmasks テーブルを更新し、ネットワークトポロジを正確に反映するようにします。DHCP サーバーが複数のリモートネットワーク上のクライアントをサポートするのであれば、それらのネットワークに対する netmasks テーブルのエントリが、最新のものであることを確認します (詳細は、netmask(4) のマニュアルページを参照してください)。
- 使用するデータ保存方法として、ローカルファイルまたはNIS+ のいずれかを選択します。
- リースポリシーを確立します。
- クライアントがルーター情報を取得する方法を決定します。

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構造またはレイアウトについてまだ計画を立てていない場合には、ルーターとクライアントの配置、およびネットワークサービスの提供を行うサーバーの配置を調べます。ネットワークトポロジのマッピングにより、DHCP サービスに使用するサーバー、および DHCP サーバーがクライアントに提供することのできる構成情報を決定することができます。

ネットワークの計画については、第 5 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。188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。クライアントに他のサービス情報を提供したい場合もあり、その場合にはサーバーのデータベースを検索する必要があります。ネットワークトポロジを調べる際に、クライアントに知らせておきたいすべてのサーバーの IP アドレスを記録しておきます。ネットワーク上に存在している可能性はあるが、その存在を 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次に示します。

- 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 System 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、複数のネットワークハードウェアインタフェースまたは複数の論理インタフェースのいずれかを介して、複数の IP ネットワークが同じネットワークハードウェア媒体を共有しているようなネットワーク環境では正しく動作しません。複数の IP ネットワークが同じ物理 LAN を通じて動作している場合、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到達し、それら IP ネットワークのすべてに対してそのクライアントが同時にアタッチされているように見えてしまいます。

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なければなりません。複数のネットワークがハードウェア媒体上に存在する場合、サーバーはクライアントのネットワークを特定することができず、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にそぐわない場合は、ネットワークを再構成する必要があります。次のような再構成が推奨候補として挙げられます。

- 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 (CDIR) 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設定し、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。ポートの設定については、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

データの保存方法は、DHCP クライアントをサポートするために必要なサーバー数に直接影響します。ローカルファイルを使用する方法では、1 つのサーバーは最大

10,000の クライアントをサポートできます。NIS+ を使用する場合、1 つのサーバーは最大 40,000 のクライアントをサポートすることができます。

190ページの「データ保存方法の選択」の節では、データの保存場所について説明しています。

システムファイルとネットマスクテーブルの更新

設定処理の間、DHCP Manager 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、サーバーの設定に使用できる情報を収集します。

DHCP Manager または dhcpconfig を使用してサーバーの設定を行う前に、システムファイル内の情報が最新のものであることを確認する必要があります。サーバーの設定後にエラーが検出された場合は、サーバー上のマクロを手動で変更し、dhcptab 構成テーブルを更新してください。

表 9–1 は、DCHP サーバーの設定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。DHCP をサーバー上に設定する前に、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。サーバーの設定後にシステムファイルを変更する場合には、DHCP Manager または dhcpconfig を再度実行し、変更を反映させる必要があります。

表 9–1 DHCP 構成のための情報

情報	提供元	説明
時間帯	システムの日時、時間帯設定	日時と時間帯は Solaris のインストール時に初期設定される。date コマンドを使用すると、日時を変更できる。時間帯を変更するには、TZ 変数が設定されている /etc/TIMEZONE ファイルを編集する
DNS パラメータ	/etc/resolv.conf	DHCP サーバーは /etc/resolv.conf を使用して、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 についての詳細は、『Solaris ネーミングの設定と構成』の「DNS クライアントの設定」を参照

表 9–1 DHCP 構成のための情報 続く

情報	提供元	説明
NIS+ パラメータ	システムのドメイン名、nsswitch.conf、NIS+	DHCP サーバーは domainname コマンドを使用してサーバーマシンのドメイン名を取得し、nsswitch.conf ファイルを使用してドメインベースの情報を検索する場所を特定する。サーバーマシンが 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、管理者の入力	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。同一ネットワーク上にないクライアントについては、DHCP サーバーは管理者を通じて情報を取得しなければならない
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 サーバーはそれ自身のネットワークインタフェースを参照し、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。リレーエージェントにより要求がすでに転送されている場合、サーバーはリレーエージェントのネットワーク上の netmasks テーブル内のサブネットマスクを検索する
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では、DHCP サーバーはネットワークインタフェースに照会することにり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

サーバー設定における決定事項

この節では、ネットワークで最初の DCHP サーバーを設定する前に決定する事項について説明します。トピックは DHCP Manager の構成ウィザード内のダイアログと並行して進行しますが、この節に示す情報は `dhcpconfig` ユーティリティを使用してサーバーの設定を行う場合にも有用です。

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、または Solaris 8 オペレーティング環境を稼動していること
- DHCP を使用する予定のクライアントが存在するすべてのネットワークに対して、直接ネットワーク経由で、あるいは BOOTP リレーエージェントを介してアクセス可能であること
- ルーティングを使用するように設定されていること
- ネットワークトポロジを反映する、正しく設定された `netmasks` テーブルがあること

データ保存方法の選択

DHCP データを保存するときは、ローカルディレクトリ内のファイルを使用して保存するか、NIS+ ディレクトリサービスの NIS+ テーブルを使用して保存するかを選択できます。NIS+ は分散データベースであるため、複数のサーバーが同じデータベースにアクセスすることができます。NIS+ はまた、本質的に高速な情報検索機能を備えています。このオプションを使用するためには、サーバーマシンが NIS+ クライアントとして設定されていなければなりません。

ファイルを使用する方法は、DHCP クライアント数が 10,000 以下のサイトでは効率的ですが、NIS+ に比べると幾分遅く、1 つのファイルシステム上にすべての DHCP データを保存しなければなりません。ファイルに保存されたデータは、NFS マウントポイントを通じてエクスポートされている場合にのみ、複数の DHCP サーバーで共有することができます。

従来の NIS は、NIS+ とは異なり、データ保存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 を使用している場合、データの保存にはファイルを使用する必要があります。

リースポリシーの設定

リースは、DHCP サーバーが DHCP クライアントに対して特定の IP アドレスの使用を許可する期間を指定するものです。サーバーの初期設定時に、リース期間と、クライアントがそれぞれのリース期間を更新することを許可するかどうかを示す、サイト全体に適用されるリースポリシーを指定する必要があります。サーバーは提供された情報を使用して、設定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。ユーザー自身が作成した構成マクロ内のオプションを設定することにより、特定のクライアント、または特定のクライアントタイプ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。クライアントが IP アドレスを割り当てられた場合 (あるいは、すでに割り当てられている IP アドレスについてリースのネゴシエーションを再度行う場合)、リース期間中に経過した時間数をクライアントの DHCP 肯定応答のタイムスタンプに加算することにより、リースが期限切れになる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 1999 年 9 月 16 日 9:15 A.M. で、リース期間が 24 時間の場合、リース満了時間は 1999 年 9 月 17 日 9:15 A.M. になります。リース満了時は、クライアントの DHCP ネットワークレコード内に保存され、DHCP Manager または `pntadm` を使用して表示できます。

リース期間は、期限切れになった IP アドレスが迅速に回収されるように、比較的小さな値に設定します。ただし、DHCP サービスが利用できなくなった場合、DHCP サービスを提供しているマシン (複数も可) が回復できるまで、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、予想されるサーバーのダウンタイムの 2 倍の時間を指定するようにします。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。リースについてのネゴシエーションが許可されている場合には、クライアントはそのリースで残っている時間を計測し、リース期間の半分が経過した時点で、リース期間を元の値まで延長するように、DHCP サーバーに要求します。リースネゴシエーションを不許可にする設定は、 IP アドレス数以上のマシンが存在するた

め、時間制限を IP アドレスの使用に課す方が好ましい環境に向いています。十分な数の IP アドレスがある場合は、クライアントが自身のネットワークインタフェースを停止して、新規のリースを取得しようとすることにより、TCP 接続 （NFS や telnet など) に悪影響をおよぼすことを回避するために、リースネゴシエーションを許可するようにします。リースネゴシエーションはサーバー設定時にサイト全体に渡って設定することが可能であり、また設定マクロの `LeaseNeg` オプションを使用して、特定のクライアントまたは特定のクライアントタイプにだけ設定することもできます。

注 - ネットワーク上でサービスを提供しているシステムはそれ自身の IP アドレスを保持する必要があり、短期的なリースの対象としてはいけません。永続的なリースにより IP アドレスを割り当てるのではなく、予約済みの (手動設定の) IP アドレスを割り当てることにより、そのようなマシンにも DHCP を使用することができます。これにより、マシンの IP アドレスがいつ使用されなくなるのか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

クライアントは、ルーターを使用して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を行うので、ルーターを使用するためにはルーターの IP アドレスを把握していなければなりません。

DHCP サーバーの設定時に、クライアントが使用可能なルーターの IP アドレスを提供する必要があります。DHCP Manager を使用する場合には、ルーター発見プロトコルを使用することにより、クライアント自身がルーターを検出するように指定することもできます。

ネットワーク上のクライアントがルーター発見をサポートしている場合、ルーターが 1 つしかない場合でも、IP アドレスを指定するのではなく、ルーター発見プロトコルを使用するようにします。ルーター発見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、新しいアドレスを持つルーターに置き変わった場合でも、新しいルーターのアドレスを取得するために新しいネットワーク構成を取得しなくても、クライアントは自動的に新しいアドレスを検出できます。

IP アドレス管理のための決定事項

この節では、DHCP で管理される IP アドレスを設定する際に必要となる決定事項について説明します。トピックは DHCP Manager のアドレスウィザードのダイアログと並行して進行しますが、この節に示す情報は `dhcpconfig` ユーティリティを使用して決定を行う場合にも有用です。

DHCP サービスの設定の一環として、サーバーにおける IP アドレスの扱い方を決定します。ネットワークに複数の DHCP サーバーが必要な場合、アドレス管理の分担方法を決定し、各サーバーにそれぞれの役割を割り当てるようにします。サーバーの設定を開始する前に、次の事項について決定します。

- サーバーが管理する IP アドレスの数または範囲
- サーバーがクライアントのホスト名を自動的に生成し、生成したホスト名に対して使用するプレフィックスを自動的に生成するようにするかどうか
- クライアントのネットワーク構成を割り当てるためにどの構成マクロを使用するか
- IP アドレスのリースを動的設定と永続的設定のどちらにするか

IP アドレスの数と範囲

DHCP Manager を使用すると、サーバーの初期設定時に、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 Manager はこの情報から連続するアドレスのリストを作成します。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、初期設定の後で DHCP Manager 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 アドレスの設定を行う前に、追加したい初期アドレスブロックにアドレスがいくつあるか、またその範囲内の開始アドレスの IP アドレスを把握しておきます。

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。Solaris DHCP サーバーは、オプショ

ンが指定されている場合には、各 IP アドレスに関連付けられたクライアント名を生成することができます。生成されたクライアント名は、/etc/hosts または NIS/NIS+ ホストテーブル内の IP アドレスに対応付けられます。クライアント名は、プレフィックス、つまりルート名、それに加え、サーバーによって割り当てられたダッシュと数字を使用します。たとえば、ルート名が charlie の場合、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。ある特定の DHCP サーバーが管理するクライアントを DHCP ネットワークテーブルで簡単に見つけることができます。ルート名は、ユーザーが選択した名前に変更することもできます。

IP アドレスを設定する前に、サーバーがクライアント名を生成するようにするかどうかを決定し、そのように決定した場合、クライアント名に使用するルート名を決定します。

注 - クライアント名は DNS ドメインに自動的に追加されないため、クライアント名はユーザーのネームサービス (NIS/NIS+) ドメインの外側では認識されませんが、手動で DNS にロードすることができます。DNSについての詳細は、『*Solaris* ネーミングの管理』「DNSの管理」と、in.named(1M) のマニュアルページ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。マクロは DHCP クライアントに送信するネットワーク構成情報を決定するために使用されます。

DHCP サーバーの初期設定時に、システムファイルから収集された情報とシステム管理者が指定した情報から、次のようないくつかのマクロが生成されます。

- ネットワークアドレスマクロ。このマクロは、クライアントネットワークの IP アドレスを使用して名前が設定され、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発見トークン、NIS/NIS+ ドメインとサーバー (サーバーが NIS/NIS+ を使用している場合) などの、ネットワークの一部を形成するクライアントが必要とする情報を含んでいます。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。

- ロケールマクロ。このマクロは、時間帯を指定するためのユニバーサル時間からのオフセット (秒単位) を含みます。
- サーバーマクロ。このマクロは、サーバーのホスト名を使用して名前が設定され、リースポリシー、タイムサーバー、DNS ドメイン、および DNS サーバーに関する情報、さらに、システムファイルから設定プログラムが取得可能であったその他の情報を含みます。このマクロはロケールマクロを含みます。

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に対する IP アドレスの設定時に、設定中のアドレスを使用して、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設定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。これは、このサーバーを使用するすべてのクライアントが必要とする情報をサーバーマクロが含んでいるためです。クライアントはネットワークアドレスマクロに含まれるオプションを、サーバーマクロに含まれるオプションの前に受け取ります。マクロ処理の順序についての詳細は、179ページの「マクロ処理の順序」を参照してください。

動的および永続的リースタイプ

リースタイプは、リースポリシー (リース期間とネゴシエーション) を設定中のアドレスに対して使用するかどうかを決定します。サーバーの初期設定時に、DHCP Manager は、追加するアドレスについて動的リースまたは永続的リースのいずれかを選択できるようにしています。`dhcpconfig` ユーティリティでは、動的リースのみが許可されます。

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。アドレスが永続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るだけで、その後クライアントは明示的にそのアドレスを解放するまでそのアドレスを所有します。アドレスが解放されると、サーバーはその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、永続的リースタイプに設定されている限り、リースポリシーの対象となることはありません。

IP アドレスの範囲を設定した場合、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、ほとんど

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。必要があれば、後で個々のアドレスを永続的リースに変更することもできますが、永続的リースの総数は最小限に抑えるようにします。

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、関連付けられた永続的リースまたは動的リースを持つことができます。予約されたアドレスが永続的リースで割り当てられた場合、そのアドレスは、そのアドレスに結合されているクライアントにのみ割り当てることができ、DHCP サーバーはそのアドレスを他のクライアントには割り当てることはできません。また、DHCP サーバーはそのアドレスを回収することもできません。

予約されたアドレスが動的リースで割り当てられた場合、そのアドレスは、そのアドレスが結合されているクライアントにだけ割り当てることができますが、そのクライアントはリース期間を計測し、そのアドレスが予約されていないものとして、リースの延長についてネゴシエートする必要があります。これにより、ネットワークテーブルを参照することで、そのクライアントがいつそのアドレスを使用しているかを追跡することができます。

初期設定時には、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 サーバーを設定して、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、データ保存方式に NIS+ を選択します。そうでなければ、ファイルを選択し、データ保存用の絶対パスに共有ディレクトリを指定します。
- 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設定します。

- 指定された時間間隔で dhcptab ファイル内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、各サーバーが最新の情報を使用するようにします。これは、in.dhcpd(1M) に –t オプションを指定することで行えます。
- すべてのクライアントがすべての DHCP サーバーに確実にアクセスできるようにして、サーバーが互いにサポートしあえるようにします。クライアントが有効な IP アドレスリースを持っており、その設定を検証しようとしているか、そのリースを延長しようとしているときに、そのクライアントのアドレスを所有しているサーバーが到達不能である場合、他のサーバーは、そのクライアントが元のサーバーに 20 秒間コンタクトを試みた後で、そのクライアントに応答することができます。あるクライアントが特定のアドレスを要求し、そのアドレスを所有しているサーバーが利用できない場合、他のサーバーのいずれかがその要求を処理します。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

初期設定が完了すると、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。ところが、システムファイルはサーバーに対してローカルではないため、ほとんどの情報は検索できず、デフォルト値を取得することはできません。そのため、ユーザー自身が情報を提供する必要があります。リモートネットワークの設定を行う前に、次の情報を入手しておきます。

- 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+ を使用する場合には、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、ローカルエリアネットワーク (LAN) 接続と PPP (ポイントツーポイント) プロトコルのどちらを使用してネットワークに接続しているか

- ルーティング – クライアントがルーター発見を使用できるか。使用できない場合には、クライアントが使用可能なルーターの IP アドレスをユーザー自身が見つける必要があります。
- 使用可能であれば、NIS ドメインと NIS サーバー
- 使用可能であれば、NIS+ ドメインと NIS+ サーバー

DHCP ネットワークを追加する手順については、243ページの「DHCP ネットワークの追加、変更、削除」を参照してください。

DHCP を設定するためのツールの選択

ここまでの節で説明したように、情報を収集し、設定方法について決定を行うと、DHCP サーバーの設定準備が完了したことになります。GUI 対応の DHCP Manager、またはコマンド行ユーティリティの `dhcpconfig` を使用して、サーバーの設定を行うことができます。どちらのツールを使用しても、オプションを選択したり、DHCP サーバーが使用する `dhcptab` とネットワークテーブルを作成するためのデータを入力したりできます。

DHCP Manager の機能

DHCP Manager は Java 対応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 サーバーとして設定されていないシステム上で DHCP Manager を最初に実行したときに自動的に起動されます。DHCP 構成ウィザードは、一連のダイアログボックスで構成され、サーバーの設定に必要な基本的な情報、たとえば、データ保存方式、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の入力を求めます。情報の中には、ウィザードによりシステムファイルから取得されるものがあります。これらの情報は正しいことを確認するだけでよく、必要な場合にのみ訂正します。

ダイアログボックスに従って作業を進め、情報を確認します。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 アドレスを設定するために、追加アドレスウィザードを起動するよう求められます。初期設定時には、DHCP に対してサーバーのネットワークだけが設定され、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。初期設定が完了した後で DHCP

Manager 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、dhtadm コマンドと pntadm コマンド用のラッパー (wrapper) スクリプトです。dhcpconfig ユーティリティは、dhcptab を読み込む間隔、DHCP サービスが提供するタイムアウト値などのサーバー起動オプションの入力を求めます。またこのユーティリティは、188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その他の情報も取得します。システムファイルから取得された情報は見ることができないため、dhcpconfig を起動する前にシステムファイルを更新しておくことが重要です。

なお、dhcpconfig はシステムファイルを更新した後で、再度実行することができ、DHCP データを適切に更新できます。

DHCP Manager と dhcpconfig の比較

表 9–2 に、2 つのサーバー設定ツールの相違点を示します。

表 9–2 DHCP Manager と dhcpconfig の比較

比較項目	**DHCP Manager**	dhcpconfig
システムから収集したネットワーク情報の表示	システムファイルから収集された情報を表示でき、必要があれば変更可能	dhcpconfig が収集した情報は見ることができない。生成後に dhcptab とネットワークテーブルを調べる必要がある
ユーザーに求められる設定経験	デフォルト値を使用することで基本的でないサーバーオプションの入力を省略し、設定作業を高速化。基本的でないオプションは初期設定後に変更可能	設定作業中にすべてのサーバーオプションを入力。後でオプションを変更するには、dhcpconfig を再度起動するか、dhtadm と pntadm コマンドを使用する
ユーザー入力のチェック	入力時にユーザー入力の有効性をチェックする	入力時にユーザー入力の有効性をチェックしない

次の章では、DHCP Manager と `dhcpconfig` の両方を使用したサーバーの設定手順について説明します。

第 10 章

DHCP サービスの設定

ネットワーク上で DHCP サービスを設定するときは、最初のDHCP サーバーの設定と起動が中心的な作業になります。その他のサーバーは後から追加することが可能であり、追加したサーバーは共有ロケーションから同じデータにアクセスできます。この章では、DHCP サーバーの設定手順と、ネットワークおよびそれらのネットワークに対応する IP アドレスを DHCP の管理下に置くための手順について説明します。また、サーバーの設定解除についても解説します。

この章では、DHCP Manager を使用した設定手順と、`dhcpconfig` を使用した設定手順を、それぞれの節で説明します。この章には次のような情報が含まれます。

- 201ページの「DHCP Manager の使用による DHCP サーバーの設定および設定解除」
- 209ページの「`dhcpconfig` を使用する DHCP サーバーの設定および設定解除」
- 221ページの「Solaris DHCP クライアントの設定および設定解除」

DHCP Manager の使用による DHCP サーバーの設定および設定解除

この節では、 DHCP Manager を使用してサーバーを設定および設定解除する手順について説明します。なお、DHCP Manager を使用するには、CDE などの X Window System が動作していなければなりません。

最初にシステム上で DHCP Manager を起動すると、図 10–1 ような画面が表示され、DHCP サーバーまたは BOOTP リレーエージェントのどちらを設定するか選択できます。

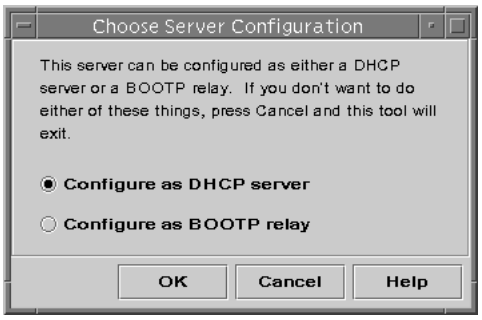


図 10–1　「サーバー構成の選択 (Choose Server Configuration)」ダイアログ

DHCP サーバーの設定

DHCP サーバーの設定を行う場合、DHCP Manager が DHCP 構成ウィザードを起動し、このウィザードはサーバーの設定に必要な情報の入力を促します。図 10–2 に示すような、ウィザードの初期画面が現れます。

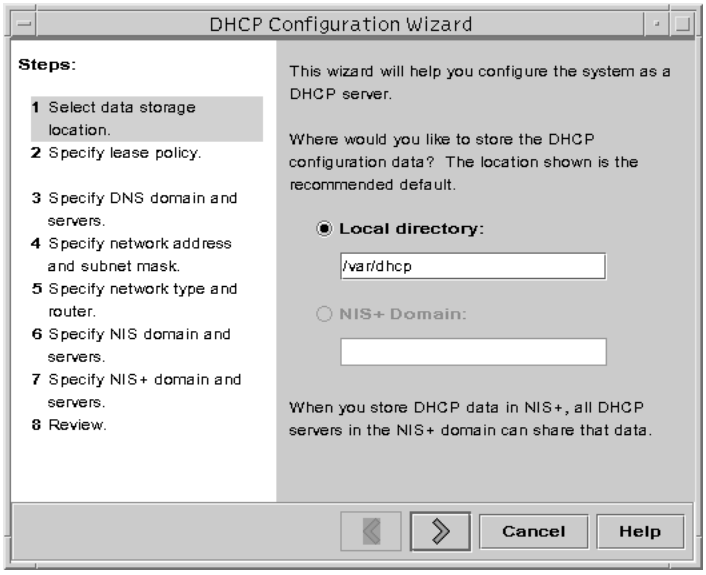


図 10–2　「DHCP 構成ウィザード (DHCP Configuration Wizard)」の初期画面

ウィザードの入力要求に応答すると、DHCP Manager は 表 10–1 に列記されている項目を作成します。

表 10–1 DHCP サーバーの設定時に作成される項目

項目	説明	内容
サービス構成ファイル /etc/default/dhcp	サーバー設定オプションのキーワードおよび値を記録する	データ保存形式とその場所、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 ファイル	dhcptab ファイルがまだ存在しない場合、DHCP Manager は空の dhcptab ファイルを生成する	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) からのローカルな時間帯のオフセット (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 サーバーを設定する管理者の入力を使用して設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 マクロと次のオプション ■ 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が設定されている場合) ■ 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なくてはならないことを意味する

表 10–1　DHCP サーバーの設定時に作成される項目　続く

項目	説明	内容
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である	DHCP サーバーを設定する管理者の入力により設定された値を持つオプションが含まれる。オプションは、マクロ名に一致する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 NISserv (NIS が設定されている場合) ■ 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 サービススクリプト /etc/init.d/dhcp へのリンク	システムのブート時に、DHCP デーモンが自動的に起動できるようにする	次のリンクが作成される ■ /etc/rc0.d/K34dhcp ■ /etc/rc1.d/K34dhcp ■ /etc/rc2.d/K34dhcp ■ /etc/rc3.d/K34dhcp

▼ DHCP サーバーの設定方法 (DHCP Manager)

1. DHCP サーバーとして使用するシステムを選択します。

190ページの「サーバー設定における決定事項」のガイドラインに従います。

2. データ保存、リースポリシー、およびルーター情報について決定を行います。

190ページの「サーバー設定における決定事項」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」オプションを選択します。

   DHCP 構成ウィザードが起動し、サーバーの設定方法を指示します。

6. 計画作成段階で決定した決定事項に基づいて、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れば、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、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」をクリックしてサーバー設定を完了します。

8. アドレス起動ウィザードウィンドウで「はい」をクリックし、サーバーのアドレスの設定を開始します。

   このウィザードでは、DHCP の制御下に置くアドレスを指定することができます。

9. 計画作成段階での決定事項に従って、入力要求に応答します。

   詳細は、193ページの「IP アドレス管理のための決定事項」を参照してください。わからないことがあれば、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、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」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。

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

ネットワークウィザードを使用して、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。247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設定

BOOTP リレーエージェントを設定するときは、DHCP Manager は次の動作を行います。

- 要求をリレーする DHCP サーバーの IP アドレスの入力要求
- BOOTP リレーサービスに必要なオプションを指定するための `/etc/default/dhcp` の編集
- システムのブート時に DHCP デーモンが自動的に起動できるよう、`/etc/init.d/dhcp` への次のリンクの作成
  - `/etc/rc0.d/K34dhcp`
  - `/etc/rc1.d/K34dhcp`
  - `/etc/rc2.d/K34dhcp`
  - `/etc/rc3.d/K34dhcp`

BOOTP リレーエージェントの設定を選択した場合に表示される画面を図 10–3 に示します。

図 10–3　「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の設定方法 (DHCP Manager)

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。

   190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設定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 サーバーとして設定されている場合には、そのサーバーの設定解除をしなければ、そのサーバーを BOOTP リレーエージェントとして設定することはできません。207ページの「DHCP サーバーと BOOTP リレーエージェントの設定解除」を参照してください。

4. 「**BOOTP** リレーとして構成」を選択します。

   「BOOTP リレーの構成」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により受信される **BOOTP** または **DHCP** 要求を処理するように設定されている、**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」をクリックします。

   「了解」をクリックすると、DHCP Manager はアプリケーションを終了するための「ファイル」メニューと、サーバーを管理するための「サービス」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設定解除

DHCP サーバーまたは BOOTP リレーエージェントを設定解除するときは、DHCP Manager は次の動作を行います。

- DHCP デーモン (`in.dhcpcd`) プロセスの停止
- システムブート時に自動起動を可能にするリンクの削除

- デーモンの起動に関する情報とデータ保存場所を記録している /etc/default/dhcp ファイルの削除

DHCP サーバーの設定解除を選択した場合の画面を図 10–4 に示します。

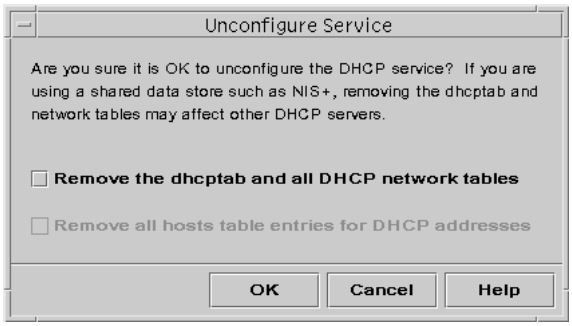


図 10–4 「サービスの解除 (Unconfigure Service)」ダイアログボックス

DHCP サーバーの設定解除を行う場合、DHCP データファイル (dhcptab と DHCP ネットワークテーブル) の取り扱いを決める必要があります。データが複数のサーバー間で共有される場合、dhcptab および DHCP ネットワークテーブルは削除しないようにします。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ります。データは NIS+ 経由で、またはエクスポートされたローカルファイルシステム上で共有することができます。/etc/default/dhcp ファイルには、使用するデータ保存方式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 サーバーを設定解除できます。データをそのままの形で残し、設定を解除すると、DHCP サーバーが使用できなくなります。

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 アドレスに対応するクライアント名をすでに生成している場合には、/etc/inet/hosts または NIS+ の hosts テーブルからそれらのエントリを削除するように選択できます。

BOOTP リレーエージェントを設定解除する前に、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の設定解除方法 (DHCP Manager)

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」メニューから、「解除」を選択します。

   「サービスの解除」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設定解除することが確認できます。サーバーが DHCP サーバーの場合、このダイアログボックス内で選択を行うことにより、DHCP データの扱いについて決定を行う必要があります。図 10–4 を参照してください。

4. 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を使用している場合 (NIS+、または NFS 経由で共有されているファイルを使用している場合)、データを削除するオプションは選択しないでください。共有データを使用していない DHCP サーバーの場合には、いずれかの、または両方のオプションを選択して、データを削除します。

   データの削除については、207ページの「DHCP サーバーと BOOTP リレーエージェントの設定解除」を参照してください。

5. 「了解」をクリックします。

dhcpconfig を使用する DHCP サーバーの設定および設定解除

この節では、dhcpconfig を使用して DHCP サーバーまたは BOOTP リレーエージェントを設定および設定解除する手順について説明します。dhcpconfig の起動時に、図 10–5 のような「DHCP Configuration」メニューが表示されます。

```
***          DHCP Configuration              ***

Would you like to:

        1) Configure DHCP Service

        2) Configure BOOTP Relay Agent

        3) Unconfigure DHCP or Relay Service

        4) Exit

Choice:
```

図 10–5　dhcpconfig のメニュー

▼ DHCP サーバーの設定方法 (dhcpconfig)

1. DHCP サーバーとして使用するシステムを選択します。

190ページの「サーバー設定における決定事項」のガイドラインに従います。

2. データ保存、リースポリシー、およびルーター情報について決定を行います。

190ページの「サーバー設定における決定事項」のガイドラインに従います。

3. 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/sbin/dhcpconfig
```

テキスト形式の「DHCP Configuration」メニューが表示されます。

5. 1 と入力し、Return を押して、「**Configure DHCP Service**」を選択します。

6. 次のプロンプトに対して応答を入力します。

ここでは、第 9 章を読んで決定した決定事項を使用します。なお、各プロンプトにおけるデフォルト値は角括弧で囲まれています。デフォルト値を使用する場合には、そのプロンプトで Return を押します。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N)Y
```

Y を押し、DHCP サービスを停止します。この操作により、サーバーが不完全な構成情報をクライアントに送信しないようにします。

```
###     DHCP Service Configuration      ###
###     Configure DHCP Database Type and  Location      ###

Enter datastore (files or nisplus) [nisplus]:
```

使用するデータ保存方法の名前、(files または nisplus) を入力します。

データ保存方法についてのより詳しい情報が必要な場合は、190ページの「データ保存方法の選択」のガイドラインを参照してください。選択結果は /etc/default/dhcp に記録されます。

```
Enter absolute path to datastore directory [/var/dhcp]:
```

データ保存方式に使用する files または NIS+ ディレクトリへのパスを入力します。データ保存方式に files を選択した場合のデフォルトディレクトリは /var/dhcp です。NIS+ を選択した場合のデフォルトディレクトリは、yourcompany.com などの、サーバーが現在使用している NIS+ ディレクトリの場所になります。

```
Would you like to specify nondefault daemon options (Y/[N]):
```

上記のプロンプトで N を入力した場合、デフォルト以外のデーモンオプションを指定しなくても、サーバーを正常に設定できます。

Y と入力した場合は、次のプロンプトが表示されます。

```
Do you want to enable transaction logging? (Y/[N]):Y
```

トランザクションログを有効にしたい場合は、Y と入力します。トランザクションログについては、第 11 章を参照してください。トランザクションログを有効に設定した場合にのみ、次のプロンプトが表示されます。

```
Which syslog local facility [0-7] do you wish to log to? [0]:
```

トランザクションログを記録するためのローカル機能については、第 11 章を参照してください。

```
How long (in seconds) should the DHCP server keep outstanding OFFERs? [10]:
```

サーバーがクライアントに提供する IP アドレスをキャッシュに保持する秒数を入力します。デフォルトの秒数は 10 秒であり、ほとんどのネットワークでは、これが適切な秒数です。秒数を増加することは可能ですが、ネットワーク性能が低下します。

```
How often (in minutes) should the DHCP server rescan the dhcptab? [Never]:
```

デフォルトでは DHCP サーバーは起動時に、あるいは DHCP Manager から読み込みの指示を受けた場合にのみ、dhcptab を読み込みます。DHCP Manager を使用すると、設定データを変更した後で dhcptab をリロードすることにより、サーバーを更新することができます。そのため、DHCP Manager を使用している場合は、dhcptab の自動再走査は必要ありません。一般に、再走査間隔は次のような状況でのみ使用します。

- データ保存が NIS+ 内に配置されており、ネットワーク上に複数の DHCP サーバーが存在する場合。再走査によって、すべてのサーバーが最新の情報を持っていることが保証されます。
- DHCP Manager ではなく、dhtadm を使用して設定変更を行った場合。dhtadm ユーティリティには、設定変更後に dhcptab の再走査を強制するオプションはありません。

dhcptab に対して自動再走査を実行する場合は、サーバーが dhcptab ファイル内にクライアントの設定情報をリロードするまでの待機時間間隔を分単位で入力します。

```
Do you want to enable BOOTP compatibility mode? (Y/[N]):
```

デフォルトでは BOOTP 互換性は無効になっています。BOOTP 互換性を有効にする場合は、253ページの「DHCP サービスを使用した BOOTP クライアントのサポート」を参照してください。

デフォルト以外のデーモンおよびサーバーオプションに関する情報の入力が終了したら、次のプロンプトが表示されます。

```
Enter default DHCP lease policy (in days) [3]:
```

リース期間の日数を入力します。デフォルト設定は 3 日です。詳細については、191ページの「リースポリシーの設定」 を参照してください。

```
Do you want to allow clients to renegotiate their leases? ([Y]/N):
```

デフォルト設定はリースネゴシエーションを有効にする Y です。リースネゴシエーションについての詳細は、191ページの「リースポリシーの設定」を参照し

てください。N と入力した場合、クライアントはリースの期限が切れた時点で IP アドレスを放棄し、新たなリースと IP アドレスを取得しなければなりません。

```
Enable DHCP/BOOTP support of networks you select? ([Y]/N):
```

この時点で、DHCP を使用するネットワークの構成が可能になります。193ページの「IP アドレス管理のための決定事項」を読んだ後に選んだ決定事項を参照してください。IP アドレスを設定する準備が整っていない場合は、N と入力します。dhcpconfig は DHCP サービスを再度実行するように促すプロンプトを表示して、初期メニューに戻ります。少なくとも 1 つのネットワーク上で DHCP または BOOTP のサポートを有効にしない限り、DHCP を使用することはできません。

IP アドレスを設定する準備が整っている場合は、Y と入力し 215ページの手順 4 へ進みます。

▼ BOOTP リレーエージェントの設定方法 (dhcpconfig)

1. 設定を行うシステム上でスーパーユーザーになります。

2. 次のように入力します。

```
# /usr/sbin/dhcpconfig
```

テキスト形式の「Configure DHCP Service」メニューが表示されます。

3. 2 と入力し、Return を押して「**Configure BOOTP Relay Agent**」を選択します。

4. プロンプトに対し、次のように応答します。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N):Y
###     BOOTP Relay Agent Configuration ###

###     Common daemon option setup      ###
Would you like to specify nondefault daemon options (Y/[N]):Y
```

このプロンプトに対して N と入力した場合、デフォルト以外のデーモンオプションを指定しなくても、サーバーを正常に設定できます。

Y と入力した場合には、次のプロンプトが表示されます。

```
Do you want to enable transaction logging? (Y/[N]):Y
```

トランザクションログを有効にしたい場合は、Y と入力します。トランザクションログについては、第 11 章を参照してください。トランザクションログを有効にした場合にのみ、次のプロンプトが表示されます。

```
Which syslog local facility [0-7] do you wish to log to? [0]:
```

トランザクションログを記録するためのローカル機能については、第 11 章を参照してください。

```
Enter destination BOOTP/DHCP servers. Type '.' when finished.
IP address or Hostname:
```

要求の転送先となる BOOTP または DHCP サーバーの IP アドレスまたはホストネームを 1 つ入力し、Return を押します。プロンプトが再表示され、アドレスおよびホストネームの入力が引き続き可能な状態になります。終了する場合はピリオド (.) を入力し、Return を押します。

```
Would you like to restart the DHCP service? (recommended) ([Y]/N):Y
```

「Configure DHCP Service」メニューが再表示されます。

5. dhcpconfig を終了するには、4 と入力します。

dhcpconfig の使用によるネットワークの構成

この項では dhcpconfig を使用して、ネットワークを DHCP の管理下に置くための手順について説明します。以下の各手順では、サーバーの設定が完了し、DHCP データ保存に新たなネットワークを追加している段階であることを前提としています。

▼ ローカルネットワークの構成方法 (dhcpconfig)

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。

```
# /usr/sbin/dhcpconfig
```

テキスト形式の「Configure DHCP Service」メニューが表示されます。

3. プロンプトに対して次のように応答し、ネットワーク構成オプションへ進みます。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N)Y
Would you like to specify nondefault daemon options (Y/[N]):N
Do you want to merge initialization data with the existing table? (Y/[N]):Y
Enable DHCP/BOOTP support of networks you select? ([Y]/N):Y
```

4. プロンプトに対して次のように応答し、ローカルネットワークを構成します。

```
Configure BOOTP/DHCP on local LAN network: 172.21.0.0? ([Y]/N):Y
```

5. クライアント名の生成についての各プロンプトに対して、次のように応答します。

```
Do you want hostnames generated and inserted in the files hosts table? (Y/[N]):
```

サーバーはホスト名を作成し、各名前を IP アドレスに対応付けることができます。詳細については、193ページの「クライアントホスト名の生成」を参照してください。

Y と入力した場合は次のプロンプトに応答します。N と入力した場合は、216ページの手順 6 へ進みます。

```
What rootname do you want to use for generated names? [yourserver-]:
```

生成されたクライアント名のデフォルトのプレフィックス、つまりルート名 (rootname) が DHCP サーバーの名前になります。この名前はそのままクライアント名として使用しても、他の名前に変更してもかまいません。

```
Is Rootname name_you_typed- correct? ([Y]/N):Y
```

入力に間違いがあった場合は、ここで N と入力し再度ルート名の入力を求めるプロンプトへ戻ります。

```
What base number do you want to start with? [1]:
```

ベース番号はクライアント名の生成に用いられるルート名に付加される最初の番号です。たとえば、デフォルトのルート名とベース番号を使用した場合、クライアント名は、yourserver-1、yourserver-2 のようになります。

6. **DHCP** の管理下に置きたいネットワークの **IP** アドレスに関する次のプロンプトに応答します。

```
Enter starting IP address [172.21.0.0]:
```

サーバーは管理対象とする IP アドレス範囲を生成する必要があります。DHCP の管理下に置きたい IP アドレス範囲の開始アドレスを入力します。詳細については、193ページの「IP アドレスの数と範囲」を参照してください。

```
Enter the number of clients you want to add (x < 65535):
```

クライアント数は DHCP の管理下に置く IP アドレスの数に相当します。dhcpconfig プログラムはこの情報とベース番号を使用して、連続した IP アドレスブロックを DHCP の管理下に追加します。

```
The dhcp network table: 172.21.0.0 already exists.
Do you want to add entries to it? ([Y]/N):
```

すでにアドレス設定が完了しているネットワーク内にアドレスブロックを追加した場合、上記のプロンプトが表示されます。Y と入力してネットワークテーブルを変更しアドレスを追加します。

```
Would you like to configure BOOTP/DHCP service on remote networks? ([Y]/N):
```

ネットワークの追加を終了するときは、N と入力し、プロンプトの指示に従ってサーバーを再起動します。

他のネットワークの IP アドレスを DHCP の管理下に置きたいときは、このプロンプトで Y と入力し、217ページの手順 5 に進みます。

▼ リモートネットワークの構成方法 (dhcpconfig)

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。

```
# /usr/sbin/dhcpconfig
```

テキスト形式の「DHPC Configuration」メニューが表示されます。

3. 1 と入力し、Return を押して「**Configure DHCP Service**」を選択します。
4. プロンプトに対して次のように応答し、リモートネットワーク構成オプションへ進みます。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N)Y
Would you like to specify nondefault daemon options (Y/[N]):N
Do you want to merge initialization data with the existing table? (Y/[N]):Y
Enable DHCP/BOOTP support of networks you select? ([Y]/N):Y
Configure BOOTP/DHCP on local LAN network: 172.21.0.0? ([Y]/N):N
```

5. プロンプトに対して次のように応答し、リモートネットワークを構成します。

```
Would you like to configure BOOTP/DHCP service on remote networks? ([Y]/N):Y
Enter Network Address of remote network, or <RETURN> if finished:
```

DHCP に構成したいネットワークの IP アドレスを入力します。ネットワークアドレスでは、IP アドレスのホスト部分に 0 が使用されます。

```
Do clients access this remote network via LAN or PPP connection? ([L]/P):
```

L または P と入力することにより、ネットワークがローカルエリアネットワーク (LAN) またはポイントツーポイントプロトコルネットワーク (PPP) のいずれであるかを示します。

```
Do you want hostnames generated and inserted in the files hosts table? (Y/[N]):
```

サーバーは IP アドレスごとにホスト名を作成し、/etc/inet/hosts ファイルまたは NIS+ hosts テーブルにエントリを作成します。193ページの「クライアントホスト名の生成」を参照してください。

```
Enter Router (From client's perspective), or <RETURN> if finished.
IP address:
```

このネットワーク上のクライアントが使用するルーター (複数も可) の IP アドレスを入力します。なお、クライアントがこのネットワークでルーター発見機能を使用するように指定することはできません。

```
Optional: Enter Remote Network's MTU (e.g. ethernet == 1500):
```

リモートネットワークが使用している固有の最大転送単位 (MTU) がわかっている場合は、この時点で入力します。そうでなければ、Return を押して、デフォルト値をそのまま使用します。

```
Enter starting IP address [172.21.0.0]
```

DHCP の管理下に置きたい IP アドレス範囲の開始 IP アドレスを入力します。デフォルト値は、ネットワークアドレスです。

```
Enter the number of clients you want to add (x < 65535):
```

DHCP の管理下に置きたい IP アドレスの個数を入力します。dhcpconfig は、この値と、直前に入力した開始 IP アドレスを基に、DHCP の管理下に置く一連の IP アドレスを決定します。入力する値は、プロンプトで示される値よりも小さくなればなりません。なお、プロンプトで表示される値は、ネットマスクを基に生成されたものです。上記の例の場合は、65535 未満の値を指定する必要があります。

```
dhcptab macro "172.21.0.0" already exists.
Do you want to merge initialization data with the existing macro? ([Y]/
N):
```

このネットワークがすでに構成済みの場合は、上記のメッセージが表示されます。既存のマクロにデータをマージする必要があるのは、追加しているネットワーク上の全クライントに指定の情報を適用する場合だけです。

```
Disable (ping) verification of 172.21.0.0 address(es)? (Y/[N]):
```

dhcpconfig は、追加しているアドレスに対して ping を実行し、それらのアドレスが使用されていないことを確認し、使用中のアドレスはスキップします。このプロンプトに対し Y と入力すると、dhcpconfig は、アドレスに対して ping を実行しません。

```
Network: 172.21.0.0 complete.
Enter Network Address of remote network, or <RETURN>  if finished:
```

別のリモートネットワークを構成したい場合は、そのネットワークアドレスを入力し、ネットワークに関するプロンプトに応答します。構成するリモートネットワークが他にない場合は、上記のプロンプトに対して Return を押します。

```
Would you like to restart the DHCP service? (recommended) ([Y]/N):
```

Y と入力して、DHCP サービスを再起動します。

dhcpconfig の使用による DHCP サーバーおよび BOOTP リレーエージェントの設定解除

DHCP サーバーの設定を解除すると、サーバープロセスが停止し、システムのリブート時に自動的に起動されなくなります。また、これによって、/etc/default/dhcp ファイルが削除されます。このファイルには、データの保存場所およびサーバーの起動オプションに関する情報が格納されています。DHCP サーバーの設定を解除する際には、DHCP データファイル (dhcptab と DHCP の各ネットワークテーブル) の扱い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てはなりません。DHCP サーバーの設定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えます。これらのテーブルを保持したままでも、DHCP サーバーの設定を解除することができます。テーブルをそのまま保持するには、テーブルを削除するかどうかを尋ねるプロンプトに、NO と応答します。

▼ DHCP サーバーまたは BOOTP リレーエージェントの設定解除方法 (dhcpconfig)

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. 次のコマンドを入力します。

```
# /usr/sbin/dhcpconfig
```

テキスト形式の「DHPC Configuration」メニューが表示されます。

3. 3 と入力し、Return を押して、「**Unconfigure DHCP or Relay Service**」を選択します。

4. プロンプトに対して次のように応答します。

```
Unconfigure will stop the DHCP service and remove /etc/default/dhcp.
Are you SURE you want to disable the DHCP service? ([Y]/N):
```

Y と入力し、サーバーの設定を解除します。

```
Are you SURE you want to remove the DHCP tables? (Y/[N]):
```

DHCP データが他の DHCP サーバーと共有されていないことが確実な場合にだけ、Y と入力します。N と入力すると、サーバーが使用できなくなりますが、データはそのままに保持されます。

Solaris DHCP クライアントの設定および設定解除

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設定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 yes と応答すると、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 を使用するためにクライアントマシン上で行う作業は、これ以外にはありません。

クライアントマシン上ですでに Solaris オペレーティング環境が稼働しているが、DHCP をまだ使用していない場合は、Solaris システムの設定を解除し、いくつかのコマンドを発行して、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。

Solaris 以外のクライアントを使用している場合の DHCP クライアントの設定方法については、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 クライアントの設定方法

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. クライアントマシン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、システムの設定を解除し、システムを停止します。

```
# sys-unconfig
```

このコマンドを使用して設定情報を削除する方法については、sys-unconfig(1M) のマニュアルページを参照してください。

3. **PROM** プロンプトが表示されたら、システムをリブートします。

```
ok boot
```

システムのリブート時に、システム設定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。

4. **DHCP** を使用してネットワークインタフェースを設定するように促すプロンプトが表示されたら、Yes を選択します。

▼ Solaris DHCP クライアントの設定解除方法

1. クライアントマシン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、システムの設定を解除し、システムを停止します。

```
# sys-unconfig
```

このコマンドを使用して設定情報を削除する方法については、sys-unconfig(1M) のマニュアルページを参照してください。

3. **PROM** プロンプトが表示されたら、システムをリブートします。

```
ok boot
```

システムの設定を解除してあるので、システムのリブート時に、システムの設定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。

4. **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促すプロンプトが表示されたら、No を選択します。

第 11 章

DHCP の管理

この章では、サーバー、BOOTP リレーエージェント、クライアントに関する作業などの、Solaris DHCP サービスを管理する際に役立つ作業について説明します。それぞれの作業には、DHCP Manager を使用して作業を実行する手順と DHCP サービスユーティリティを使用して同様の作業を実行する手順が含まれています。DHCP サービスユーティリティについては、マニュアルページで詳細に説明されています。

DHCP サービスと初期ネットワークの初期設定を完了してからこの章を使用してください。第 10 章では、DHCP の設定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 Manager

DHCP Manager は、DHCP サービスで管理作業を実行するために使用する、Java ベースの GUI (Graphical User Interface) です。

DHCP Manager ウィンドウ

DHCP Manager 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設定が DHCP サーバーか BOOTP リレーエージェントかによって異なります。

サーバーの設定が DHCP サーバーになっている場合、DHCP Manager 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 Manager は次のタブを特徴としています。

- アドレス – DHCP が管理しているすべてのネットワークと IP アドレスを一覧表示する。「アドレス」タブから、ネットワークや IP アドレスを個別にもしくは、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 Manager を起動すると、「アドレス」タブで開かれる
- マクロ – DHCP 設定データベース (`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一覧表示する。「マクロ」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一覧表示する。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」、「ベンダー」、または「サイト」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

図 11–1 に、DHCP サーバー上で起動した場合の DHCP Manager ウィンドウを示します。

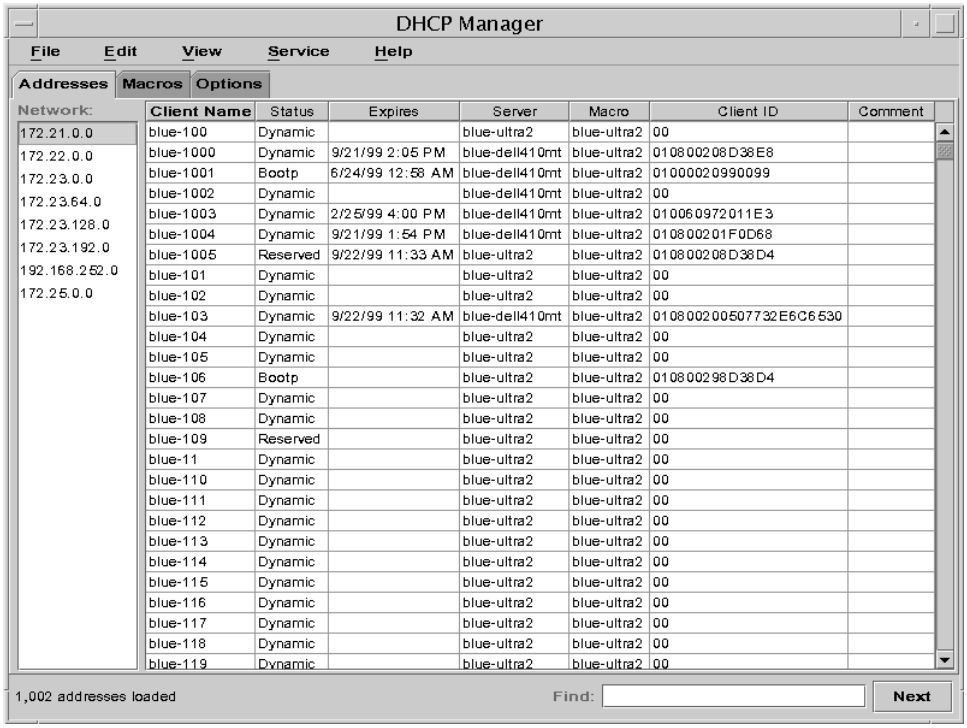


図 11–1　DHCP サーバーシステムで実行されている DHCP Manager

サーバーが BOOTP リレーエージェントとして設定されている場合、このリレーエージェントではタブに表示される情報は不要なため、DHCP Manager ウィンドウにはこれらのタブが表示されません。BOOTP リレーエージェントの属性を変更し、DHCP Manager 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。図 11–2 は、BOOTP リレーエージェントとして設定されたシステム上で起動した場合の DHCP Manager ウィンドウです。

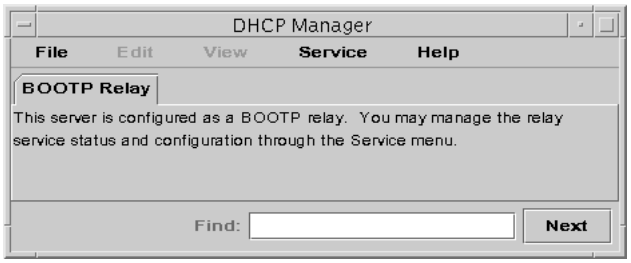


図 11–2　BOOTP リレーエージェントシステム上で実行されている DHCP Manager

DHCP Manager のメニュー

DHCP Manager のメニューには、次の内容が含まれます。

- 「ファイル」– DHCP Manager を終了する
- 「編集」– 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」– 現在選択されているタブの表示を変更する
- 「サービス」– DHCP デーモンを管理する
- 「ヘルプ」– Web ブラウザを開いて、DHCP Manager のヘルプを表示する

DHCP Manager が BOOTP リレーエージェントで実行されている場合、「編集」メニューと「表示」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」メニューと「サービス」メニューで実行されます。「編集」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、または変更ができます。また、「アドレス」タブが選択されている場合、「編集」メニューはウィザードも表示します。ウィザードは、ネットワークと複数の IP アドレスの作成を容易にするダイアログのセットです。「サービス」メニューは、DHCP デーモンを管理するコマンドを一覧表示し、開始と停止、有効と無効、サーバー設定の変更、サーバーの設定解除を可能にします。

DHCP Manager の起動と停止

スーパーユーザーとして DHCP サーバーで DHCP Manager を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して他の UNIX システムからリモートで DHCP Manager を表示させることができます。

▼ DHCP Manager を起動する方法

1. **DHCP** サーバーマシンでスーパーユーザーになります。
2. リモートで **DHCP** サーバーマシンにログインしている場合、次の手順に従ってローカルのマシンに **DHCP Manager** を表示させることができます。
   a. ローカルマシンで次のように入力します。

```
# xhost +server-name
```

b. リモートの **DHCP** サーバーマシン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 Manager ウィンドウが開いて、サーバーが DHCP サーバーに設定されている場合には「アドレス」タブを表示します。BOOTP リレーエージェントに設定されている場合には、タブは表示されません。

▼ DHCP Manager を停止する方法

- 「ファイル」メニューから「終了」を選択します。

DHCP サービスの起動と停止

DHCP サービスの起動と停止をするには、DHCP デーモンの動作に影響する可能性がある処理を実行する必要があります。望む結果を得るための適切な手順を選択するには、DHCP サービスの開始と停止、有効と無効、構成と構成解除の違いについて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。

- **DHCP** サービスの開始、停止、再起動 – 現在のセッションでのみデーモンの実行に影響します。たとえば、DHCP サービスを停止した場合、現在実行中のデーモンも終了しますが、システムを再起動すると終了したデーモンも再起動します。DHCP データテーブルはサービスの停止に影響されません。
- **DHCP** サービスの有効と無効 – 現在と将来のセッションでデーモンの実行に影響します。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** サービスの構成解除 – 現在実行中のデーモンを停止してシステムの再起動時に起動しないように設定し、ユーザーが DHCP データテーブルの削除を選択できるようにします。構成解除については、第 10 章で説明しています。

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、244ページの「DHCP サービス用のネットワークインタフェースの監視と無視」を参照してください。

この節では、DHCP サービスの起動と停止、有効または無効にする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 Manager)

1. **DHCP** サーバーマシン上でスーパーユーザーになります。

2. **DHCP Manager** を起動します。

   この手順については、226ページの「DHCP Manager を起動する方法」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」メニューから「起動」を選択して、**DHCP** サービスを起動します。

   b. 「サービス」メニューから「停止」を選択して、**DHCP** サービスを停止します。

      DHCP デーモンは、手動で再起動されるか、システムが再起動するまで停止します。

   c. 「サービス」メニューから「再開」を選択して、**DHCP** サービスを停止させ、すぐに再起動します。

▼ DHCP サービスを開始および停止する方法 (コマンド行)

1. サーバーマシン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
   ```

   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、システムが再開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 Manager)

1. **DHCP Manager** 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」メニューから「有効」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設定します。

   b. 「サービス」メニューから「無効」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設定します。

▼ DHCP サービスを無効にする方法 (コマンド行)

1. スーパーユーザーとして次のコマンドを入力し、`dhcpconfig` を起動します。

   ```
   # /usr/sbin/dhcpconfig
   ```

2. `3` と入力して「**Unconfigure DHCP or Relay Services**」を選択します。

3. 次のプロンプトに対して Y と入力して、**DHCP** を無効にします。

```
Unconfigure will stop the DHCP service and remove /etc/default/dhcp.
Are you SURE you want to disable the DHCP service? ([Y]/N): Y
```

▼ DHCP サービスを有効にする方法 (コマンド行)

この手順は、データを損なわずにサービスを無効にしたことがある場合のみ必要となります。

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. 次のコマンドを入力して dhcpconfig を起動します。

```
# /usr/sbin/dhcpconfig
```

3. 「**Configure DHCP Service**」または「**Configure BOOTP Relay Agent**」を、状況に応じて選択します。
4. Return キーを押して、すべてのプロンプトに対してデフォルト値を選択していくと、最後に次のプロンプトが表示されます。

```
Enable DHCP/BOOTP support of networks you select? ([Y]/N):
```

5. このプロンプトに対して Y と入力し、**DHCP** サービスを使用可能にします。
6. 次のプロンプトに対して次のように回答し、ネットワーク設定に関するプロンプトを回避します。

   データを損なわずにサービスを無効にしたことがある場合は、このネットワーク情報を設定し直す必要はありません。

```
###     Configure Local Networks        ###
Configure BOOTP/DHCP on local LAN network: 172.21.0.0? ([Y]/N):N
###     Configure Remote Networks       ###
Would you like to configure BOOTP/DHCP service on remote networks? ([Y]/
N):N
```

7. 次のプロンプトに対して Return キーを押して、**DHCP** サービスを再起動します。

```
Would you like to restart the DHCP service? (recommended) ([Y]/N):
```

DHCP サービスオプションの変更

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 Manager を使用した初期設定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設定した場合、これらのオプションの多くについて値の選択を要求されている場合があります。DHCP Manager の「サービスオプションの変更」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。

次の作業マップには、サービスオプションに関する作業と、使用する手順が示されています。

表 11–1 DHCP サービスオプション変更の作業マップ

作業	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能にし、DHCP トランザクションのログを使用可能または使用不能にし、syslog 機能を選択して DHCP トランザクションログに使用する	236ページの「詳細 DHCP ログメッセージを生成する方法 (DHCP Manager)」 236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)」 237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 Manager)」 237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)」 239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」
重複 IP アドレス検出の使用可能または使用不能	IP アドレスがまだ使用されていないことを確認してからそのアドレスをクライアントに提供する DHCP サーバーによる確認を使用可能または使用不能にする	241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 Manager)」 242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 サーバーの設定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 の読み込みを使用可能または使用不能にする。また、読み込み間隔を変更する	241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 Manager)」 242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)」

表 11–1 DHCP サービスオプション変更の作業マップ 続く

作業	説明	参照先
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	241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 Manager)」 242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)」
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	241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 Manager)」 242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)」

図 11–3 に、DHCP Manager 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 11–3　「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` の詳細については、`syslogd(1M)` と `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 デーモンに関するオプションを使用することで、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。236ページの「詳細 DHCP ログメッセージを生成する方法 (DHCP Manager)」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 サーバーの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
- EXTEND – 拡張機能のリース
- RELEASE – IP アドレスのリリース
- DECLINE – アドレス割り当てが減少するクライアント
- INFORM – ネットワーク設定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クライアント
- 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、提供の可能性がある 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能になっています。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。しかし、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なログファイルを指定した場合、他の通知とは別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能にできます。そして、237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 Manager)」で説明しているように、0 から 7 までの異なる `syslog` 機能を指定することができます。また、サーバーシステムの `syslog.conf` ファイルを編集する場合、239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 Manager)

1. 「サービス」メニューから「変更」を選択します。
2. 「詳細ログメッセージ」を選択します。
3. 「サーバーの再起動」をまだ選択していない場合は、選択します。
4. 「了解」をクリックします。

   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、デーモンの効率は削減される場合があります。

▼ 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が入ります。

デーモンは、このセッションについてだけ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、デーモンの効率が削減される場合があります。

▼ 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 Manager)

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」メニューから「変更」を選択します。
2. 「**syslog** へのログトランザクション」を選択します。

   トランザクションの記録を無効にするには、このオプションの選択を解除します。
3. 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` の設定内容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、239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使用可能にすると、メッセージファイルは急速に大きくなります。
4. 「サーバーの再起動」をまだ選択していない場合は、選択します。
5. 「了解」をクリックします。

▼ 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 -l syslog-local-facility
```

syslog-local-facility には、0 から 7 までの数字が入ります。このオプションを省略すると、デフォルトで 0 が入ります。237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 Manager)」を参照してください。

注 - トランザクションログを使用不能にするには、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 の設定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、239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使用可能にすると、メッセージファイルは急速に大きくなります。

▼ すべて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** サーバーシステムでスーパーユーザーになります。
2. 次のように入力して、dhcpconfig を起動します。

```
# /usr/sbin/dhcpconfig
```

3. 「**Configure DHCP Service**」を選択します。
4. 次のプロンプトに対して Return キーを押して、デフォルト値を承認します。デフォルト値は、ここで示す値と異なる場合があります。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N):
###     DHCP Service Configuration      ###
###     Configure DHCP Database Type and  Location      ###
Enter datastore (files or nisplus) [nisplus]:
Enter absolute path to datastore directory [dhcp.test.]:
Warning: Setting NIS_GROUP to admin.dhcp.test.
```

5. 次に示すプロンプトに対して `Y` と入力します。

```
###     Common daemon option setup      ###
Would you like to specify nondefault daemon options (Y/[N]):Y
Do you want to enable transaction logging? (Y/[N]):Y
```

6. 次のプロンプトに対して `0` から `7` の数字を入力します。

```
Which syslog local facility [0-7] do you wish to log to? [0]:
```

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` の設定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、239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、次のフォーマットを持つ行を追加します。

```
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 機能番号が入り、*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が入ります。

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、表 11–2 で説明します。

表 11–2 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
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。
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 アドレスを要求している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。この機能を使用不能にして、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。

表 11–2　DHCP サービスの性能に影響するオプション　続く

サーバーオプション	説明
指定された間隔での dhcptab の自動読み込み	指定した間隔で 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。ネットワークの設定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 Manager 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。
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あと、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期間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。

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 Manager)

1. 「サービス」メニューから「変更」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個数を変更するには、リレーエージェントホップの最大数を入力します。
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」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** を読みこむ周期」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あと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」フィールドに秒数を入力します。
6. 「サーバーの再起動」が選択されていない場合は、選択します。

7. 「了解」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)

この手順を使用してオプションを変更すると、現在のサーバーのセッションにだけ影響します。DHCP サーバーシステムを再起動すると、DHCP サーバーは、サーバー設定中に指定された設定を使用して起動します。以降のすべてのセッションに適用されるように設定を変更したい場合は、`dhcpconfig` を実行して、210ページの「DHCP サーバーの設定方法 (dhcpconfig)」に説明してあるように、プロンプトに対して入力する必要があります。

1. **DHCP サーバーシステムで上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能にする。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 クライアントを探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能にし、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

DHCP サーバーを設定する際に、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設定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。

この節では次の内容について説明します。

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作業手順を一覧表示します。

表 11–3 DHCP ネットワークを使用した作業マップ

作業	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能	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	246ページの「ネットワークインタフェースを無視するように DHCP を設定する方法」 246ページの「ネットワークインタフェースを監視するように DHCP を設定する方法」
DHCP サービスに新しいネットワークを追加	ネットワーク上で 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	248ページの「DHCP ネットワークを追加する方法 (DHCP Manager)」
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	249ページの「DHCP ネットワークの設定を変更する方法 (DHCP Manager)」 250ページの「DHCP ネットワークの設定を変更する方法 (コマンド行)」
DHCP サービスからのネットワークの削除	ネットワークを削除して、そのネットワーク上の IP アドレスを以降 DHCP に管理されないようにする	251ページの「DHCP ネットワークを削除する方法 (DHCP Manager)」 252ページの「DHCP ネットワークを削除する方法 (コマンド行)」

DHCP サービス用のネットワークインタフェースの監視と無視

デフォルトでは、`dhcpconfig` と DHCP Manager の構成ウィザードの両方によって DHCP サーバーが設定され、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が監視されます。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 Manager 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、DHCP Manager にある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。

`dhcpconfig` ユーティリティを使用してネットワークインタフェースを無視するように設定することはできません。

この節では、DHCP Manager 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して、ネットワークインタフェースを無視したり、新しいネットワークインタフェースを監視したりする手順について説明しています。このダイアログボックスを図 11–4 に示します。

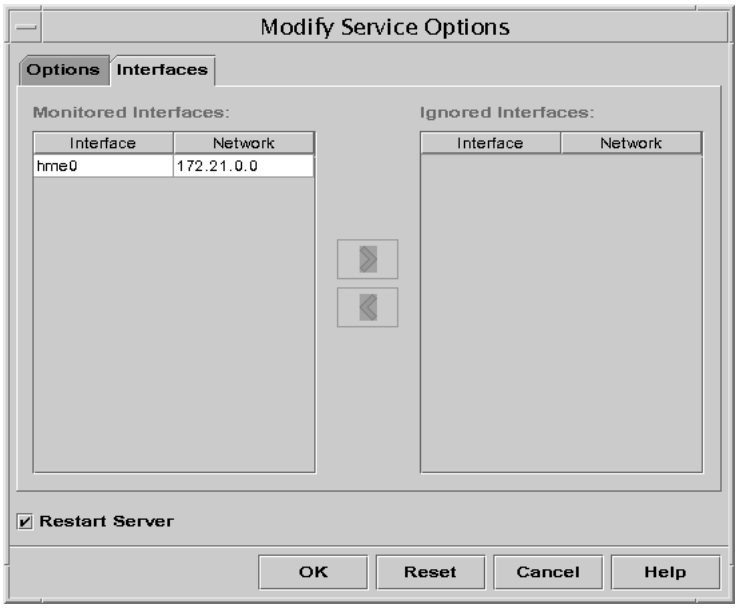


図 11–4　「サービスオプションの変更 (Modify Service Options)」ダイアログボックスの「インタフェース (Interfaces)」タブ

▼ ネットワークインタフェースを無視するように DHCP を設定する方法

1. 「サービス」メニューから「変更」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. 「インタフェースの監視」リストで、**DHCP** サービスを受けないネットワークインタフェースを選択します。

4. 右矢印ボタンをクリックして、そのネットワークインタフェースを「削除するインタフェース」リストに移動します。

5. 「了解」をクリックします。

▼ ネットワークインタフェースを監視するように DHCP を設定する方法

1. 「サービス」メニューから「変更」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. 「削除するインタフェース」リストで、**DHCP** サービスを受けるネットワークインタフェースを選択します。

4. 左矢印ボタンをクリックして、そのネットワークインタフェースを「インタフェースの監視」リストに移動します。

5. 「了解」をクリックします。

DHCP ネットワークの追加

初めに生成されるネットワークは通常、一次インタフェースにあるローカルなネットワークで、サーバーを設定する際に DHCP 構成ウィザードを使用して設定されます。この節では、DHCP Manager のネットワークウィザードを使用して DHCP の管理下に追加ネットワークを置く手順を説明します。

注 - コマンド行を使用したネットワークの追加の詳細については、214ページの「dhcpconfig の使用によるネットワークの構成」を参照してください。

図 11–5 に、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。

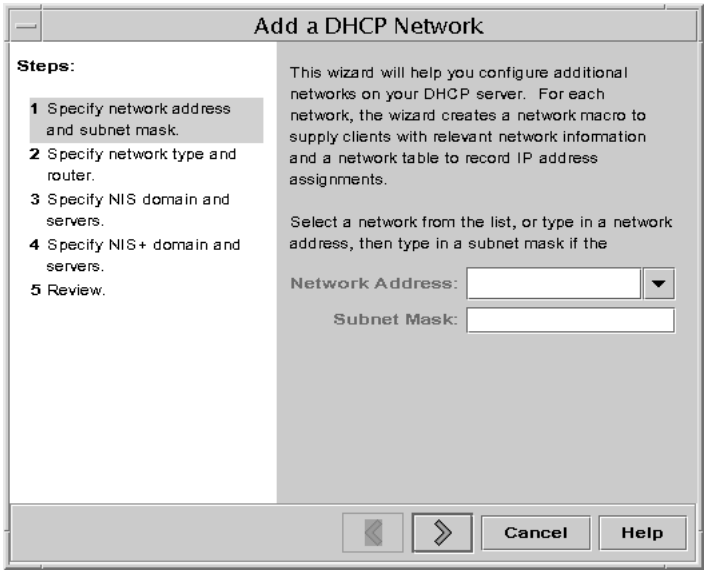


図 11–5 「DHCP Manager」のネットワークウィザード

新しいネットワークを設定すると、DHCP Manager が次の内容を作成します。

- データ記憶領域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 Manager の「アドレス」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。

コマンドを使用して新しいネットワークを追加するには、必要な標準 DHCP オプションまたはトークンのラベルについて理解している必要があります。これらの情

報はクライアントに渡されます。標準 DHCP オプションについては、dhcptab(4) のマニュアルページを参照してください。

▼ DHCP ネットワークを追加する方法 (DHCP Manager)

1. **DHCP Manager** の「アドレス」タブをクリックします。
2. 「編集」メニューから「ネットワークウィザード」を選択します。
3. 計画フェーズでの決定に基づいて、オプションまたはタイプを必要とする情報を選択します。

   計画については、197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。

   ウィザードを使用するのが難しい場合は、ウィザードウィンドウのヘルプをクリックして、ブラウザを開き、DHCP ネットワーク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たあと、「完了」をクリックしてネットワークの設定を終了します。

   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。「マクロ」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一覧表示されます。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。

DHCP ネットワークの設定の変更

ネットワークを DHCP サービスに追加すると、最初に入力した設定情報を変更できるのは、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する場合に限られます。

図 11–6 に、DHCP Manager の「マクロ (Macros)」タブを示します。

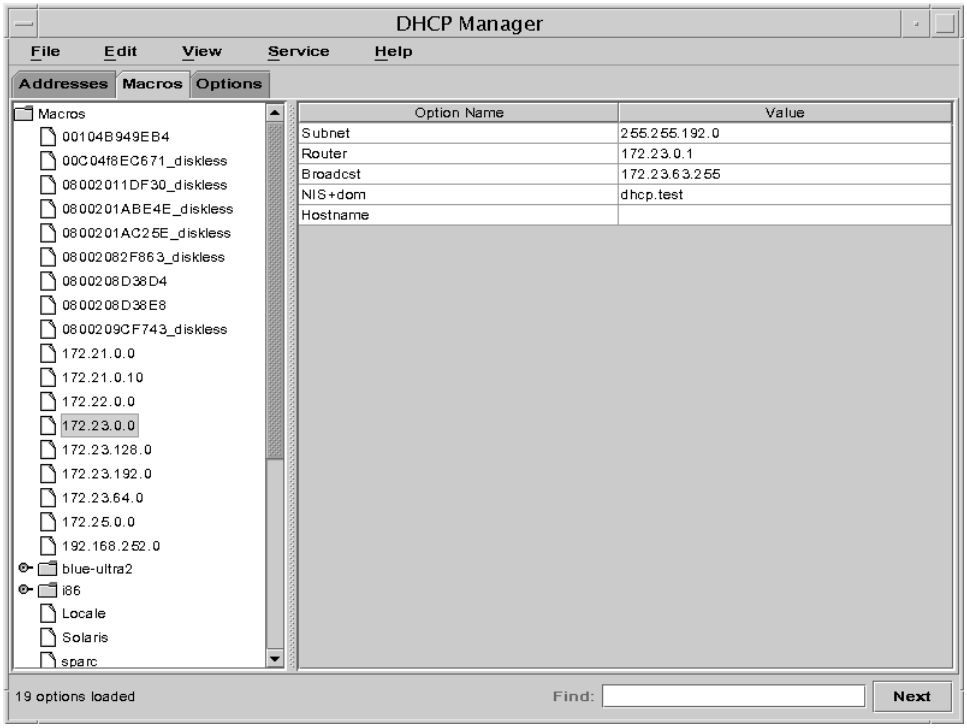


図 11–6 「DHCP Manager」 の「マクロ (Macros)」タブ

▼ DHCP ネットワークの設定を変更する方法 (DHCP Manager)

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一覧表示されます。

2. 設定を変更したいネットワークと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」メニューから「属性」を選択します。

   マクロ属性ダイアログボックスが開き、マクロに含まれるオプションの表が表示されます。

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」をクリックします。

   ダイアログボックスで選択をクリックして、ここでオプションを追加することもできます。マクロの変更の詳細については、278ページの「DHCP マクロの変更」を参照してください。

6. 「オプションの選択」を選択して、「了解」をクリックします。

   この変更は dhcptab に加えられます。DHCP サーバーは dhcptab を再読み込みするように信号を受け、この変更を有効にします。

▼ DHCP ネットワークの設定を変更する方法 (コマンド行)

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判定します。

   ネットワークマクロの名前は、ネットワークの IP アドレスと一致している必要があります。

   この情報が含まれているマクロがわからない場合、dhcptab データベースを表示させて、dhtadm -P コマンドを使い、すべてのマクロを一覧表示させることができます。

2. 次のフォーマット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、172.25.62.0 のマクロのリース期間を 57600 秒に変更し、NIS ドメインを sem.west.com に変更するには、次のように入力します。

```
# dhtadm -M -m 172.25.62.0 -e 'LeaseTim=57600'
# dhtadm -M -m 172.25.62.0 -e 'NISdmain=sem.west.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 Manager 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。図 11–7 は、DHCP Manager の「ネットワークの削除 (Delete Networks)」ダイアログボックスを示しています。

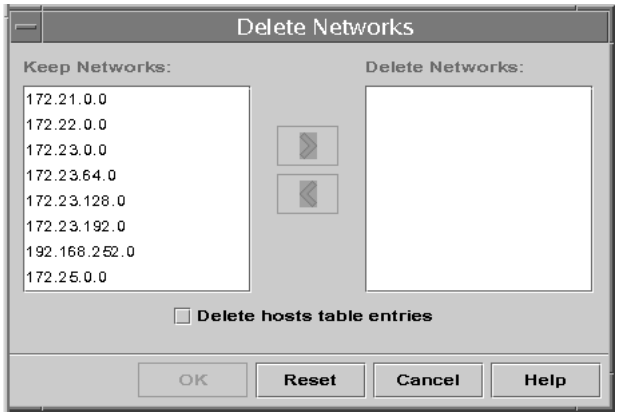


図 11–7　「ネットワークの削除 (Delete Networks)」ダイアログボックス

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 Manager)

1. 「アドレス」タブを選択します。
2. 「編集」メニューから「ネットワークの削除」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」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」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」を選択します。

6. 「了解」をクリックします。

▼ DHCP ネットワークを削除する方法 (コマンド行)

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することに注意してください。この手順によって、hosts ファイルからホスト名が確実に削除されます。

1. サーバーシステムでスーパーユーザーになります。

2. 次の形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 -yIP-address
```

   たとえば、アドレス 172.25.52.1 を削除するには、次のように入力します。

```
# pntadm -D -y 172.25.52.1
```

   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、次のように入力して、**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 172.25.52.0 を削除するには、次のように入力します。

```
# pntadm -R 172.25.52.0
```

DHCP サービスを使用した BOOTP クライアントのサポート

DHCP サーバーで 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 サーバーのデータベースに登録したり、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。

次に示す方法のどちらかを使用して、BOOTP クライアントのサポートを設定することができます。

- 自動 **BOOTP** サポート – DHCP が管理するネットワークや、BOOTP リレーエージェントによって DHCP が管理するネットワークに接続されたネットワーク上の BOOTP クライアントはすべて、サーバーから IP アドレスを取得することができます。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ります。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、特に役立ちます。
- 手動 **BOOTP** サポート – DHCP サービスを使用して手動で登録された BOOTP クライアントだけが、サーバーからの応答を受け取ります。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ります。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です。

注 - BOOTP アドレスは常時割り当てられます。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、その作業手順を一覧表示します。

表 11–4 BOOTP サポート作業マップ

作業	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	254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 Manager)」 256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 (コマンド行)」
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	255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 Manager)」 259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 (コマンド行)」

▼ すべての BOOTP クライアントのサポートを設定する方法 (DHCP Manager)

1. 「サービス」メニューから「変更」を選択します。
   サービスオプション変更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」セクションで、「自動」を選択します。
3. 「サーバーの再起動」が選択されていない場合は、選択します。
4. 「了解」をクリックします。
5. **DHCP Manager** の「アドレス」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」メニューから「属性」を選択します。
   「複数アドレスの変更」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」を選択します。
   残りのオプションは「現在の設定を維持」に設定しておきます。

9. 「了解」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 Manager)

1. 「サービス」メニューから「変更」を選択します。
   「サービスオプションの変更」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」セクションで、「手動」を選択します。

3. 「サーバーの再起動」が選択されていない場合は、選択します。

4. 「了解」をクリックします。

5. **DHCP Manager** の「アドレス」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」メニューから「属性」を選択します。
   「アドレスの属性」ダイアログボックスが開きます。

8. 「リース」タブを選択します。

9. 「クライアント **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 操作環境を実行している BOOTP クライアントの ID は、Ethernet の ARP タイプ (01) にそのクライアントの 16 進 Ethernet アドレスから導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。ARP タイプについては、表 13–4 を参照してください。

ヒント - クライアントマシンのスーパーユーザーになって `ifconfig -a` を入力し、そのインタフェース用の Ethernet アドレスを取得してください。

---

10. 「予約」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」を選択します。

12. 「了解」をクリックします。

    「アドレス」タブでは、BOOTP は「状態」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」フィールドに一覧表示されます。

▼ すべての BOOTP クライアントのサポートを設定する方法 (コマンド行)

1. スーパーユーザーとしてシステムにログインするか、スーパーユーザーになって、次のように入力します。

```
# /usr/sbin/dhcpconfig
```

   テキスト形式の DHCP 設定メニューが表示されます。

2. `1` を入力して `Return` キーを押し、「**Configure DHCP Service**」を選択します。

3. 次のプロンプトに対して表示されているように入力して、**BOOTP** 互換オプションをスキップします。

   何も入力しない場合は、`Return` キーを押すとデフォルトの設定になります。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N)
Enter datastore (files or nisplus) [files]:
Enter absolute path to datastore directory [/var/dhcp]:
Would you like to specify nondefault daemon options (Y/[N]):
Would you like to specify nondefault server options (Y/[N]):Y
How long (in seconds) should the DHCP server keep outstanding OFFERs? [10]:
How often (in minutes) should the DHCP server rescan the dhcptab? [Never]:
```

4. プロンプトに対して次のように回答して、**BOOTP** の互換性を使用可能にします。

```
Do you want to enable BOOTP compatibility mode? (Y/[N]):Y
Do you want the server to allocate IP addresses to new BOOTP clients? ([Y]/N)
```

5. 次のプロンプトに対して表示されているように回答して、アドレス作成のプロンプトに進みます。

```
###     Initialize dhcptab table        ###
The dhcptab table already exists.
Do you want to merge initialization data with the existing table? (Y/[N]):
###     Select Networks For BOOTP/DHCP Support  ###
Enable DHCP/BOOTP support of networks you select? ([Y]/N):
```

ローカルネットワーク上に BOOTP アドレスを作成する場合は、次の手順に進みます。

リモートネットワーク上に BOOTP アドレスを作成する場合は、258ページの手順 7 に進みます。

6. 次のプロンプトに対して表示されているように回答して、ローカルネットワーク上に **BOOTP** アドレスを作成します。

この手順が、ネットワーク 172.21.0.0 に 4 つのアドレスを追加する例であることに注意してください。実際の手順では各ネットワークに合う回答に置き換える必要があります。

```
###     Configure Local Networks        ###
Configure BOOTP/DHCP on local LAN network: 172.21.0.0? ([Y]/N):
Do you want hostnames generated and inserted in the nisplus hosts table? (Y/[N]):
Enter starting IP address [172.21.0.0]: 172.21.0.15
Enter the number of clients you want to add (x < 65535): 4
BOOTP compatibility with automatic allocation is enabled.
Do you want any of your 4 addresses to be BOOTP specific? ([Y]/N):
How many (x <= 4): 4
The dhcp network table: 172.21.0.0 already exists.
Do you want to add entries to it? ([Y]/N):
dhcptab macro "172.21.0.0" already exists.
Do you want to merge initialization data with the existing macro? ([Y]/N):N
Disable (ping) verification of 172.21.0.0 address(es)? (Y/[N]):
/ 75% Complete.
Configured 4 entries for network 172.21.0.0.
```

7. リモートネットワーク上に **BOOTP** アドレスを作成する場合、プロンプトに対して次のように回答します。

   この手順が、クライアントが LAN 接続経由でアクセスするネットワーク 172.23.0.0 に 4 つのアドレスを追加する例であることに注意してください。実際の手順では、ネットワークに合った回答に置き換える必要があります。

```
###     Configure Remote Networks       ###
Would you like to configure BOOTP/DHCP service on remote networks? ([Y]/N):
Enter Network Address of remote network, or <RETURN> if finished: 172.23.0.0
Do clients access this remote network via LAN or PPP connection? ([L]/P):
Do you want hostnames generated and inserted in the nisplus hosts table? (Y/[N]):
Enter Router (From client's perspective), or <RETURN> if finished.
IP address:
Optional: Enter Remote Network's MTU (e.g. ethernet == 1500):
Enter starting IP address [172.23.0.0]: 172.23.0.10
Enter the number of clients you want to add (x < 65535): 4
BOOTP compatibility with automatic allocation is enabled.
Do you want any of your 4 addresses to be BOOTP specific? ([Y]/N):
How many (x <= 4): 4
The dhcp network table: 172.23.0.0 already exists.
Do you want to add entries to it? ([Y]/N):
dhcptab macro "172.23.0.0" already exists.
Do you want to merge initialization data with the existing macro? ([Y]/N):N
Disable (ping) verification of 172.23.0.0 address(es)? (Y/[N]):
/ 75% Complete.
Configured 4 entries for network 172.23.0.0.
Enter Network Address of remote network, or <RETURN> if finished:
```

8. 4 と入力して Return キーを押し、dhcpconfig を終了します。

▼ 登録された BOOTP クライアントのサポートを設定する方法 (コマンド行)

1. スーパーユーザーとしてシステムにログインするか、スーパーユーザーになり、次のように入力します。

```
# /usr/sbin/dhcpconfig
```

テキスト形式の DHCP 設定メニューが表示されます。

2. 1 を入力して Return キーを押し、「**Configure DHCP Service**」を選択します。

3. 次のプロンプトに対して表示されているように回答し、**BOOTP** 互換オプションを省略します。

何も入力しない場合は、Return キーを押すとデフォルトの設定になります。

```
Would you like to stop the DHCP service? (recommended) ([Y]/N)Y
Enter datastore (files or nisplus) [files]:
Enter absolute path to datastore directory [/var/dhcp]:
Would you like to specify nondefault daemon options (Y/[N]):
Would you like to specify nondefault server options (Y/[N]):Y
How long (in seconds) should the DHCP server keep outstanding OFFERs? [10]:
How often (in minutes) should the DHCP server rescan the dhcptab? [Never]:
```

4. **BOOTP** プロンプトに対して次のように回答します。

```
Do you want to enable BOOTP compatibility mode? (Y/[N]):Y
Do you want the server to allocate IP addresses to new BOOTP clients? ([Y]/N):N
```

N と入力すると、登録されていない BOOTP クライアントで IP アドレスを取得できなくなります。これは、DHCP Manager の「手動」オプションと類似しています。

```
The dhcptab table already exists.
Do you want to merge initialization data with the existing table? (Y/[N]):N
Enable DHCP/BOOTP support of networks you select? ([Y]/N):N
```

N と入力すると、DHCP サービスにネットワークを追加するためのプロンプトが表示されません。

```
Would you like to restart the DHCP service? (recommended) ([Y]/N):Y
```

5. 4 と入力して Return キーを押し、dhcpconfig を終了します。

6. 次のどちらかのフォーマットを使用してコマンドを入力し、アドレスを変更または追加して特定の **BOOTP** クライアント用に予約します。

   a. **BOOTP** 用の既存アドレスを変更するには次のように入力します。

```
# pntadm -M ip-address -i client-id -f BOOTP -e -l -m macro-name network-ip-address
```

たとえば、アドレス 172.21.20.33 を変更するには、Ethernet ハードウェアアドレスが 8:0:20:89:a1:d2 のクライアントにそのアドレスを割り当てて BOOTP フラグを設定し、次のように入力します。

```
# pntadm -M 172.21.20.33 -i 0108002089A1D2 -f BOOTP
```

   b. 新しい **BOOTP** アドレスを追加するには、次のように入力します。

```
# pntadm -A ip-address -i client-id -f BOOTP -m macro-name network-ip-address
```

たとえば、アドレス 172.21.20.34 を追加するには、Ethernet ハードウェアアドレスが 8:0:20:89:a1:d2 のクライアントにそのアドレスを割り当てて BOOTP フラグを設定し、そのクライアントが blue2 マクロの内容を受け取るようにしてから、次のように入力します。

```
pntadm -A 172.21.20.34 -i 0108002089A1D2 -f BOOTP -m blue2 172.21.0.0
```

ネットワークの各 BOOTP クライアントに対して 1 つの BOOTP アドレスを予約する必要があります。

DHCP サービスで IP アドレスを使用して作業する

DHCP Manager 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することができます。IP アドレスを使用した作業を始める前に、IP アドレスの属性を知っておくため、表 11–6 を参照してください。この表を使用して、DHCP Manager と pntadm のユーザー向け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 11–6 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 の詳細については、pntadm(1M) 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、および削除する際に実行する必要がある作業と、その手順を一覧表示します。

表 11–5 DHCP における IP アドレスの作業マップ

作業	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 Manager 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	267ページの「単一の IP アドレスを作成する方法 (DHCP Manager)」
		267ページの「既存の IP アドレスを複製する方法 (DHCP Manager)」
		268ページの「複数のアドレスを作成する方法 (DHCP Manager)」
IP アドレスの属性を変更する	表 11–6 で説明している IP アドレスの属性を変更する	230ページの「DHCP サービスを有効にする方法 (コマンド行)」

表 11–5 DHCP における IP アドレスの作業マップ 続く

作業	説明	参照先
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	271ページの「アドレスを使用不可に指定する方法 (DHCP Manager)」 272ページの「DHCP サービスから IP アドレスを削除する方法 (DHCP Manager)」
一貫したアドレスを 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	274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 Manager)」

表 11–6 で、IP アドレスの属性を一覧表示して説明します。

表 11–6 IP アドレスの属性

属性	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 Manager 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 172.21.0.0 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* 172.21.0.0**
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 Manager 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 オプションが付随する たとえば、IP アドレス 172.21.5.12 を変更するには次のように入力する **pntadm -M 172.21.5.12 *options* 172.21.0.0**

表 11–6 IP アドレスの属性 続く

属性	説明	pntadm コマンドで指定する方法
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たときに、DHCP Manager または dhcpconfig によって自動的に生成される。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、172.21.5.12 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 172.21.5.12 -h carrot12 172.21.0.0**
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 172.21.5.12 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 172.21.5.12 -s blue2 172.21.0.0**
設定マクロ	DHCP サーバーが dhcptab データベースからネットワーク設定オプションを取得するために使用するマクロ。サーバーを設定して、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。マクロの詳細については、178ページの「マクロについて」を参照のこと。サーバーマクロはデフォルトでは、DHCP Manager または dhcpconfig がアドレスを作成したときに選択される。	–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 172.21.5.12 に割り当てるには、次のように入力する **pntadm -M 172.21.5.12 -m blue2 172.21.0.0**

表 11–6 IP アドレスの属性 続く

属性	説明	pntadm コマンドで指定する方法
クライアント ID	クライアントの 16 進ハードウェアアドレスから導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 ハードウェアコードのすべてのリストについては、表 13–4 を参照 たとえば、16 進 Ethernet アドレス 8:0:20:94:12:1e を持つクライアントは、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 Manager と pntadm に一覧表示される。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、そのアドレスとクライアントを手動で結び付ける ヒント : クライアントマシンのスーパーユーザーとして、ifconfig -a を入力すると、そのインタフェースに関する Ethernet アドレスを取得することができる	–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 172.21.5.12 に割り当てるには、次のように入力する **pntadm -M 172.21.5.12 -i 0108002094121E 172.21.0.0**
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設定。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 172.21.5.12 の予約を指定するには、次のように入力する **pntadm -M 172.21.5.12 -f MANUAL 172.21.0.0**

表 11–6 IP アドレスの属性 続く

属性	説明	pntadm コマンドで指定する方法
リースタイプ	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設定。リースは、動的または固定。詳細については、195ページの「動的および永続的リースタイプ」を参照のこと	–f オプションを使用して、アドレスが固定で割り当てら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 172.21.5.12 を永続的リースに指定するには、次のように入力する **pntadm -M 172.21.5.12 -f PERMANENT 172.21.0.0**
リース有効期限	リースが切れる日付。動的リースが指定された場合のみ利用できる。この日付は、*mm/dd/yyyy* のフォーマットで指定され、DHCP サーバーによって計算される。	–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 2000 年 1 月 1 日に指定するには、次のように入力する **pntadm -M 172.21.5.12 -e 01/01/2000 172.21.0.0**
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設定。BOOTP クライアントのサポートに関する詳細については、253ページの「DHCP サービスを使用した BOOTP クライアントのサポート」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 172.21.5.12 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 172.21.5.12 -f BOOTP 172.21.0.0**
使用不能設定	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、アドレスを使用不能に指定する たとえば、IP アドレス 172.21.5.12 を使用不能に指定するには、次のように入力する **pntadm -M 172.21.5.12 -f UNUSABLE 172.21.0.0**

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の詳細については、247ページの「DHCP ネットワークの追加」を参照してください。

DHCP Manager または `dhcpconfig` を使用してアドレスを追加することができます。コマンドを使用してアドレスを追加したい場合は、214ページの「`dhcpconfig` の使用によるネットワークの構成」の説明に従って `dhcpconfig` を使用します。

DHCP Manager を使用した複数の方法で、すでに DHCP サービスで管理されているネットワークにアドレスを追加することができます。

- 単一の **IP** アドレスの作成 – 1 つの新しい 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** アドレスの作成 – 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

図 11–8 は、「アドレスの作成 (Create Address)」ダイアログボックスを示しています。「アドレスの複製」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一です。

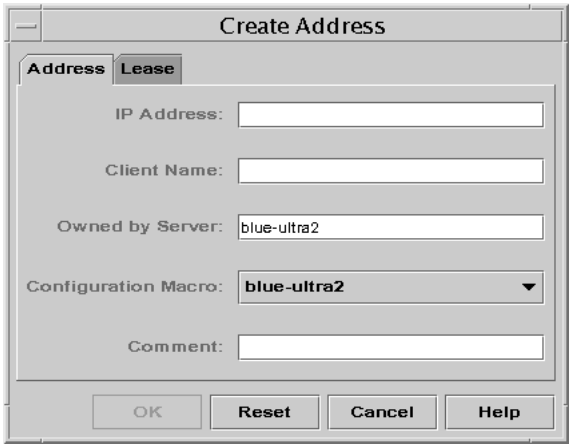


図 11–8　「アドレスの作成 (Create Address)」ダイアログボックス

図 11–9 は、一定範囲の IP アドレスを作成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ています。

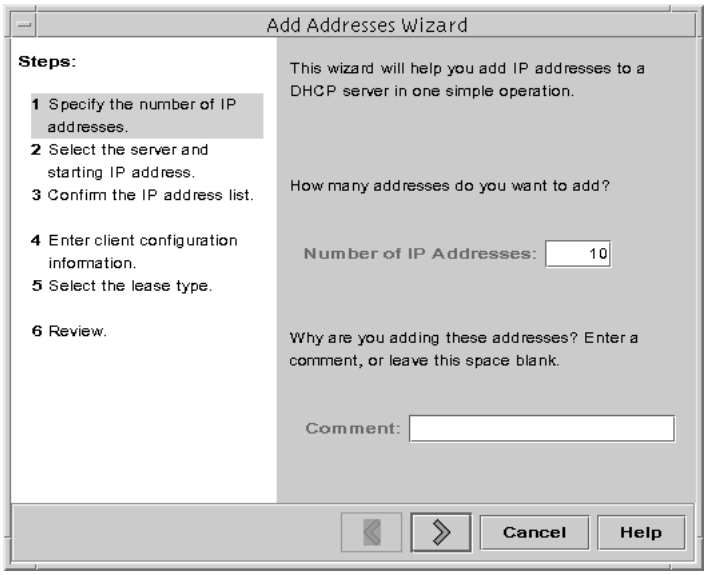


図 11–9　アドレスウィザード

▼ 単一の IP アドレスを作成する方法 (DHCP Manager)

1. 「アドレス **(Address)**」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」メニューから「作成」を選択します。
   「アドレスの作成」ダイアログボックスが開きます。
4. 「アドレス」と「リース」タブで、値を選択または入力します。
   設定の詳細については、表 11–6 を参照してください。
5. 「了解」をクリックします。

▼ 既存の IP アドレスを複製する方法 (DHCP Manager)

1. 「アドレス **(Address)**」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」メニューから「複製」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」をクリックします。

▼ 複数のアドレスを作成する方法 (DHCP Manager)

1. 「アドレス **(Address)**」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」メニューから「アドレスウィザード」を選択します。

   アドレスウィザードが起動し、IP アドレスの属性の値を入力するよう求めるダイアログを表示します。これらの属性の詳細については、表 11–6 を参照してください。193ページの「IP アドレス管理のための決定事項」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」をクリックします。

   「アドレス (Address)」 タブに新規アドレスが更新されます。

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 Manager または `pntadm -M` コマンドを使用して、表 11–6 で説明している属性を変更することができます。`pntadm -M` の使用に関する詳細については、`pntadm(1M)` のマニュアルページを参照してください。

図 11–10 は、IP アドレスの属性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を示します。

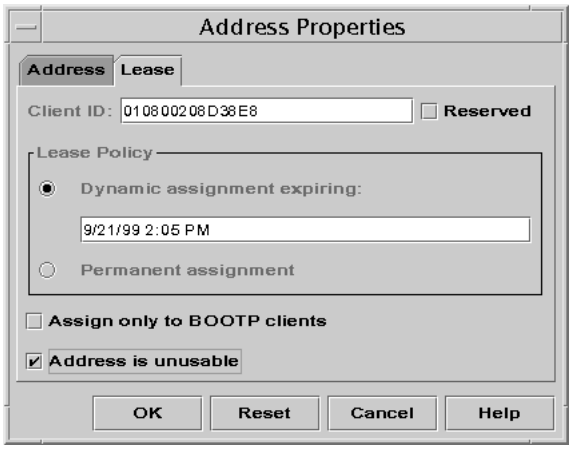


図 11–10　「アドレスの属性 (Address Properties)」ダイアログボックス

図 11–11 は、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。

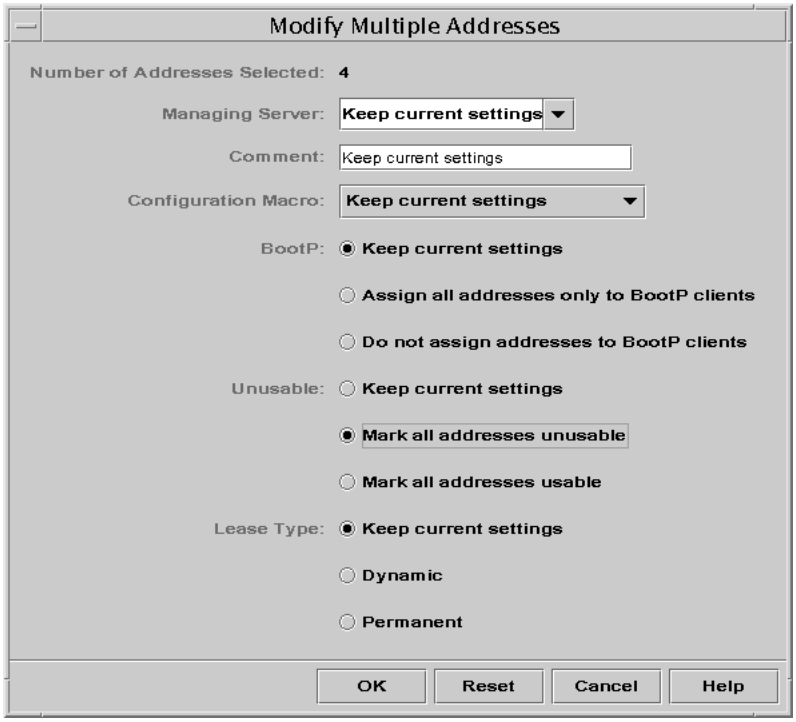


図 11–11　「複数アドレスの変更 (Modify Multiple Addresses)」ダイアログボックス

▼ IP アドレスの属性を変更する方法 (DHCP Manager)

1. 「アドレス **(Address)**」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」メニューから「属性」を選択します。

   「アドレスの変更」ダイアログボックスまたは「複数アドレスの変更」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。

   属性の詳細については、表 11–6 を参照してください。
6. 「了解 **(OK)**」をクリックします。

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決まります。

- 一時的にアドレスを使用不能にするには、270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能に指定できます。
- 永続的に DHCP クライアントがアドレスを使用不能にするには、271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` オプションを付けた `pntadm -M` コマンドを使用して、コマンド行からアドレスを使用不能に指定することができます。

DHCP Manager では、次の手順に示すとおり、図 11–10 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が指定でき、図 11–11 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が指定できます。

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 Manager の「アドレスの削除 (Delete Address)」ダイアログボックスが使用できます。

図 11–12 は、「アドレスの削除 (Delete Address)」ダイアログボックスを示します。

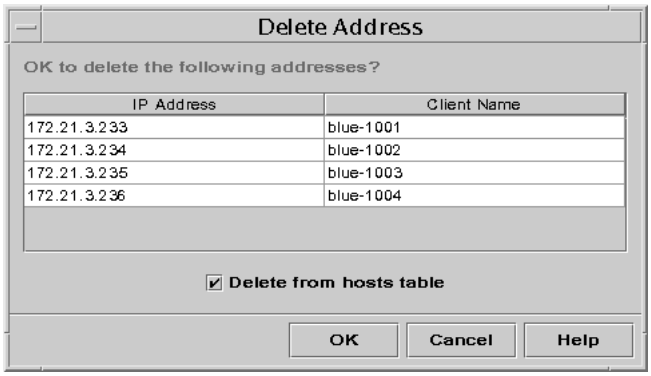


図 11–12　「アドレスの削除 (Delete Address)」ダイアログボックス

▼ アドレスを使用不可に指定する方法 (DHCP Manager)

1. 「アドレス」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. 使用不能に指定したい **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能に指定する場合は、`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。また、`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」メニューから「属性」を選択します。

「アドレスの変更」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」タブを選択します。

6. 「アドレスを使用しない」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」を選択します。

7. 「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 (DHCP Manager)

1. 「アドレス」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」メニューから「削除」を選択します。

   削除する内容を確認できる「アドレスの削除」ダイアログボックスが開いて、選択したアドレスが一覧表示されます。

5. **DHCP Manager** または `dhcpconfig` によって生成されたホスト名について、その名前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」を選択します。

6. 「了解」をクリックします。

一定の 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。しかし、動的リースを使用している場合は除きます。

ネットワークの機能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 アドレスを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設定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、手動で割り当てることによって、クライアントがその設定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したりできます。予約されているアドレスは動的リースに設定してアドレスの使用を追跡しやすくしたり使用追跡が不要な場合に永続的リースに設定したりできます。しかし、固定リリースはお奨めしません。いったん永続的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らず、更新された設定情報を取得できなくなるためです。

`pntadm -M` コマンドまたは DHCP Manager 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。

図 11–13 は、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)」タブを示しています。

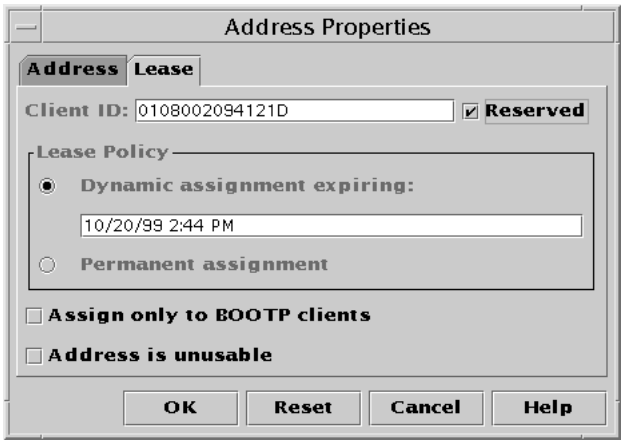


図 11–13　「アドレスの属性 (Address Properties)」の「リース (Lease)」タブ

▼ 固定 IP アドレスを DHCP クライアントに割り当てる方法 (DHCP Manager)

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 の決め方の詳細については、表 11–6 のクライアント ID の項目を参照してください。

2. **DHCP Manager** の「アドレス」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**」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。

7. 「予約」オプションを選択して、その **IP** アドレスがサーバーから返還を要請されないようにします。

8. 「アドレスの属性」ウィンドウの「リースポリシー」領域で、「動的」または「常時」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」を選択します。「予約」を選択したので、動的リースを使用した場合でもこのアドレスは返還を要請されることはありません。このリースについて有効期限を入力する必要はありません。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」の選択はお奨めしません。トランザクションログを使用可能にしていない限り、IP アドレスの使用を追跡できないためです。

DHCP マクロを使用した作業

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、クライアントに伝える必要があるオプションをまとめます。サーバーが設定されると、DHCP Manager 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、178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の詳細については、第 10 章を参照してください。

ネットワークに変更が生じると、クライアントに伝える設定情報を変更しなければならない場合があります。そのためには、マクロの追加、変更、複製、または削除といった、マクロを使用した作業をする必要があります。

マクロを使用した作業には、DHCP 標準オプションの知識が必要です。この知識は、`dhcptab(4)` のマニュアルページで説明されています。

次の作業マップでは、DHCP マクロの表示、変更、追加、および削除に関する作業が一覧表示されています。

表 11–7 DHCP マクロの作業マップ

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	277ページの「DHCP サーバーで定義されたマクロを表示する方法 (DHCP Manager)」
DHCP マクロの追加	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	282ページの「DHCP マクロを追加する方法 (DHCP Manager)」

表 11–7 DHCP マクロの作業マップ 続く

作業	説明	参照先
DHCP クライアントに伝えら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	279ページの「DHCP マクロ内のオプションに関する値を変更する方法 (DHCP Manager)」 279ページの「DHCP マクロにオプションを追加する方法 (DHCP Manager)」 280ページの「DHCP マクロからオプションを削除する方法 (DHCP Manager)」
DHCP マクロの削除	使用しない DHCP マクロを削除する	284ページの「DHCP マクロを削除する方法 (DHCP Manager)」

図 11–14 は、「DHCP Manager」ウィンドウの「マクロ (Macro)」タブを示します。

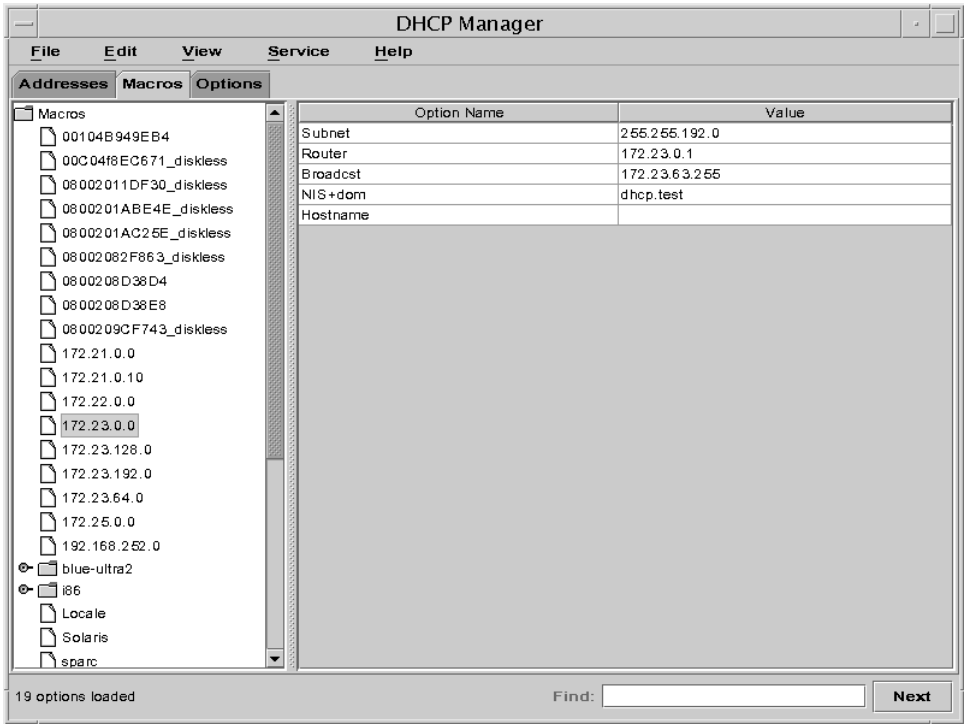


図 11–14　「DHCP Manager」の「マクロ (Macros)」タブ

▼ DHCP サーバーで定義されたマクロを表示する方法 (DHCP Manager)

DHCP Manager または `dhtadm -P` を使用して、DHCP サーバーで定義されたすべてのマクロを表示させることができます。

1. 「マクロ **(Macros)**」タブを選択します。

   ウィンドウ左側の「マクロ」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。頭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。頭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一覧表示されます。

3. マクロの内容を表示させるには、マクロ名をクリックして、ウィンドウの右側の領域を見ます。

   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 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。標準的な DHCP オプションは、DHCP Manager のヘルプと `dhcptab(4)` のマニュアルページに一覧表示されています。

`dhtadm -M -m` コマンドまたは DHCP Manager を使用して、マクロを変更することができます。`dhtadm` の詳細については、`dhtadm(1M)` のマニュアルページを参照してください。

図 11–15 は、DHCP Manager の「マクロの属性 (Macro Properties)」ダイアログボックスを示します。

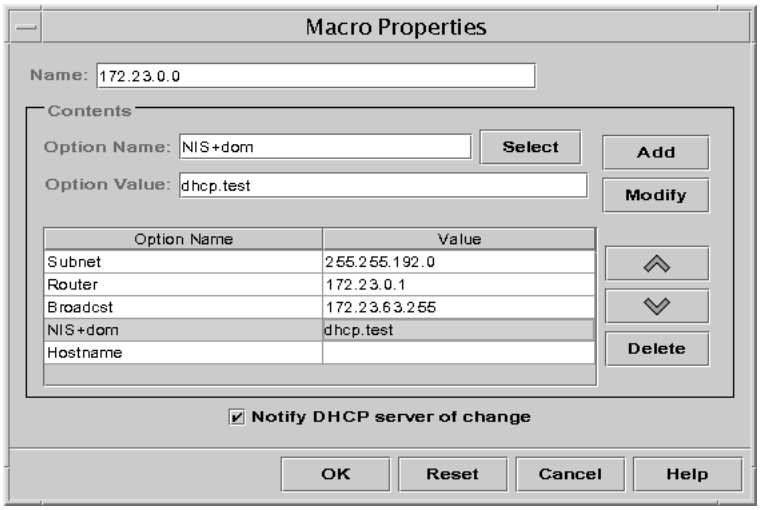


図 11–15　「マクロの属性 (Macro Properties)」ダイアログボックス

▼ DHCP マクロ内のオプションに関する値を変更する方法 (DHCP Manager)

1. 「マクロ」タブを選択します。
2. 変更したいマクロを選択します。
3. 「編集」メニューから「属性」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」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」と「オプションの値」のフィールドに表示されます。
5. 「オプションの値」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」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」を選択します。
   この選択によって、DHCP サーバーに `dhcptab` の再読み込みの指示が伝わり、「了解」をクリックすると直ちに変更が実現されます。
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 (DHCP Manager)

1. 「マクロ」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」メニューから「属性」を選択します。
   「マクロの属性」ダイアログボックスが開きます。

4. 「オプション名」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」フィールドの隣にある「選択」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」ダイアログボックスに、「標準」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一覧表示されます。「標準」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」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。

      マクロカテゴリの詳細については、178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」フィールドにオプションの値を入力してください。

   オプション名を **`Include`** と入力した場合は、「オプションの値」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」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したい場合は、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」を選択します。

   この選択によって、DHCP サーバーに `dhcptab` の再読み込みの指示が伝わり、「了解」をクリックすると直ちに変更が実現されます。

8. 「了解」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 (DHCP Manager)

1. 「マクロ」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」メニューから「属性」を選択します。
   「マクロの属性」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」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」を選択します。
   この選択によって、DHCP サーバーに `dhcptab` の再読み込みの指示が伝わり、「了解」をクリックすると直ちに変更が実現されます。

7. 「了解」をクリックします。

DHCP マクロの追加

DHCP サービスに新しいマクロを追加して、特定のニーズ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 Manager の「マクロの作成」ダイアログボックスを使用して、マクロが追加できます。`dhtadm` の詳細については、`dhtadm(1M)` のマニュアルページを参照してください。

図 11–16 は、DHCP Manager の「マクロの作成 (Create Macro)」ダイアログボックスを示しています。

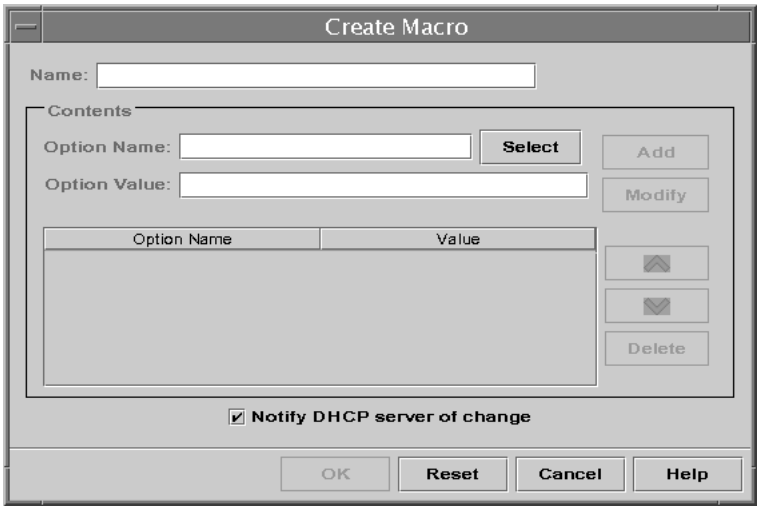


図 11–16　「マクロの作成 (Create Macro)」ダイアログボックス

▼ DHCP マクロを追加する方法 (DHCP Manager)

1. 「マクロ」タブを選択します。

2. 「編集」メニューから「作成」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. そのマクロに固有の名前を入力します。

   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、178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」フィールドの隣にある「選択」ボタンをクリックします。

   「オプションの選択」ダイアログボックスが開き、「標準」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一覧表示されます。

5. 「標準」カテゴリに含まれていないオプションを追加したい場合は、「カテゴリ」リストを使用して、カテゴリを選択することができます。

オプションカテゴリの詳細については、177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」をクリックします。

   「マクロの属性」ダイアログボックスが、「オプション名」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」フィールドにオプションの値を入力します。

8. 「追加」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、283ページの手順 6 から 283ページの手順 8 までを繰り返します。

   オプションの順序を再度変更したい場合は、オプション名を選択して、矢印キーをクリックし、オプションリスト内でその名前を上下に移動させます。

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」を選択します。

    この選択によって、DHCP サーバーに `dhcptab` の再読み込み指示が伝わり、「了解」をクリックすると直ちに変更が実現されます。

11. 「了解 **(OK)**」をクリックします。

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と、関連するサーバーマクロも使用されなくなるため削除することができます。

`dhtadm -D -m` コマンドまたは DHCP Manager 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 Manager)

1. 「マクロ」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」ダイアログボックスが開き、指定したマクロの削除を確認を求められます。
3. 「**DHCP** サーバーに変更を通知」を選択します。
4. 「了解」をクリックします。

DHCP オプションの使用

オプションは、DHCP サーバーがクライアントに伝えるネットワーク設定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 Manager の「オプション」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。Solaris DHCP 以外のクライアントを使用している場合は、新しいオプションまたはシンボルの追加に関する情報について、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 でのオプションの詳細については、177ページの「オプションについて」を参照してください。

DHCP Manager または `dhtadm` コマンドを使用して、オプションを作成したり変更したりまたは削除したりできます。

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、その手順を一覧表示します。

表 11–8 DHCP オプションの作業マップ

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	288ページの「DHCP オプションを作成する方法 (DHCP Manager)」
		289ページの「DHCP オプションを作成する方法 (コマンド行)」
		292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	290ページの「DHCP オプションの属性を変更する方法 (DHCP Manager)」
		291ページの「DHCP オプションの属性を変更する方法 (コマンド行)」
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	292ページの「DHCP オプションを削除する方法 (DHCP Manager)」
		292ページの「DHCP オプションを削除する方法 (コマンド行)」

オプションを作成する前に、表 11–9 に一覧表示してある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 11–9 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。コードはオプションカテゴリに適している必要がある ベンダー – 各ベンダークラスにつき 1 ～ 254 のコード値 サイト – 128 ～ 254 のコード値 拡張 – 77 ～ 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 – このブール型のデータ型に関連する値はない。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 NUMBER – 署名のない番号。たとえば、MTU オプションは 1500 のような数字を受け付ける 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
最小値	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 のデータ型で最小値が 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。NUMBER のデータ型では、最小値がそれぞれ 1、2、4、または 8 のオクテットを指定できる

表 11–9 DHCP オプションの属性 続く

オプションの属性	説明
最大値	オプションについて指定可能な値の最大値。前の例を基にすると、最大値が 2、最小値が 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 アドレスを含むことができる
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、ベンダー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、たとえば SUNW.Javastation のように、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。このオプションのタイプによって、同一クラスのすべてのクライアントとそのクラスのクライアントだけに伝えられる設定パラメータが定義できる 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 IA-32 ベースのマシンについては、ベンダークライアントクラスは常に SUNW.i86pc。Sparc ベースのマシンについては、ベンダークライアントクラスは、クライアントで uname --i と入力すると取得できる。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 コマンドによって文字列 SUNW,Ultra-1 が返される場合、ベンダークライアントクラスを SUNW.Ultra-1 と指定する

DHCP オプションの作成

DHCP プロトコルの既存のオプションにはないクライアント情報を伝える必要がある場合は、オプションを作成することができます。Solaris DHCP で定義されたすべてのオプションのリストについて、dhcptab(4) のマニュアルページを参照してから独自のオプションを作成してください。

dhtadm -A -s コマンドまたは DHCP Manager の「オプションの作成 (Create Option)」ダイアログボックスを使用して、新しいオプションを作成することができます。

図 11–17 は、DHCP Manager 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。

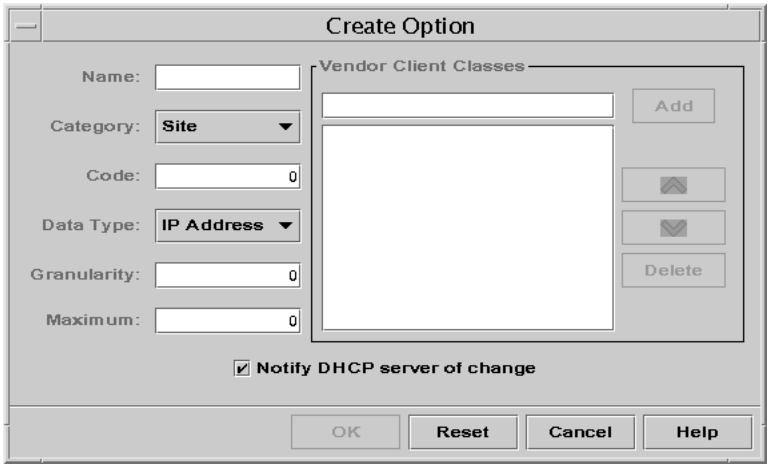


図 11–17　「オプションの作成 (Create Option)」ダイアログボックス

▼ DHCP オプションを作成する方法 (DHCP Manager)

1. 「オプション」タブを選択します。
2. 「編集」メニューから「作成」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、空白文字以外で 8 文字までの英数字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、タイプまたは値の選択をします。
   各設定の詳細については、表 11–9 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」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、クライアントに伝える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 (コマンド行)

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. 次のフォーマットでコマンドを入力します。

```
# dhtadm -A -s option-name-d 'category,code,data-type,granularity,maximum'
```

各項目には、次の内容を入力します。

option-name	8 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判定の詳細については、表 11–9 を参照のこと。
code	表 11–9 で説明されているように、オプションカテゴリに適する数値
data-type	`ASCII`、`IP`、`BOOLEAN`、`NUMBER`、または `OCTET`
granularity	表 11–9 で説明されているように、負にならない数字
maximum	表 11–9 で説明されているように、負にならない数字

次の 2 つのコマンドは、2 つの例です。

```
# dhtadm -A -s NewOpt -d 'Site,130,NUMBER,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は、DHCP Manager または `dhtadm` コマンドを使用して、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 Manager の「オプションの属性」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様の変更を反映する必要があることに注意してください。292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。

図 11–18 は、DHCP Manager 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。

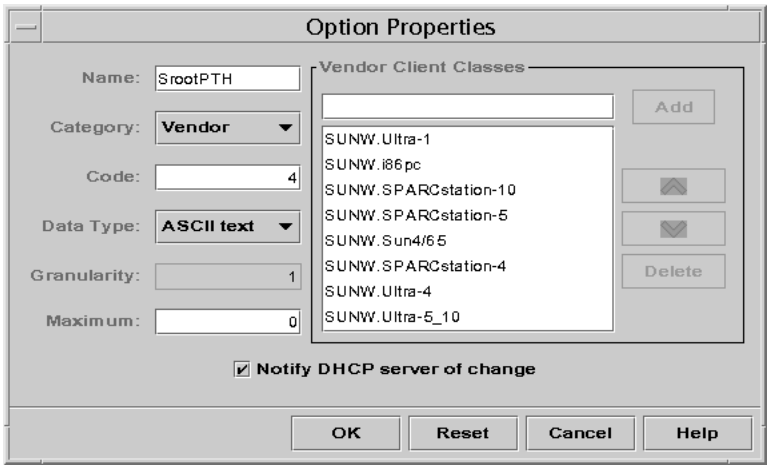


図 11–18　「オプションの属性 (Option Properties)」ダイアログボックス

▼ DHCP オプションの属性を変更する方法 (DHCP Manager)

1. 「オプション」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」メニューから「属性」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。
   これらの属性の詳細については、表 11–9 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」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 (コマンド行)

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. 次のフォーマットでコマンドを入力します。

```
# dhtadm -M -s option-name-d 'category,code,data-type,granularity,maximum'
```

各項目には、次の内容を入力します。

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.Ultra-1 SUNW.i86pc` となる。
code	表 11–9 で説明されているように、オプションカテゴリに適する数値
data-type	`ASCII`、`IP`、`BOOLEAN`、`NUMBER`、または `OCTET`
granularity	表 11–9 で説明されているように、負にならない数字
maximum	表 11–9 で説明されているように、負にならない数字

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。

次の 2 つのコマンドは、2 つの例です。

```
# dhtadm -M -s NewOpt -d 'Site,135,NUMBER,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が、DHCP サービス用のオプションを独自に定義した場合、DHCP Manager または `dhtadm` コマンドを使用して、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 Manager)

1. 「オプション」タブを選択します。
2. 「編集」メニューから「削除」を選択します。
   「オプションの削除」ダイアログボックスが開きます。
3. 「了解」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 (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. 次のフォーマットを使用して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は、オプションまたはシンボルの追加の詳細について、そのクライアント用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/default/inittab` ファイルを編集し、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。そのオプションをサーバー上であとから変更する場合は、その変更に応じてクライアントの `/etc/default/inittab` ファイルのエントリを変更する必要があります。

`/etc/default/inittab` ファイルの構文の詳細については、`dhcp_inittab(4)` のマニュアルページを参照してください。

注 - Solaris DHCP の以前のリリースで、`dhcptags` ファイルに DHCP オプションを追加した場合は、そのオプションを `/etc/default/inititab` ファイルに追加する必要があります。詳細については、326ページの「DHCP オプション情報」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クライアントのサポート

DHCP を使用すると、ネットワーク上の一定のクライアントマシンに Solaris 操作環境をインストールすることができます。Sun Enterprise Ultra マシンと、Solaris 操作環境を実行するためのハードウェアの必要条件を満たす Intel マシンに限り、この機能を使用することができます。

次の作業マップは、クライアントで 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が必要な高度な作業を示します。

表 11–10 DHCP ネットワークインストールの作業マップ

作業	説明	参照先
インストールサーバーの設定	Solaris サーバーを設定して、ネットワークから Solaris 操作環境をインストールしたいクライアントをサポートする	『*Solaris 8* のインストール (上級編)』の「ネットワーク上で Solaris ソフトウェアをインストールする準備」
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設定する	`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 8* のインストール (上級編)』の「ネットワーク上で Solaris ソフトウェアをインストールする準備」 `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 Manager または `dhtadm` を使用して、新しいベンダーオプションと、DHCP サーバーがインストール情報をクライアントに伝えるために使用するマクロを作成する	294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、そのスクリプトは DHCP 設定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 操作環境を適切にインストールするために必要な情報を伝える必要があります。表 11–11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 11–11 SUNW 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

名前 **(Name)**	コード **(Code)**	データ型 **(Data Type)**	最小値 **(Granularity)**	最大値 **(Maximum)**	ベンダークライアントクラス **(Verndor Client Class)**	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

表 11–11 SUNW 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 続く

名前 **(Name)**	コード **(Code)**	データ型 **(Data Type)**	最小値 **(Granularity)**	最大値 **(Maximum)**	ベンダークライアントクラス **(Vernder Client Class)**	説明
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
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* というフォーマットでの、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* というフォーマットでの、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。ここに一覧表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の決定については、表 11–9 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。表 11–12 では、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一覧表示しています。

表 11–12 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 と sparc のマクロ
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。

* 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決定については、表 11–9 を参照してください。

dhtadm コマンドまたは DHCP Manager 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方法が、最も容易でしょう。この方法をお奨めします。

298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」で、dhtadm コマンドを使用したスクリプトのサンプルを示します。DHCP Manager を使用する場合は、299ページの「 DHCP Manager 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

次の例を各システムに適用して Korn シェルスクリプトを作成し、表 11–11 に一覧表示されたすべてのオプションと一部の有用なマクロ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、パスに必ず変更してください。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 11–1 オプションとマクロを追加してネットワークインストールをサポートするサンプルスクリプト

```
# 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
# Define all Solaris-generic options under this macro named Solaris.
dhtadm -A -m Solaris -d \
':SrootIP4=172.21.0.2:SrootNM="blue2":SinstIP4=172.21.0.2: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.
```

(続く)

```
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 DHCP server.
dhtadm -M -m 172.20.64.64 -e BootSrvA=172.21.0.2
dhtadm -M -m 172.20.64.0 -e BootSrvA=172.21.0.2
dhtadm -M -m 172.20.64.128 -e BootSrvA=172.21.0.2
dhtadm -M -m 172.21.0.0 -e BootSrvA=172.21.0.2
dhtadm -M -m 172.22.0.0 -e BootSrvA=172.21.0.2
# Make sure we return hostnames 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=172.23.128.2:SrootNM="orange-svr-2":SrootPTH="/export/root/172.23.128.12":'
```

スーパーユーザーとしてこのスクリプトを実行し、オプションとマクロを dhcptab に追加します。このスクリプトを実行すると、Vendor= 文字列に一覧表示されるネットワーククライアントクラスに、DHCP を使用してネットワークからインストールできます。

DHCP Manager を使用したインストールオプションとマクロの作成

DHCP Manager を使用して、表 11–11 に一覧表示されたオプションと表 11–12 に一覧表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の図については、図 11–17 と図 11–16 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 Manager)

1. **DHCP Manager** で「オプション」を選択します。
2. 「編集」メニューから「作成」を選択します。
   「オプションの作成」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、そのオプションにふさわしい値を入力します。

   表 11–11 を使用して、作成する必要があるオプションのオプション名と値を調べます。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 を使用してインストールする実際のクライアント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の判定に関する情報については、表 11–9 を参照してください。

4. すべての値を入力したら、「了解」をクリックします。

5. 「オプション」タブで、今作成したオプションを選択します。

6. 「編集」メニューから「複製」を選択します。

   「オプションの複製」ダイアログボックスが開きます。

7. 別の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、最小値、最大値は通常は、変更する必要があります。これらの値については、表 11–11 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、300ページの手順 4 から 300ページの手順 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伝えるマクロが作成できます。

▼ 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 Manager)

1. **DHCP Manager** で「マクロ」を選択します。

2. 「編集」メニューから「作成」を選択します。

   「マクロの作成」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 11–12 を参照してください。

4. 「選択」ボタンをクリックします。

「オプションの選択」ダイアログボックスが開きます。

5. 「カテゴリ」リストで「ベンダー」を選択します。
   作成した「ベンダー」オプションが一覧表示されます。

6. マクロに追加するオプションを選択して、「了解」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 11–11 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、300ページの手順 4 から 301ページの手順 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」をクリックします。

第 12 章

DHCP の障害追跡

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、設定完了後 DHCP を使用する際の問題を解決する手助けとな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

サーバーを設定する際に検出される問題は一般に次のカテゴリに分類されます。

- データ保存方法として 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 データ保存方法として NIS+ を使用する場合に、検出される問題は次のように分類できます。

- NIS+ をデータ保存方法として選択できない
- NIS+ が適切に設定されない

- 権限の不足と資格が原因の NIS+ アクセス問題

NIS+ をデータ保存方法として選択できない

NIS+ をデータ保存方法として選択しようとして、DHCP Manager のデータ保存方法の選択肢に 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` で返されたりすることがあります。これは、NIS+ はネットワークで使用されている可能性はあるが、使用しているサーバーについて設定されていないことを意味します。NIS+ をデータ保存方法として選択する前に、サーバーマシンを NIS+ クライアントとして設定する必要があります。

サーバーを NIS+ クライアントとして設定する前に、ドメインを設定し、ドメインのマスターサーバーを実行しておく必要があります。ドメインのテーブルのマスターサーバーを生成する必要があり、ホストテーブルには、新しいクライアントのマシン (DHCP サーバーマシン) に関するエントリが必要です。『*Solaris* ネーミングの設定と構成』の「NIS+ クライアントの構成」で、NIS+ クライアントの構成について詳細に説明されています。

NIS+ が適切に設定されない

DHCP とともに正常に NIS+ を使用している場合に、NIS+ にあとから変更を加えるとエラーが検出され、設定上の問題が発生する場合があります。表 12–1 を使用して、設定問題の原因を判定してください。

表 12–1 NIS+ の設定問題

起こりうる問題	問題の判定方法	問題への対応
ルートオブジェクトが NIS+ ドメインに存在しない	コマンド `/usr/lib/nis/nisstat` を入力する このコマンドは、ドメインに関する統計情報を表示する。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* ネーミングの設定と構成』を参照して NIS+ ドメインを設定する
`password` 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させる `cat /etc/nsswitch.conf` この「nisplus」キーワードに関する `password` と `publickey` の項目を確認する	ネームサービススイッチの設定については、『*Solaris* ネーミングの設定と構成』を参照
ドメイン名が空である	次のコマンドを入力する `domainname` このコマンドによって空の文字列が一覧表示された場合は、このドメインについてドメイン名が設定されていない	データ保存方法に関する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* ネーミングの設定と構成』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、ファイルの存在を判定する `cat /var/nis/NIS_COLD_START`	データ保存方法に関する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* ネーミングの設定と構成』を参照

NIS+ アクセスに関する問題

NIS+ アクセス問題によって、不正な DES 資格や、NIS+ オブジェクトまたはテーブルを更新する権利が不適切であるというエラーメッセージを受け取る場合があります。表 12–2 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 12–2 NIS+ アクセス問題

起こりうる問題	問題の判定方法	問題への対応
DHCP サーバーマシンに、NIS+ ドメイン内の org_dir オブジェクトに対する作成アクセス権がない	次のコマンドを入力する `nisls -ld org_dir` アクセス権は、`r---rmcdrmcdr---` の形式で一覧表示される。この場合に、権利は、未認証、所有者、グループ、その他に個別に適用される。このオブジェクトの所有者は、次に一覧表示される	`nischmod` コマンドを使用して、`org_dir` の権利を変更する たとえば、グループに対する作成アクセス権を追加するには、`nischmod g+c org_dir` と入力する
	通常は、`org_dir` ディレクトリオブジェクトによって、所有者とグループの両方に、読み取り、変更、作成、削除といったすべての権利が与えられ、一方、その他と未認証には読み取りアクセスだけが与えられる	詳細は `nischmod(1)` のマニュアルページを参照
	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。次のコマンドを使用してこのグループを一覧表示する `nisls -ldg org_dir`	
DHCP サーバーに、`org_dir` オブジェクトの下に表を作成するアクセス権がない 通常は、サーバーマシンの主体名が `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` `Group「admin.myco.com」`に似た行を検索する 次のコマンドを使用して、グループの主体名を一覧表示する `nisgrpadm -l` *groupname*	`nisgrpadm` コマンドを使用してサーバーマシン名を追加する たとえば、サーバー名 `pacific` をグループ `admin.myco.com` に追加するには、次のように入力する `nisgrpadm -a admin.myco.com pacific.myco.com`
	たとえば、`nisgrpadm -l admin.myco.com` となる このサーバーマシン名は、グループの明示的なメンバーであるか、またはグループの暗黙のメンバーである必要がある	詳細は、`nisgrpadm(1)` のマニュアルページを参照

表 12–2 NIS+ アクセス問題 続く

起こりうる問題	問題の判定方法	問題への対応
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) の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 コマンドを使用して、DHCP サーバーマシンのセキュリティ資格を追加する 次の例では、ドメイン Faxco.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 -p unix.mercury@Faxco.COM \ -P mercury.Faxco.COM. DES Faxco.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 に記録されたり、サーバーデバッグ出力に表示されたりする場合があります。

表 12–3 IP アドレス割り当てとリースの問題

エラーメッセージ	説明	解決
There is no *n.n.n.n* dhcp-network table for DHCP client's network.	クライアントは特定の IP アドレスを要求するか、または現在の IP アドレスでリースを拡張しようとする。しかし DHCP サーバーは、そのアドレスに関する 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 Manager または dhcpcpnfig を使用して、ネットワークテーブルを再作成できる
ICMP ECHO reply to OFFER candidate: *n.n.n.n*, disabling	DHCP クライアントに提供しようとしている IP アドレスが、すでに使用されている。この状態は、複数の DHCP サーバーがこのアドレスを所有しているか、または 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設定されている場合に発生する	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しようとしている IP アドレスが、ネットワークテーブルの中にレコードを持っていない。この状態は、IP アドレスが選択されたあと、重複アドレスチェックが完了する前に、そのアドレスのレコードが DHCP ネットワークテーブルから削除された場合に発生する	DHCP Manager または pntadm を使用して、DHCP ネットワークテーブルが表示できる。その IP アドレスが失われている場合は、DHCP Manager (「アドレス」タブで「編集」メニューから「作成」を選択) または pntadm を使用してそのアドレスを作成する
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 Manager または pntadm を使用して、DHCP ネットワークテーブルが表示できる。その IP アドレスが失われている場合は、DHCP Manager (「アドレス」タブで「編集」メニューから「作成」を選択) または pntadm を使用してそのアドレスを作成する
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 Manager 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
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。サーバーはこのクライアントに別なアドレスを割り当てることはできない	DHCP Manager 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なアドレスを手動で割り当てる

表 12–3 IP アドレス割り当てとリースの問題 続く

エラーメッセージ	説明	解決
`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。割り当てるのは、1 つのアドレスであることが必要。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 Manager または `pntadm` を使用して、追加の手動割り当てを削除する
`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 Manager または `pntadm` を使用して、このネットワーク用の新しい IP アドレスを作成する
`Client:` *clientid* `lease on` *n.n.n.n* `expired.`	このリースにネゴシエーションの余地がなく、期限切れである	クライアントを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るようにする
`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	このクライアントで新たな探索メッセージを自動的に発行するようにする。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 Manager では、「サービス」メニューから「変更」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。この状態は、クライアントが、更新した DHCP プロトコル、RFC 2131 に準拠していない場合に発生する可能性がある	クライアントのソフトウェアを更新する

表 12–3 IP アドレス割り当てとリースの問題 続く

エラーメッセージ	説明	解決
`Client: clientid is trying to renew n.n.n.n, an IP address it has not leased.`	このクライアントについて DHCP ネットワークテーブルに記録された 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 Manager 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
`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、このアドレスでは DHCP ネットワークテーブルに登録されていない。そのため、要求が 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 アドレスを使用しているのに、このクライアントのレコードを削除してしまった場合もある DHCP Manager 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

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。

- 310ページの「DHCP サーバーとの通信の問題」
- 321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て、マシンを再起動すると、クライアントは DHCP サーバーに通信してそのネットワークの設定を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、次のようなメッセージが表示されます。

```
DHCP or BOOTP server not responding
```

問題を判定するには、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。

収集した情報を使用して、問題がクライアント側にあるのかサーバーマシン側にあるのか、あるいはリレーエージェントにあるのかを判定し、解決法を見つけることができます。

▼ 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実行している場合は、クライアントのデバッグモードでの実行については、そのクライアント用のマニュアルを参照してください。

Solaris DHCP クライアントを実行している場合は、次の手順に従います。

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. **DHCP クライアントデーモンを終了し、次のコマンドを使用してそのデーモンをデバッグモードで再起動します。**

```
# 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、313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. **DHCP** サーバーデーモンを終了し、次のコマンドを使用してそのデーモンをデバッグモードで再起動します。

```
# pkill -x in.dhcpd
# /usr/lib/inet/in.dhcpd -d -v
```

また、デーモンを実行する際に通常使用する `in.dhcpd` コマンド行オプションも使用する必要があります。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、314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
```

たとえば、次のように入力します。

```
# /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あと 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。

   クライアントの再起動については、311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」で説明されています。

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んだ出力ファイルを表示させます。

```
# /usr/sbin/snoop -i snoop-output-filename -x0 -v
```

たとえば、次のように入力します。

```
# /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* を、たとえば le0 のようなクライアントのネットワークインタフェースの名前に置き換えてください。

出力の解釈については、318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 12–1 では、DHCP クライアントが DHCP 要求を送信し、DHCP サーバーから設定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 **12–1** 通常の DHCP クライアントデバッグ出力

```
/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
```

(続く)

```
/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
/sbin/dhcpagent: info: setting IP address on le0 to 102.23.3.233
/sbin/dhcpagent: info: setting broadcast address on le0 to 102.23.63.255
/sbin/dhcpagent: info: added default router 102.23.0.1 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 12–2 のようなデバッグ出力が表示されます。

例 **12–2** クライアントがサーバーの回答を受信しない場合の DHCP クライアントデバッグ出力

```
/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
```

このメッセージが表示された場合は、サーバーの回答はクライアントに届いていません。つまり、要求がサーバーに届いていないか、またはサーバーがクライアントに回答を送信できないということになります。312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」で説明しているように、サーバーで snoop を実行して、クライアントからのパケットがサーバーに届いたかどうかを判定します。

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、サーバーの設定情報とそれに続く、各ネットワークインタフェースの情報を表示します。そのあと、デバッグ出力は、デーモンが処理した要求の情報を表示します。次の例では、DHCP サーバーと BOOTP リレーエージェントに関する出力の例を示しています。例 12–3 は DHCP サーバーに関するデバッグ出力を示します。このサーバーは起動直後で、

応答しない別の DHCP サーバーが所有するアドレスを使用するクライアントのリースを拡張します。

例 12–3 DHCP サーバーに関するデバッグ出力

```
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...
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 102.21.255.255
Netmask: 255.255.0.0
Address: 102.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
Broadcast: 102.22.255.255
Netmask: 255.255.0.0
Address: 102.22.0.1
Monitor (0007/qe0) started...
Thread Id: 0007 - Monitoring Interface: qe0 *****
MTU: 1500      Type: DLPI
Broadcast: 102.23.63.255
Netmask: 255.255.192.0
Address: 102.23.0.1
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 102.21.0.4
Datagram received on network device: qe0
Client: 0800201DBA3A is requesting verification of address owned by 102.21.0.4
Datagram received on network device: qe0
Client: 0800201DBA3A is requesting verification of address owned by 102.21.0.4
Datagram received on network device: qe0
Client: 0800201DBA3A maps to IP: 102.23.3.233
Unicasting datagram to 102.23.3.233 address.
Adding ARP entry: 102.23.3.233 == 0800201DBA3A
DHCP EXTEND 0934312543 0934316143 102.23.3.233 102.21.0.2
          0800201DBA3A SUNW.SPARCstation-10 0800201DBA3A
```

例 12–4 は、BOOTP リレーエージェントとして起動し、クライアントから DHCP サーバーへ要求をリレーし、サーバーの回答をクライアントにリレーする DHCP デーモンからのデバッグ出力を示します。

例 **12–4**　BOOTP リレーに関するデバッグ出力の例

```
Relay destination: 102.21.0.4 (blue-servr2)           network: 102.21.0.0
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: 102.21.255.255
Netmask: 255.255.0.0
Address: 102.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
Broadcast: 102.22.255.255
Netmask: 255.255.0.0
Address: 102.22.0.1
Monitor (0007/qe0) started...
Thread Id: 0007 - Monitoring Interface: qe0 *****
MTU: 1500      Type: DLPI
Broadcast: 102.23.63.255
Netmask: 255.255.192.0
Address: 102.23.0.1
Relaying request 0800201DBA3A to 102.21.0.4, server port.
BOOTP RELAY-SRVR 0934297685 0000000000 0.0.0.0 102.21.0.4 0800201DBA3A N/A 0800201DBA3A
Packet received from relay agent: 102.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 102.23.3.233 address.
Adding ARP entry: 102.23.3.233 == 0800201DBA3A
BOOTP RELAY-CLNT 0934297688 0000000000 102.23.0.1 102.23.3.233 0800201DBA3A N/
A 0800201DBA3A
Relaying request 0800201DBA3A to 102.21.0.4, server port.
BOOTP RELAY-SRVR 0934297689 0000000000 0.0.0.0 102.21.0.4 0800201DBA3A N/A 0800201DBA3A
Packet received from relay agent: 102.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 102.23.3.233 address.
Adding ARP entry: 102.23.3.233 == 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 12–4 を使用してエラーメッセージまたは条件を検索し、解決法を見つけてください。

表 12–4 DHCP サーバーのエラーメッセージ

メッセージ	説明	解決法
ICMP ECHO reply to OFFER candidate: *ip_address* disabling	DHCP サーバーは、クライアントに IP アドレスを提供する前に、アドレスを ping 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	設定するアドレスが使用されていないことを確認する
No more IP addresses on *network_address* network.	ネットワークテーブルごとのクライアントのアドレスの中に利用可能な IP アドレスがない	DHCP Manager または pntadm を使用して IP アドレスを追加割当する。DHCP デーモンが複数のサブネットを監視している場合は、追加のアドレスが必ずクライアントが割り当てられているサブネット用であること
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 Manager 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 サーバーの設定中に、サブネット用の DHCP ネットワークテーブルが作成されなかった	DHCP Manager 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 サーバーの設定中に、サブネット用の DHCP ネットワークテーブルが作成されなかった	DHCP Manager 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

表 12–4　DHCP サーバーのエラーメッセージ　続く

メッセージ	説明	解決法
`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	このデバイスを設定する必要がない場合は、このメッセージを無視する

`Client` *client_id* `is trying to verify unrecorded address` *ip_address*`, ignored.`	クライアント上の `/etc/dhcp/`*interface*`.dhc` ファイルにある IP アドレスとクライアント ID が、DHCP サーバーで確認された DHCP ネットワークデータベースにある IP アドレスやクライアント ID と一致しない
	
	この状態は、ローカルファイルを DHCP データ保存方法として使用し、情報を共有しない複数の DHCP サーバーを持っているか、DHCP ネットワークテーブルを変更した場合に発生する
		次のコマンドを入力して、クライアントの DHCP プロトコルを再起動する
		
		`ifconfig` *interface* `dhcp release`
		
		`ifconfig` *interface* `dhcp start`

DHCP snoop 出力

snoop 出力に、DHCP クライアントマシンと DHCP サーバーマシンの間でパケットが交換されていることが表示されます。各マシンの IP アドレスと、中間のリレーエージェントやルーターがパケットごとに表示されます。パケットの交換が表示されない場合は、クライアントマシンがサーバーマシンとまったく通信できていない場合がありますが、これは下位の問題です。一般的な障害追跡の方法については、120ページの「一般的な障害追跡方法」を参照してください。

snoop 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通して送受信される必要があるなど、期待される動作の内容を理解しておく必要があります。また、関連するシステムの MAC アドレスや IP アドレス、システムが複数の場合はネットワークインタフェースについても把握し、それらの値が期待されるものであ

ることを判定できるようにしておく必要があります。例 12–5 では、blue-servr2 上にある DHCP サーバーから、MAC アドレスが 8:0:20:8e:f3:7e で、割り当てられた IP アドレスが 192.168.252.6、ホスト名が white-6 のクライアントに送信された DHCP 確認メッセージに関する通常の snoop 出力を示します。標準ネットワークオプションのいくつかと複数のベンダー固有のオプションも、クライアントに送信されます。

例 12–5 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 = 102.21.0.4, 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
```

(続く)

```
DHCP: Your client address (yiaddr) = 192.168.252.6
DHCP: Next server address (siaddr) = 102.21.0.2
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 = 102.21.0.4
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 = 102.21.0.4
DHCP: DNS Domain Name = sem.west.dor.com
DHCP: DNS Servers at = 102.21.0.1
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/s28/base.s28s_nxt/latest/Solaris_8/Tools/Boot"
DHCP:   (12) 32 octets  "/export/s28/base.s28s_nxt/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.west.dor
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
```

(続く)

```
400: 616e 7469 630f 0578 7465 726d 0435 2f65    ific...xterm.5/e
416: 7870 6f72 742f 7332 382f 6261 7365 2e73    xport/s28/base.s
432: 3238 735f 776f 732f 6c61 7465 7374 2f53    28s_nxt/latest/S
448: 6f6c 6172 6973 5f38 2f54 6f6f 6c73 2f42    olaris_8/Tools/B
464: 6f6f 740c 202f 6578 706f 7274 2f73 3238    oot. /export/s28
480: 2f62 6173 652e 7332 3873 5f77 6f73 2f6c    /base.s28s_nxt/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設定情報の中に、誤った NIS ドメイン名や不正確なルーター IP アドレスのように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のオプションの値を調べる必要があります。

次の一般的なガイドラインを使用して、不正確な情報がクライアントに伝えられた場所を判定してください。

- 277ページの「DHCP サーバーで定義されたマクロを表示する方法 (DHCP Manager)」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。179ページの「マクロ処理の順序」にある情報を再検討して、このクライアントについて自動的に処理されたマクロを判定します。
- ネットワークテーブルを調べて、クライアントの IP アドレスに設定マクロとして割り当てられたマクロがもしあればその内容を判定します。詳細については、261ページの「DHCP サービスで IP アドレスを使用して作業する」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伝えられるようにします。278ページの「DHCP マクロの変更」を参照してください。

第 13 章

DHCP のリファレンス

この章では、DHCP に関する有益な情報を一覧表示します。ファイルと、それらのファイルを使用するコマンドとの間の関係は説明しますが、コマンドの使用方法は説明しません。コマンドはそれらのマニュアルページにリンクしているため、コマンドをクリックして、そのコマンドの関する情報を検索することができます。

DHCP のコマンド

表 13–1 では、ネットワーク上で DHCP を管理する際に役立つコマンドを一覧表示します。

表 13–1 DHCP で使用されるコマンド

コマンドのマニュアルページ	コマンドの説明
`dhtadm(1M)`	`dhcptab` ファイルのオプションとマクロを変更するために使用する。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。データベース内で特定のオプション値を迅速に検索するには、`dhtadm` に `-P` オプションを付けて使用し、それを `grep` コマンドに渡す
`pntadm(1M)`	クライアント ID を IP アドレスに割り当て、場合によっては設定情報を IP アドレスに関連付ける DHCP ネットワークテーブルを変更するのに使用する
`dhcpconfig(1M)`	DHCP サーバーと BOOTP リレーエージェントの設定と設定解除を実行するために使用する。`dhcpconfig` は、`dhtadm` と `pntadm` を使用して、`dhcptab` と DHCP ネットワークテーブルを作成し、変更する

表 13–1 DHCP で使用されるコマンド 続く

コマンドのマニュアルページ	コマンドの説明
in.dhcpd(1M)	DHCP サーバーデーモン。このコマンドは、DHCP サービスの起動と停止のスクリプトである /etc/init.d/dhcp で使用する。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(1M)	DHCP Manager。DHCP サービスの設定と管理を行うために使用されるグラフィカルツール。DHCP Manager は、推奨される Solaris DHCP 管理ツール
ifconfig(1M)	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設定したりする。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(1)	クライアントマシンのシステム起動スクリプトによって使用され、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。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(1M)	ネットワーク間でやり取りされるパケットの内容を取りこみ、表示するのに使用される。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(1M)	DHCP クライアントデーモン。DHCP プロトコルのクライアントサイドを実装している

DHCP のファイル

表 13–2 は、Solaris DHCP に関連するファイルを一覧表示します。

表 13–2 DHCP のデーモンとコマンドが使用するファイル

ファイルのマニュアルページ	説明
dhcptab(4)	割り当てられた値とともにオプションとして記録されている設定情報の表。これらの値はマクロにグループ化される。dhcptab ファイルの位置は、DHCP 情報について使用するデータ保存方法によって決まる
dhcp_network(4)	IP アドレスをクライアント ID と設定オプションに割り当てる。DHCP ネットワークテーブルは、172.21.32.0 のようなネットワークの IP アドレスに従って命名される。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位置は、DHCP 情報について使用するデータ保存方法によって決まる

表 13–2 DHCP のデーモンとコマンドが使用するファイル 続く

ファイルのマニュアルページ	説明
dhcp(4)	DHCP デーモンの起動オプションと、DHCP サービスで使用するデータ保存方法の位置とタイプを記録する。このファイルは、/etc/default ディレクトリに存在する
nsswitch.conf(4)	ネームサービスデータベースの位置と、さまざまな種類の情報を検索する際にデータベースを検索する順序を指定する。正確な設定情報を得るために DHCP サーバーを設定する際に、この nsswitch.conf ファイルが調べられる。このファイルは、/etc ディレクトリに存在する
resolv.conf(4)	DHS リゾルバが使用する情報が入っている。DHCP サーバーの設定中に、このファイルは、DNS ドメインと DNS サーバーに関する情報について調べられる。このファイルは、/etc ディレクトリに存在する
dhcp.*interface*	DHCP が、dhcp.qe0 のようにファイル名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使用されることを示す。dhcp.*interface* ファイルには、クライアント上で DHCP を起動するために使用される 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る場合がある。このファイルは、/etc ディレクトリに存在する
interface.dhc	一定のネットワークインタフェースについて、DHCP から取得される設定パラメータが含まれる。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に関する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。これらのパラメータの詳細については、このファイル自体か、デーモンの dhcpagent(1M) のマニュアルページを参照
dhcp_inittab(4)	データ型のような DHCP オプションコードの様相を定義し、ニーモニックラベルを割り当てる。dhcpinfo は、/etc/dhcp/inittab ファイルにある情報を使用して、さらに意味のある情報をユーザーに提供する。このファイルは、/etc/dhcp/dhcptags ファイルに代わる。この入れ替えの詳細については、326ページの「DHCP オプション情報」を参照

DHCP オプション情報

DHCP オプション情報は従来、サーバーの dhcptab ファイル、クライアントの dhcptags ファイル、in.dhcpd、snoop、dhcpinfo、dhcpmgr の内部テーブルなど、Solaris DHCP 内の複数の場所に保管されていました。オプション情報を統合するため、Solaris 8 DHCP 製品に /etc/dhcp/inittab ファイルが導入されました。このファイルの詳細については、dhcp_inittab(4) を参照してください。

inittab ファイルは現在、dhcptags ファイルに代わるものとして Solaris DHCP クライアントで使用されています。dhcptags ファイルは、DHCP パケットで受信されたオプションコードに関する情報を取得するために使用されていました。現在、inittab ファイルがこの情報を提供します。将来のリリースでは、DHCP サーバーの in.dhcpd、snoop、dhcpmgr のプログラムも、このファイルを使用するようになります。

注 - Solaris DHCP を使用するサイトの多くは、この変更に影響されません。サイトが影響されるのは、Solaris 8 にアップグレードしようとしているが、以前に新しい DHCP オプションを作成して /etc/dhcp/dhcptags ファイルを変更していて、それらの変更を維持したい場合のみです。アップグレードするときに、dhcptags ファイルが変更されており inittab ファイルを変更する必要があることを、アップグレードログが通知します。

---

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ており、使用する構文が違います。

dhcptags のエントリの例は次の通りです。

```
33 StaticRt - IPList Static_Routes
```

33 は DHCP パケットで伝えら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、より説明的な名前です。

inittab ファイルは、各オプションを記述した 1 行のレコードから構成されています。そのフォーマットは、dhcptab でシンボルを定義するための形式と似ています。inittab の構文について、表 13–3 で説明します。

表 13–3 DHCP inittab 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。有効なタイプは、IP、Ascii、Octet、Number、Bool、Unumber8、Unumber16、Unumber32、Snumber8、Snumber16、Snumber32。number に関しては、先頭に付いている「U」または「S」は、その number が符号付きまたは符号なしであることを表し、あとに続く数字は、ビット数を数字で示したもの
granularity	このオプションの値全体を構成するデータの単位数を記述する。Number の場合、granularity はバイト数を数字で示したもの
maximum	このオプションについて許容される値全体の数を記述する。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。この値は `s`、`d`、`m`、`i` のいずれかにする。`s`、`d`、`m`、`i` は次のとおり
	`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`、最小値が `2`、最大値が無限大 (`0`) であるため、期待される値は、潜在的には無限の

IP アドレスの組です。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
```

最小値 1 と最大値 1 は、このオプションについて 1 つの IP アドレスが予想されることを意味しています。

ネットワークハードウェアの ARP 割り当て

ネットワークハードウェアの ARP (Address Resolution Protocol) 割り当ては、DHCP ネットワークテーブルにあるクライアント ID の先頭で指定されます。表 13–4 を使用して、クライアント ID で使用するコードの内容を判定してください。

コードは http://www.iana.com/numbers.html にある ARP registrations セクション内の Internet Assigned Numbers Authority (IANA) によって割り当てられています。

表 13–4 ARP タイプ

コード	ハードウェアのタイプ
1	Ethernet (10MB)
2	実験用 Ethernet (3MB)

表 13–4　ARP タイプ　続く

コード	ハードウェアのタイプ
3	Amateur Radio AX.25
4	Proteon ProNET トークンリング
5	Chaos
6	IEEE 802
7	ARCNET
8	Hyperchannel
9	Lanstar
10	Autonet Short Address
11	LocalTalk
12	LocalTalk (IBM PCNet または SYTEK LocalNET)
13	Ultra Link
14	SMDS
15	Frame Relay
16	ATM (Asynchronous Transmission Mode)
17	HDLC
18	Fibre Channel
19	ATM (Asynchronous Transmission Mode)
20	シリアル回線
21	ATM (Asynchronous Transmission Mode)
22	MIL-STD-188-220
23	Metricom
24	IEEE 1394.1995

表 **13–4** ARP タイプ 続く

コード	ハードウェアのタイプ
25	MAPOS
26	Twinaxial
27	EUI-64
28	HIPARP

第 14 章

IPv6 の概要

Internet Protocol、バージョン 6 (IPv6) は、現在の IPv4 から正常進化をしながら、飛躍的な進歩を図った Internet Protocol (IP) の新バージョンです。定義されている移行機構を利用して IPv6 を導入しても今までの操作方法が混乱することはありません。IPv6 ではアドレス空間が増え、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。

IPv6 の機能

IP4 から IP6 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るとともに、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ました。

  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。これはノードセットを識別して、任意キャストアドレスに送信されたパケットを指定ノードに配信する機能です。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。この変更によってアドレスのサイズは増えましたが、パケットの共通処理や IPv6 ヘッダーの帯域幅は、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、オプションの長さに関する制限が緩和されて、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適したパケットのラベル指定が可能になりました。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。

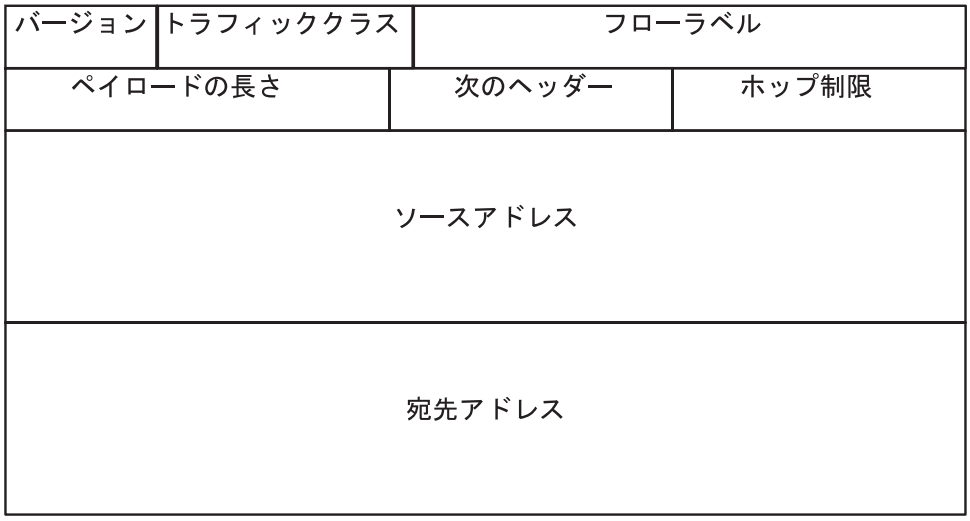


図 14–1　IPv6 ヘッダーフォーマット

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 = 6
- トラフィッククラス – 8 ビットトラフィッククラスの値 (355ページの「トラフィッククラス」を参照)
- フローラベル – 24 ビットフィールド (353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 – オクテットによる 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のヘッダータイプを識別する。IPv4 プロトコルフィールドと同じ値を使用する (334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 (334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 (経路を発信元が指定するヘッダー) 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。パケットのデリバリパスでは、どのルーターも、最終的な宛先に到着するまでほとんどの IPv6 拡張ヘッダーの確認や処理は行いません。そのため、オプションがあるパケットのルーター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。IPv4 オプションと異なるその他の改良点として、IPv6 拡張ヘッダーは長さを任意に設定でき、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 オクテットの整数の倍数の長さで、後続のヘッダーの配列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 – 飛び飛び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、ユニキャスト、任意キャスト、マルチキャストの 3 種類のタイプがあります。

- 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。その場合、任意キャストアドレスに送信される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グループを識別する。その場合、マルチキャストアドレスに送信されるパケットは、そのグループのすべてのインタフェースに配信される

IPv6 にはブロードキャストアドレスはない。マルチキャストアドレスが代わりをする

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。これは、IPv4 のアドレス領域の 40 億 x 40 億倍の大きさに相当します。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低下するため、利用できるアドレス数は減ります。ただし現状では、IPv6 で提供するアドレス領域で十分です。

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。表 14–1 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	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

表 14–1 フォーマットプレフィックスの割り当て 続く

割り当て	プレフィックス (バイナリ)	アドレス領域の端数
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

割り当てでは、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領域が予約されています。残りのアドレス領域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領域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、アドレス領域の約 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、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

各記号は次の内容を示します。

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 を割り当てた組織用であり、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。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 では、リンク上のインタフェースを識別します。Interface ID は、リンク上で一意であるものとします。さらに広い範囲で一意であってもかまいません。通常、インタフェース識別子は識別子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。

ローカル用アドレス

ローカル用アドレスは、ローカルルート範囲だけのユニキャストアドレス (サブネットまたは加入者ネットワーク内) であり、ローカルまたはグローバルな一意の範囲が適用されます。このアドレスは、サイト内において、グローバルアドレスまでのプラグアンドプレイローカル通信とブートストラップに使用します。

ローカル用アドレスには、リンクローカルとサイトローカルの 2 種類が定義されています。リンクローカル用は 1 つのリンク用であり、サイトローカル用は 1 つのサイト用です。表 14–3 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	*n* ビット	118-*n* 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	*n* ビット	*m* ビット	118-(*n*+ *m*) ビット
1111111011	0	Subnet ID	Interface ID

どちらのローカル用アドレスタイプも、インタフェース ID はそれを使用するドメインで一意であるものとします。通常、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にすると、他のアドレス指定をしなくても大規模なプライベートインターネットを構築できます。

ローカル用アドレスでは、グローバルインターネットに接続していない組織でも、グローバルインターネットアドレス領域のアドレスプレフィックスを要求せずに操作ができます。組織からグローバルインターネットにあとから接続する場合、Subnet ID と Interface ID をグローバルプレフィックスと組み合わせれば (たとえば Registry ID + Provider ID + Subscriber ID)、グローバルアドレスを作成できます。インターネットに接続するときにプライベート (非グローバル) IPv4 アドレスを使用して番号をマニュアルで指定し直さなければならない IPv4 に比べ、大幅に強化された点です。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、ホストとルーター向けに、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 ビットの 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。そのフォーマットを表 14–5 に示します。

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。このアドレスはアプリケーション、API、とオペレーティングシステムの実装内で主に使用します。このタイプのアドレスを IPv4 マップ IPv6 アドレスと呼びます。そのフォーマットを表 14–6 に示します。

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 (通常は異なるノードに所属) に割り当てるアドレスであり、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をルートシーケンスの一部に使用すると、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ノードで選択できます。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、任意キャストアドレスを設定して、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)。このような任意キャストを、IPv6 ルーティングヘッダーの中間アドレスに使用すれば、特定のプロバイダやプロバイダシーケンスでパケットを配信できます。また、任意キャストアドレスは特定のサブネットに関連づけられたルーター

セットや、特定のルーティングドメイン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とき、つまり任意キャストに変換する場合、任意キャストアドレスであることを表すため、アドレスを割り当てるノードを明示的に設定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	Group 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する

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、指定グループ内で、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 におけるルーティングは、CIDR における IPv4 の場合のルーティングとほぼ同じですが、IPv4 の 32 ビットアドレスに対し、IPv6 では 128 ビットアドレスを使用します。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。次にそれを示します。

- 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で IPv6 アドレスのシーケンスを作成します。IPv6 ソースでは、ルーティングオプションで複数の中間ノード (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 のルーズソースとルートオプションによく似た機能です。

アドレスシーケンスを一般的に使用する場合、通常は、(パケットが IPv6 認証ヘッダーで正しく認証された場合) アドレスシーケンスを保存した受信パケットのルートを IPv6 ホストで逆戻りして発信者にパケットを戻す必要があります。IPv6 ホストの実装では、この方式によりソースルートの処理と逆引きをサポートしています。これが、プロバイダ選択や拡張アドレスなどの新機能を実装するホストで IPv6 ホストの実装を使用するためのポイントです。

IPv6 の近傍探索

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。

- ルーター発見 – 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 – リンク MTU (最大伝送単位) などのリンクパラメータ、またはホップ制限値などのインターネットパラメータをノードが調べて出力パケットを転送する
- 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 アドレス以外不明な場合に、ノードが近傍のリンク層を判定する
- 次のホップの決定 – 宛先のトラフィックを送信する宛先になる近傍の IP アドレスに到る IP 宛先アドレスの対応付けがアルゴリズムで決まる。次のホップはルーターか宛先になる
- 近傍不到達検出 – 近傍にアクセスできない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、別のノードが使用している場合にそれを判定する

- 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

近傍探索では、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、リダイレクトメッセージという 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。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、ホストはルーターに対して、次回に予定されているタイミングではなく、今すぐルーター通知を生成するように求めるルーター要請を送信できる
- ルーター通知 – 定期的に、あるいはルーター要請メッセージに応じて、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。ルーター通知には、オンリンク判定またはアドレス設定、あるいはホップ制限値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、あるいは近傍がキャッシュリンク層アドレスでアクセスできるかどうかを確認するための情報としてノードから送信される。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、分離近傍不到達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、オンリンク判定や自動アドレス設定に使用するプレフィックスリストが含まれます。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。ホストでは、パケット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、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。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の使用することを指定するホップ制限などのインターネットパラメータや、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。これにより、ルーターに設定されて関連付けられたすべてのホストに伝達される重要なパラメータを簡単に集中管理できます。

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ない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。そのためには、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、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肯定確認を戻します (たとえば、最も新しい TCP 肯定を受信したなど)。肯定応答が得られない場合、ノードは次のホップからのアクセス確認として、近傍通知を要請するユニキャスト近傍要請メッセージを送信します。不要なネットワークト

ラフィックを避けるため、検証メッセージはノードがアクティブでパケットを送信中の近傍にだけ送信されます。

上記の一般的な問題以外に、近傍探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、小数の (非要請) 近傍通知パケットを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
- 入力負荷調整 – インタフェースを複写したノードでは、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複写の負荷調整ができる。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

  ルーターでは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させ、これによって近傍では近傍要請メッセージでルーターのリンク層アドレスを確認することで負荷を調整する。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では、等価サービスを提供するノードセットの 1 つを識別し、同じリンクの複数のノードは同じ任意キャストを認識するように設定できる。近傍探索では、ノードが同じ宛先に対する複数の近傍通知を受け取るよう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。これで複数の通知の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、たとえばこのプロトコルを実装していないノードを処理するといった一般的な機能ではない

IPv4 との比較

IPv6 近傍探索プロトコルは、IPv4 プロトコル ARP (アドレス解決プロトコル)、ICMP ルーター発見、ICMP リダイレクトを組み合わせたようなものです。ホスト条件ではデッドゲートウェイ検出 (近傍不到達検出の課題) に対応できる可能性のあ

るアルゴリズムがいくつか指定されていますが、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。

近傍探索プロトコルでは、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。

- ルーター発見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スヌープする必要はない
- 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
- ルーターでは、MTU が定義されていないすべてのノードで同じ MTU 値を使用するように、ホストがリンクで使用する MTU を通知する
- 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し、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、リンクにない次のホップを指定する。IPv6 リダイレクト機構は XRedirect 機能に似ている。ノードがオンリンク宛先のすべてのプレフィックスを知ることが望ましくない場合、あるいは不可能な場合は、非ブロードキャストと共有媒体リンクで有効
- 近傍不到達検出は、障害ルーター、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、リンク層アドレスが変更されたノードがあるときの、パケットの頑強性を大幅に強化した主要機能のひとつ。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、近傍探索では、ハーフリンクエラー (近傍不到達検出を利用) 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。近傍不到達検出で、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 (ルーター通知とリダイレクトメッセージ用に) ルーターを一意に識別しておけば、サイトの番号付けの変更で新しいグローバルプレフィックスが使用されても、ホストでルーター関連付けを維持できる
- 近傍探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り、標準 IP 認証とセキュリティ機構を必要に応じて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決定します。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、その他情報、または両方)、そしてアドレスの場合、ステートレス機構で取得するか、ステートフル機能で取得するか、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。アドレスはこの 2 つ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設定情報、またはその両方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が一意で正しくルートできれば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。正確なアドレス管理に対してサイトで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、インタフェースに結合さ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。アドレス結合の終了を効率的に行うため、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結合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は、そのアドレスを使用中のアプリケーションにおいて、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だけに使用します。

重複アドレスの検出アルゴリズム

所定のリンクにおける設定済みアドレスをすべて一意にするため、インタフェースにアドレスを割り当てる前に、ノードは重複アドレス検出アルゴリズムを実行します。ステートレスとステートフルのどちらの自動設定で得られたアドレスにも重複アドレス検出アルゴリズムは実行されます。

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能により、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の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検出手順が済んでいる場合もあります。

IPv6 プロトコルの概要

ここでは、インタフェース自身の自動設定の概要について説明します。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。

ただし、リンクローカルアドレスをインタフェースに割り当てて使用する場合、この仮アドレスが別のノードやリンクで使用されていないことを確認する必要があります。特に、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するときは注意してください。別のノードが目的のアドレスを使用済みの場合、近傍要請でその旨を伝える内容が戻ります。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、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。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。新しい (一意と思われる) インタフェース識別子を利用して自動設定機構を実行すると、簡単にこの状態から復帰できます。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。

ノードにおいて仮リンクローカルアドレスが一意であると判定されると、インタフェースに割り当てられます。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が、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、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クスと寿命値が保存され、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、前の通知で受け取った情報を追加し更新します。

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、アドレスの一意性は、主にインタフェース識別子から生成されるアドレスの段階で決まります。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、マニュアルやステートフルアドレス自動設定で得られたアドレスの場合は、個別に一意であることを確認する必要があります。重複アドレス検出を実行するのに手間がかかりすぎて意味がなくなるサイトの場合は、インタフェース別設定フラグの管理設定で重複アドレス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には、ルーター通知の待機とリンクローカルアドレスをホストで生成を並列に実行します (さらに一意性を確認)。これはルーターにおけるルーター要請に対する応答が遅れる場合があり、上記 2 つの手順を 1 つずつ実行すると、自動設定全体の時間が大幅に延長される場合があります。

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で行われるので、移動ノード、ホストまたはルーターが宛先のパケットは、そのホームリンク (ホーム IPv6 サブネットが存在するリンク) に関連付けられていないとノードにアクセスできません。ノードの移動に関係なく通信を継続するには、新しいリンクに移動するたびに、移動ノードの IP アドレスを変更する必要があります。ただし、移動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接続と上位層接続が失われます。以上のことから、将来、インターネットの発展に移動コンピュータが依存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 サポートでは、移動ノードがリンク間を移動しても移動ノードの IP アドレスは変更されません。そのため、移動ノードに対する IP アドレスの割り当ては、ホームリンク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、移動ノードのホームアドレスにルートされたパケットは、移動ノードが現在インターネットのどこに関連付けられていても宛先にアクセスできます。移動ノードが新しいリンクに移動しても他のノード (固定または移動) との通信は途切れません。

ただし、ホームを離れた移動ノードとの透過ルーティングパケットの問題は IPv6 移動サポートで解決できますが、移動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片づく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)
- 移動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

ホストでは IPv6 ヘッダーで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して、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など、IPv6 ルーターによる特別処理を必要とするパケットを識別します。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

IPv6 ヘッダーで 24 ビットフローラベルフィールドをソースに使用して、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など、IPv6 ルーターによる特別処理を必要とするパケットにラベルを設定できます。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。フローラベルフィールドの機能をサポートしていないホストやルーターでは、パケットの生成時にフィールドをゼロに設定し、パケットの送信時にフィールドの値を前送りし、パケットの受信時にフィールドを無視する必要があります。

フローとは

フローは特定のソースから、ソースがルーターの逆引きによる特別処理を必要とする特定の (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シーケンスです。特別な処理の特性は、リソース予約プロトコルなどの制御プロトコル、あるいはホップバイホップオプションなど、フローのパケットそのものに保存された情報によってルーターに伝達されます。

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、フローラベルゼロを運びます。

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 (疑似的な) ランダム選択で 1 から FFFFFF 16 進数の範囲から均等に選択します。このランダム割り当てにより、フローラベルフィールド内のビットセットは、フローの状態をルーターが調べるときのハッシュキーとして利用できるようになります。

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あると、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されます (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を除く)。これらのパケットのどれかにルーティングヘッダーがあると、ルーティングヘッダーまでは、すべての拡張ヘッダーが同じ内容で生成されます (ルーティングヘッダーの次のヘッダーフィールドを除く)。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。条件違反がある場合、フローラベルフィールドの上位オクテットを表す ICMP パラメータ問題メッセージ、コード 0 でソースに報告します (すなわち、IPv6 パケットのオフセット 1)。

制御プロトコル、ホップバイホップオプション、その他の手段で明示的なフロー情報が与えられていなくても、ルーターでは、場合によっては、任意のフローのフロー処理状態をセットアップできます。たとえば、未知のゼロ以外のフローラベルで特定のソースからパケットを受信すると、フローラベルがゼロである場合と同様に、ルーターではその IPv6 ヘッダーと必要な拡張ヘッダーを処理できます。この処理には、次のホップインタフェースの判定と、場合によってはホップバイホップオプションの更新、ルートヘッダーのポインタとアドレスの加算、あるいはトラフィッククラスフィールドにもとづくパケットのキューイングの方法の決定など、その他の動作が含まれることがあります。ルーターでは、ソースアドレスとフローラベルをキャッシュキーとして、これらの処理手順の結果を記憶してその情報をキャッシュに保存できます。同じソースアドレスとフローラベルで後続のパケットについては、先のパラグラフの条件によれば、フローの最初のパケット以後、どのフィールドも変更されていないと考えられるため、調べなくてもキャッシュ情報を参照するだけで処理できます。

トラフィッククラス

最初のノードや転送先のルーターは、IPv6 ヘッダーの 8 ビットトラフィッククラスフィールドを使用して、IPv6 パケットの異なるクラスまたは優先順位の特定と識別を行なうことができます。

トラフィッククラスフィールドには、以下の一般的な条件が適用されます。

- 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、上位プロトコルに対して、トラフィッククラスのビットの値を、上位プロトコルで生成されたパケットで供給する手段を提供する必要があります。すべての 8 ビットについて、デフォルトの値は必ず 0 になるようにします。
- 一部またはすべてのトラフィッククラスビットの固有の使用方法をサポートするノードは、それぞれの方法に従って生成、転送、または受信するパケット内のビットの値を変更することができます。ノードはトラフィッククラスフィールド内の、固有の使用方法をサポートしないすべてのビットを無視し、変更しないようにしなければなりません。

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、アプリケーション層の下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。

最初のオプションと、IPv6 認証ヘッダーという拡張ヘッダーは、IPv6 データグラムに機密機構なしの認証と一貫性を提供します。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が、ワールドワイドインターネットで相互運用性を確保するにはキー付き MD5 を使用する必要があります。これでホストマスカレード侵害など、主なネットワーク侵害が避けられます。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。インターネット層に配置することで、現在は充分に保護されていない上位層プロトコルとサービスでホスト送信認証が可能になります。

2 番めのオプションである、IPv6 カプセル化セキュリティヘッダーと呼ばれる拡張ヘッダーは、IPv6 データグラムに一貫性と機密性を与えます。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。グローバルネットワークで相互運用性を活かすため、DES CBC は、IPv6 カプセル化セキュリティヘッダーの標準アルゴリズムとして使用されています。

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。IPsecII の概要については、第 18 章を参照してください。IPsec の実装方法については、第 19 章を参照してください。

第 15 章

IPv4 から IPv6 への移行

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あとも、IPv4 だけをサポートしているノード (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 への移行方法と、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。

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 IPv6 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、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のために IPv4 の機能を残す。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到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 の両方を組み込み、残りスタックが同一である状況を表します。この場合、同じ伝送プロトコル (TCP、UDP など) と同じアプリケーションが IPv4 と IPv6 の両方で実行します。

図 15–1 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。

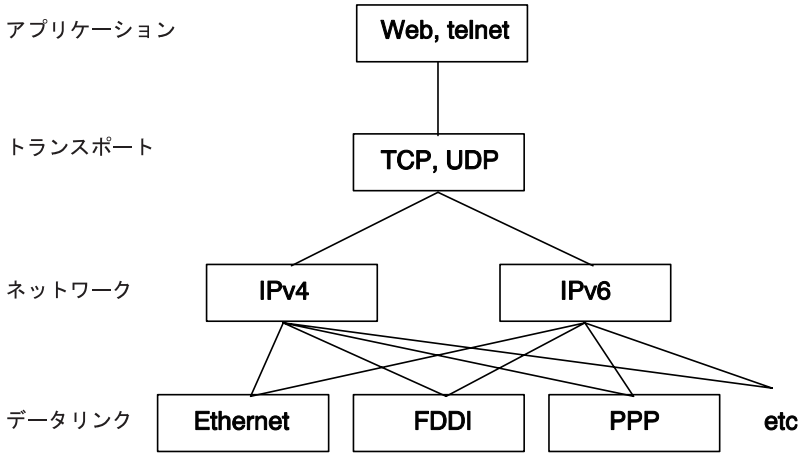


図 15–1　デュアルスタックプロトコル

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あとのノードからも IPv4 で常に IPv4 専用ノードと相互運用できます。そのため、IPv4 からデュアルスタックにアップグレードしても失われるものはありません。

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。そのため、ネームサービスでどんな情報を伝達するかを制御する必要があります。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 アドレスを定義すると、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。

ただし、IPv6 アドレスをネームサービスに指定した場合も、ネームサービスから情報を得たすべてのノードが IPv6 でそのノードにアクセスできます。たとえば、NIS に IPv6 アドレスを指定すると、その IPv6 ホストはその NIS ドメインに所属するすべての IPv6 とデュアルノードからアクセスできます。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 の場合も同様であり、たとえばメールの配信と HTTP プロキシの操作は、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。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、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、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。

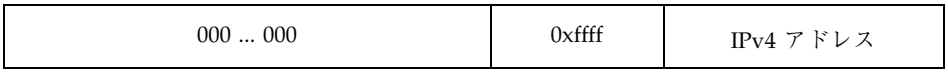
ネームサービスのアクセスに使用するプロトコル (DNS、NIS、NIS+ など) は、ネームサービスで検索できるアドレスタイプに依存しません。このネームサービスサポートでは、デュアルスタックとの組み合わせにより、デュアルノードから、IPv4 専用ノードとの通信では IPv4、IPv6 ノードとの通信では宛先までの IPv6 ルートがあれば IPv6 を使用できます。

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

- IPv4 マップアドレス



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 機構や静的構成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。IPv4 アドレスを 128 ビットマップアドレスで表現して 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すると、アプリケーションで便利です。ただ、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 ノード間のあるすべてのノードで 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 15–2 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。

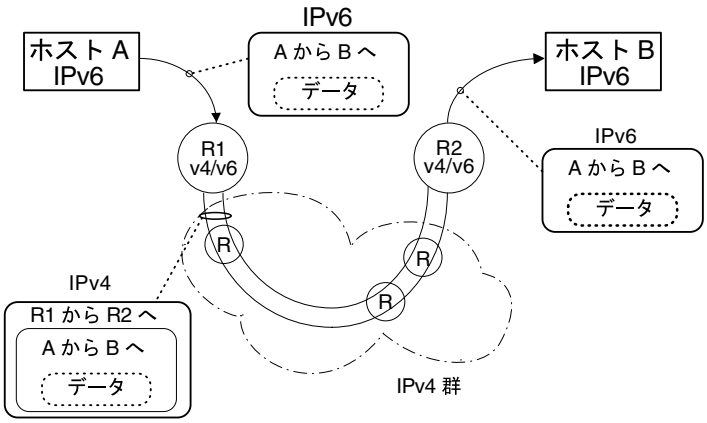


図 15–2　トンネル機構

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、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。アプリケーション側で変更が必要な API (ソケットなど) を使用する場合や、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アプリケーションでは、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 を使用しません。どちらの場合も、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併行して既存の 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。その場合、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 専用サーバーだけでなく、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 15–1 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 15–1 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 15–1 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処理はエラーになりますが、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。

サイト移行シナリオ

サイトや ISP では、それぞれ事情も異なれば移行段階の手順も異なります。ここでは、サイト移行シナリオの例をいくつか紹介します。

サイト移行では最初に、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、383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。

- IPv4 互換アドレスと自動トンネルで、ホストを 1 つずつアップグレードします。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た方法であり、IPv6 の機能のすべてが利用できるわけではありません。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、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に使用します。
- ルーター間に設定したトンネルを使用して、サブネットを 1 つずつアップグレードします。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、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。また、IPv6 専用ノード (または IPv4 アドレスのないデュアルノード) と IPv4 専用ノードの間では先に説明した方法は相互運用ができませんでした。ほとんどの実装はデュアルにできますが (デュアルかどうかはコードのメモリーフットプリントだけの問題)、現実には、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 アドレス領域があるかどうかが問題です。

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 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 互換か IPv4 マップアドレスであること)、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。変換時に情報が失われないよう暗号化されているパケットや、ソースルーティングなど使用頻度の低い機能を除外することで、これらトランスレータのサポートが IPv6 プロトコルに組み込まれています。

第 16 章

IPv6 の管理

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。トンネル、ルーター発見、ステートレスアドレス自動設定を実装するために新しいモジュールが追加されました。この章では、IPv6 の Solaris 実装に伴う概念について説明します。

Solaris IPv6 実装の概要

IPv4 から IPv6 への移行で、IPv6 では、IPv6 パケットにカプセル化された IPv6 パケットと同じく、IPv6 パケットを IPv4 パケット内にカプセル化するメソッドが指定されます。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 ユーティリティでプラムされ、設定されます。これ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、IPv6 スタックが作成され、新しいパラメータがサポートされました。これらについてはこの章で、あとから説明します。

ルーター発見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` など物理的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、380ページの「IPv6 の Solaris トンネルインタフェース」を参照)
Module	プラム時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	アタッチの物理的ポイント

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るため、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。ノードは近傍探索デーモン `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 (IPv6 では一般的ではありませんが)、`ifconfig` ユーティリティのコマンドインタフェースを使用して追加できます。その結果、ホスト名のアドレスが `/etc/hostname6.`*interface* (または `/etc/hostname.`*interface*) に保存され、内容が `ifconfig` に伝わります。

この場合、ファイルのエントリは、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 アドレスだけです。たとえば、`smc0` が `ahaggar` というマシンのプライマリネットワークインタフェースだとします。その `/etc/hostname6.*` ファイル名は `/etc/hostname6.smc0` となり、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、インタフェース数と、ルーティングデーモンとパケットの送信を開始するための `/etc/inet/ndpd.conf` ファイルの有無を調べます (390ページの「Solaris IPv6 ルーターの設定方法」を参照)。

ifconfig ユーティリティに対する IPv6 拡張機能

ifconfig ユーティリティは、トンネルモジュール同様に、IPv6 インタフェースをプラムできるよう変更されました。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、394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
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、388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 コマンドでは、hme0:3 論理インタフェースが 1234::5678/64 IPv6 アドレスに作成され、up オプションで有効になり、状態が報告され、無効になり、インタフェースが削除されます。

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 コマンドでは、物理的インタフェース名に関連付けられているデバイスが開き、デバイスを使用できるよう TCP/IP に必要なストリームがセットアップされ、その状態が報告され、トンネルソースと宛先アドレスが設定され、設定後の新しい状態が報告されます。

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
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 6–1 に示すマシン 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。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。 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、プライマリ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マシン 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 の設定では、/etc/hostname6.le0 と /etc/hostname.le1 に対するインタフェースが必要です。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.ridpd – 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 16–2 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	インタフェース関連情報ごとに印刷する
–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

/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) のパラメータセットは、in.ndpd の動作を制御します。

/etc/inet/ndpd.conf が存在すると構文解析され、ノードをルーターに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直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* ファイルはノード別に管理され、定期的に更新される状態ファイルであるため、処理が失敗して再起動したときはルーターがなくても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、先のルーター通知で得られた情報とともに保存されています。

注 - 状態ファイルの内容は変更する必要はありません。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、テンプレート (コメント行) 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
```

(続く)

```
#AdvManagedFlag
#AdvOtherConfigFlag
#AdvLinkMTU
#AdvReachableTime
#AdvRetransTimer
#AdvCurHopLimit
#AdvDefaultLifetime
#
# Per Prefix:  AdvPrefixList configuration varialbes
#
#
#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 RIPng ルーティングプロトコルを実装します。RIPng は、Bellman-Ford 距離ベクトルアルゴリズムに基づく IPv4 ルーティングプロトコルで広く使用されてきた RIP の IPv6 等価を定義します。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
-s	ルーターとしての機能に関係なく、ルーティング情報が強制される
-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 有効化サーバーでは、対応するクライアントで利用している内容に応じて、IPv4 アドレスか IPv6 アドレスを処理できます。/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 有効化で /etc/inet/inetd.conf ファイルにサービスを指定するには、*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。サービスに tcp6 または udp6 の *proto* 値を指定すると、inetd は所定のデーモン 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
```

AF_INET (IPv4 専用) ソケットまたは AF_INET6 (IPv6 と IPv4) ソケットを inetd から継承できるよう IPv6 有効化サーバーを書き込むと、サービスの *proto* 値が 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、1 つは *proto* で tcp として、もう 1 つは *proto* で 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、『ネットワークインタフェース』を参照してください。

Solaris バンドルサーバーには、いずれも *proto* 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 バンドルサーバー用の 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
```

以上のユーティリティの *server_pathname* として TCP ラッパー (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) を指定するには、TCP ラッパーが IPv6 対応であることが条件です。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 に含まれるユーティリティを別の実装と交換する場合、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、*proto* 値を tcp か udp に指定します。

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、tcp や udp は指定しないでください。

ソケットを使用する IPv6 有効化サーバーの記述については、『ネットワークインタフェース』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、370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が選択できます。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、396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、近傍探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、400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、401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。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。

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 を指定する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 - コマンド行引数の一部として使用される 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 (存在する場合) で設定した値を無効にします。

関連の操作については、399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 はアタッチメントの物理的ポイントです。

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。これは、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 (IPv4 の上に IPv6)、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、インタフェースがプラムされたあとに ifconfig(1M) に渡り、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。2 つのインタフェースが設定され、ip.tun0 インタフェースがコメントで記述され、論理インタフェース (ip.tun0:1) には、addif コマンドによってソース IPv6 アドレスと宛先 IPv6 アドレスが与えられます。

すでに述べたとおり、システムを複数ユーザーとして起動すると、これらの構成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
```

次の構文で構成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が、トンネルに対して 1 つ以上のプレフィックスを通知している IPv6 ルーターの場合、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、必要なのは `tsrc` と `tdst` だけです。

場合によっては、所定のトンネルについて、特定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、構成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` の特定の情報については、`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、361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、402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 リリースで導入された IPv6 の実装による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、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+ データを検索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 アドレスと IPv6 アドレスの両方が保存されます。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、IPv4 アドレスを /etc/inet/hosts に保存するのと同じ方法で IPv6 アドレスを保存すると便利なこともあります。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、90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 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。IPv4 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
```

(続く)

```
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` マップと `hosts.byaddr` マップは、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の便宜上変更されていません。

IPv6 の NIS+ 拡張機能

`ipnodes.org_dir` という名前の新しいテーブルが NIS+ 用に追加されました。IPv4 アドレスと IPv6 アドレスの両方を保存します。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。既存の 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 アドレス用に予約されています。各ニブルには `ip6.int` が追加されて、対応する 16 進 ASCII 値に変換されます。

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 ルックアップをサポートするように変更されました。ipnodes 行を /etc/nsswitch.conf ファイルに追加した結果、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。次に例を示します。

```
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。

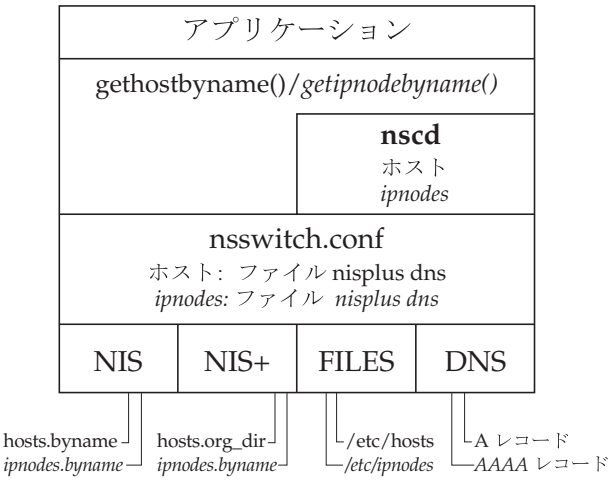


図 16–1　`nsswitch.conf` とネームサービスの関係

命名サービスの詳細については、『*Solaris* ネーミングの設定と構成』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、384ページの「IPv6 の NIS 拡張機能」、384ページの「IPv6 の NIS+ 拡張機能」、および 384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、404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 IPv6 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。

第 17 章

IPv6 の実装

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、IPv6 の追加をコマンドによって診断する方法、IPv6 ネームサービス情報の表示方法について説明します。

次に、この章で紹介する手順説明を一覧で示します。

関連情報	掲載章
IPv6 の概要	第 14 章
IPv4 から IPv6 への移行	第 15 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ための作業マップ

表 17–1 IPv6 ノードを有効にするための作業マップ

作業	説明	操作方法の掲載個所
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)	389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	390ページの「Solaris IPv6 ルーターの設定方法」

表 17–1　IPv6 ノードを有効にするための作業マップ　続く

作業	説明	操作方法の掲載個所
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	391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	392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、ルーター発見パケットが発信されて、ルーターはプレフィックスを応答し、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、主なネットワーキングデーモンも IPv6 モードで再スタートします。

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。

   a. **NIS** と **NIS+** については、391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。

   b. **DNS** については、392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. システムをリブートします。

注 - 近傍探索 (in.ndpd) からホストにサブネットアドレスプレフィックスがリレーされます。RIPng ルーティングプロトコル (in.ripngd) も自動的に実行されます。

例 – ndpd.conf ルーター構成ファイル

```
# 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の便宜上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。この処理は自動では行われません。概要については、383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+ テーブルの操作方法の詳細については、『*Solaris* ネーミングの管理』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 ipnodes.byname と ipnodes.byaddr という 2 つの新しいマップが追加されました。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。既存の 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の便宜上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-aware ツールは、新しい NIS マップと NIS+ マップおよびテーブルを占有使用します。

▼ 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

次のコマンドは編集されて、IPv6 の Solaris 実装に対応します。

- `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、370ページの「`ifconfig` ユーティリティに対する IPv6 拡張機能」と 378 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 監視の作業マップ

表 17–2 IPv6 監視の作業マップ

作業	説明	操作方法の掲載個所
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	394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 IPv6 アドレスファミリー、netstat コマンドによるインタフェース別統計 - IPv6/ICMPv6 カウンタ	396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	inet_type という名のファイルの作成と、そのファイル内の DEFAULT_IP 変数の設定による 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	399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 IPv6 パケットの表示	400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	401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	401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 inet 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
```

(続く)

```
      *.*                  *.*                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 – IPv4 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
```

(続く)

```
        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

/etc/default ディレクトリで inet_type という名のファイルを作成し、DEFAULT_IP 変数の値を指定すれば、netstat コマンドと ifconfig コマンドの出力は制御できます。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、次のどれかのエントリを作成します。
   a. **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
```

   b. **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
```

```
DEFAULT_IP=IP_VERSION6
```

注 - ifconfig の –4 フラグと –6 フラグ、および netstat の –f フラグの設定は、inet_type ファイルに設定された値 (存在する場合) を優先します。

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_VERSION 変数を inet_type ファイルで設定する場合

```
% 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

```
% 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. スーパーユーザーでログインします。
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、380ページの「IPv6 の Solaris トンネルインタフェース」と 361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. 次の手順に従って、ファイル `/etc/hostname6.ip.tun`***n*** **(*n* は、0、1、2 など)** とエントリを作成します。

   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は自動的に設定されるため、設定する必要はありません。

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で通知するようにルーターを設定する必要もあります (404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 構成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 — 手動でアドレスを設定するための IPv6 構成ファイルのエントリ

次に、グローバルソース、サイトローカルソース、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。

```
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. 次の手順でファイル /etc/inet/ndpd.conf を編集し、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<i>n</i> AdvSendAdvertisements 1
```

   b. 必要に応じてプレフィックスを追加します。

```
prefix <i>interface-address</i> ip.tun<i>n</i>
```

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 17–3 IPv6 ネームサービス情報を表示する作業マップ

作業	説明	操作方法の掲載個所
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	405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 コマンドでパラメータを q=PTR と表示に設定し、DNS IPv6 PTR レコードを表示する	406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	407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	407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	408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. 終了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 コマンドで DNS IPv6 PTR レコードを表示するには、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. **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. 終了を入力して、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 17–1　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 17–1　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 17–2　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 18 章

IPsec の概要

IP セキュリティアーキテクチャー (IPsec) は、IP データグラムを保護します。保護には機密、データの強い完全性、シーケンスの部分的完全性 (再生保護)、データ認証があります。IPsec は、IP プロセス内部で実行され、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。IPsec は、ネットワークトラフィックの保護に有効なツールですが、セキュリティ問題を解消できるわけではありません。

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。

図 18–1 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。

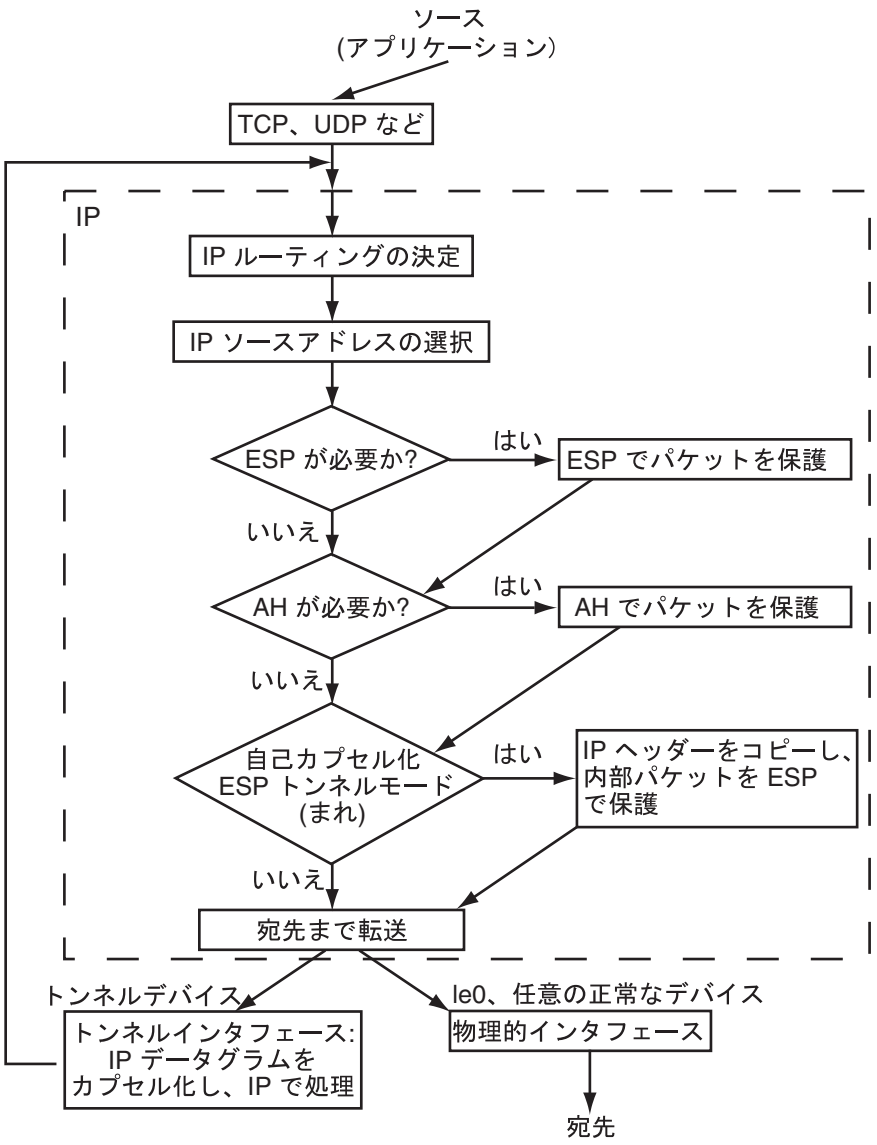


図 18–1　出力パケットプロセスに適用された IPsec

図 18–2 は、IPsec 入力プロセスを示したものです。

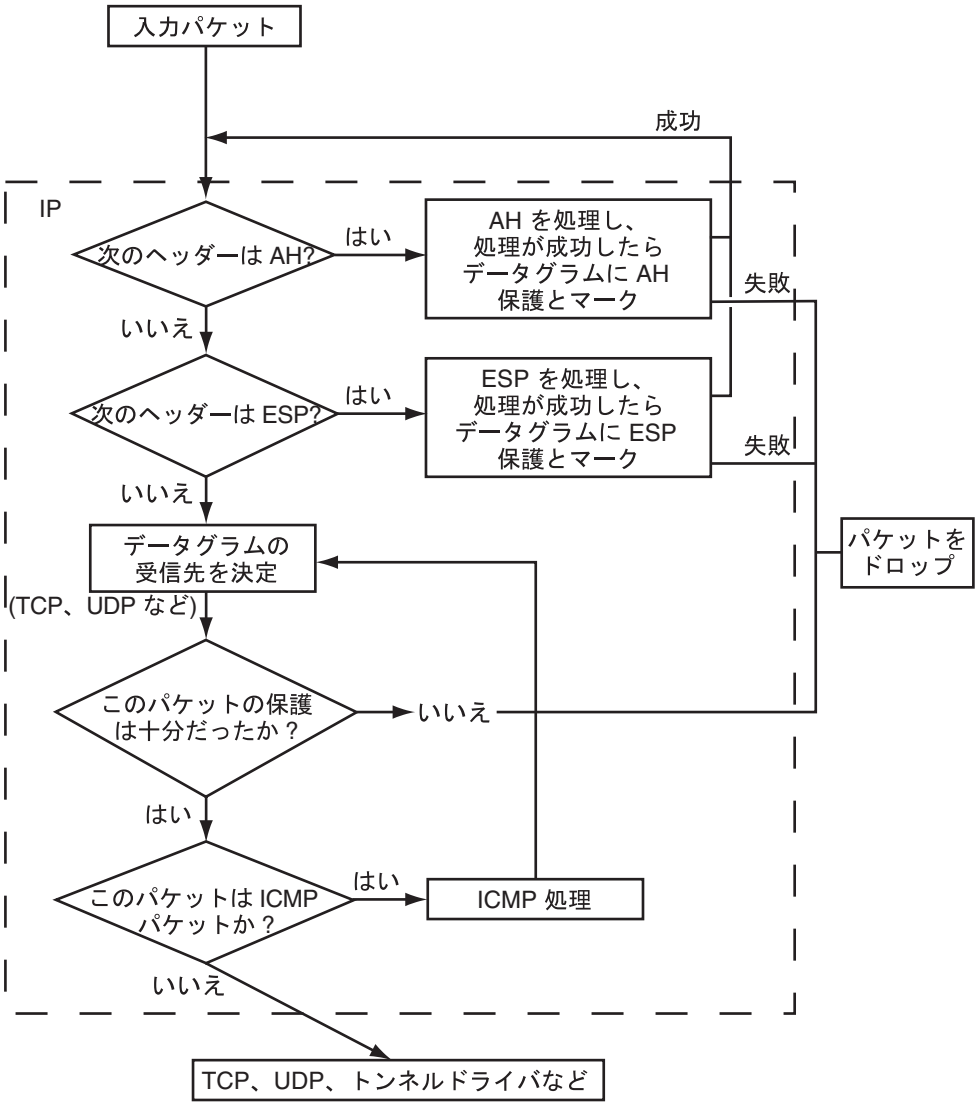


図 18–2　IPsec を入力パケットプロセスに適用

セキュリティアソシエーション

セキュリティアソシエーション (SA) では、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します。互いに通信し合う 2 つのシステムが安全に通信するには、少なくとも 2 つの SA が必要です。`pf_key(7P)` インタフェースは、セキュリティアソシエーションを管理します。IPsec は自動 SA 管理をサポートしていませんが、`ipseckey(1M)` をコマンド行フロントエンドとして使用できます。AH または ESP、宛先 IP アドレス、セキュリティパラメータインデックス (SPI) は、IPsec SA を識別します。任意の 32 ビット値のセキュリティパラメータインデックスは、AH パケットまたは ESP パケットで転送されます。SPI が保護パケットのどこにあるかについては、`ipsecah(7P)` と `ipseesp(7P)` のマニュアルページを参照してください。

キー管理

セキュリティアソシエーションには、キー情報、アルゴリズムの選択、エンドポイントの識別、その他パラメータがあります。SA の管理をキー管理といいます。現在のところ、キー管理は手動で行います。

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、あるいはもう 1 つの 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 デバイス

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、将来の認証アルゴリズムが AH の先頭にロードできる他、`ndd(1M)` で AH を調整します。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 `autsha1(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、応答時のすべての問題が 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、見ようと思え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 データグラムの IP 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 だけで機密は守れますが、機密だけを守っても、応答侵害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 デバイス

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、将来の認証アルゴリズムが AH の先頭にロードできる他、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) と Triple-DES (3DES) 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。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、盗聴侵害の他、カットアンドペースト暗号化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。合衆国の輸出管理法の適用を受けるので、合衆国外で販売される SunOS の場合、ESP で得られる暗号化は、強度が低くなります。

認証アルゴリズムと暗号化アルゴリズム

IPsec では、次の 2 種類のアルゴリズムを使用します。

- 認証
- 暗号化

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。ダイジェストとキーのサイズについては、認証アルゴリズムのマニュアルページ (authmd5h(7M) や authsha1(7M) のマニュアルページなど) を参照してください。

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。ブロックサイズとキーサイズについては、暗号化アルゴリズムのマニュアルページ (encrdes(7M) や encr3des(7M) のマニュアルページなど) を参照してください。

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の決定の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 ヘッダーのあとに、ESP または AH を呼び出してデータグラムを保護するとき、これをトランスポートモードといいます。たとえば、パケットが次のように始まる場合です。

IP ヘッダー	TCP ヘッダー	

トランスポートモードで、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。

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、これをトンネルモードといいます。AH はその前にあるほとんどの IP を保護するため、トンネルモードは通常は ESP だけで実行します。先の例のデータグラムは、次のようにトンネルモードで保護されます。

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-aware ネットワークプログラムで ESP 付き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。これは、ESP で保護する必要のある IP ヘッダーオプションの場合に行われます。

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-in-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。

IPsec トンネルのトンネルモジュール

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、380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限り、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を使用可能にする

IPsec で、仮想プライベートネットワーク (VPN) 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 18–3 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。

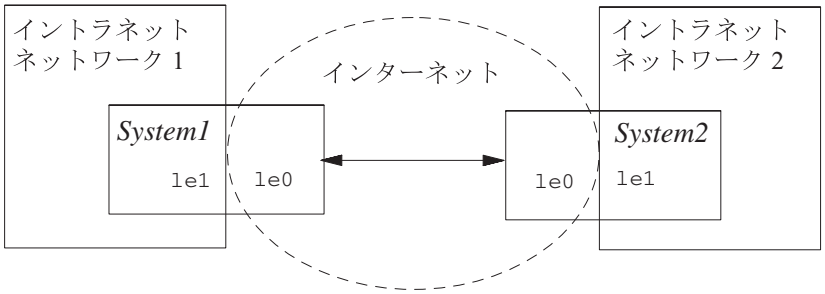


図 18–3　仮想プライベートネットワーク

セットアップ手順については、431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 の管理

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。IPsec の管理手順については、第 19 章を参照してください。

IPsec 初期化構成ファイル

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、個々の IPsec エントリを利用して IPsec 初期化構成ファイルを作成する必要があります。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。

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 1.4 99/04/28 SMI"
#
# Copyright (c) 1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# This file should be copied to /etc/inet/ipsecinit.conf to enable IPsec.
# 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,
#
```

(続く)

```
#  {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 to/from the host with ESP using DES and
# MD5.  Also:
#
#  {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.
#
#
# 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 on-subnet (i.e. reachable without a default route).
#
#  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.
#
```

グローバルポリシーの設定

ホストの IPsec ポリシーを設定するには、ipsecconf(1M) コマンドを使用します。ポリシーを設定すると、IPsec は、ホストに出入りするときにすべての出力データグラムと入力データグラムに対してポリシー検査を適用します。エントリが見つからない場合、ポリシー検査は行われず、すべてのトラフィックが通過します。送信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。送信パケットを保護する方法については、iconfig(1M) と tun(7M) のマニュアルページを参照してください。iconfig コマンドは、/etc/inet/ipsecpolicy.conf ファイルからポリシーエントリを削除するときや、既存の設定を終了するときに使用します。

このコマンドはスーパーユーザーでないと呼び出せません。エントリごとに保護されるトラフィックは 1 方向、すなわち出力か入力のどちらかです。そのため、両方向のトラフィックを保護するには、各方向ごとにエントリを設定する必要があります。

システムに設定されたポリシーを確認するには、引数なしでこのコマンドを実行します。このコマンドによって番号があとについたインデックスとともにエントリが

表示されます。インデックスに –d オプションを指定すると、システム内の指定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。

IPsec では、リブートするとポリシーエントリが削除されます。そのため、システムのリブートのたびにポリシーエントリを追加する必要があります。起動プロセスの初期にポリシーを設定するには、/etc/inet/ipsecinit.conf ファイルでポリシーをセットアップし、inetinit 起動スクリプトで読み取らせます。

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、設定ポリシーも変更される可能性があります。そのため、/etc/inet/ipsecpolicy.conf ファイルのコピーをネットワークで送信してはいけません。

ポリシーは connect(3N) または accept(3N) を発行した TCP/UDP ソケットにラッチされます。新しいポリシーエントリを追加しても、これらは変更されません。このラッチ機能は将来変更される可能性があるので、この機能に依存してはいけません。

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。ipseckey 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。

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では、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があり、そのあとに多くの (0 以上) 拡張メッセージが続きます。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、ベースメッセージ、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。

マニュアルキープログラム

ipsecesp(7P) ネットワークセキュリティサービスと ipsecah(7P) ネットワークセキュリティサービスでセキュリティアソシエーションデータベースを手動で操作するには、ipseckey(1M) コマンドを使用します。また、自動キー管理が無効な場合に、通信パーティ間のセキュリティアソシエーションをセットアップするときも、ipseckey コマンドを使用します。

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 コマンドでは、微妙な暗号キー情報を扱います。特権ユーザーは暗号キー情報を入力できます。場合によっては、不正にこの情報をアクセスして IPsec トラフィックを損なうことも可能です。ipseckey コマンドを使用するときは、次のことに注意してください。

1. 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)。

- 接続されている場合は、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(1M)` 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侵害には無防備です。

2. 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` 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。

既存のユーティリティに対する IPsec 拡張機能

ifconfig

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、指定したアルゴリズムで、トンネルに IPsec AH を使用できます。フォーマットは次のとおりです。

```
auth_algs authentication_algorithm
```

アルゴリズムは番号かアルゴリズム名です。パラメータ *any* も使用できます。その場合、特定のアルゴリズム設定は指定されません。IPsec トンネル属性は、すべての

同じコマンド行で指定してください。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
```

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。フォーマットは次のとおりです。

```
encr_auth_algs authentication_algorithm
```

このアルゴリズムの場合、番号かアルゴリズム名を指定できます。パラメータ *any* も使用できます。この場合、特定のアルゴリズム設定は指定されません。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。

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。フォーマットは次のとおりです。

```
encr_auth_algs encryption_algorithm
```

このアルゴリズムの場合、番号か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、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 *null* になります。

snoop(1M)

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 19 章

IPsec の実装

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。

IPsec の概要については、第 18 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に例に応じた説明があります。

IPsec 実装の作業マップ

表 19–1 IPsec 実装の作業マップ

作業	説明	操作方法の掲載個所
システム間のトラフィックの保護	/etc/hosts ファイルに対するアドレスの追加、/etc/inet/ipsecinit.conf ファイルの編集、セキュリティアソシエーションの追加、ipsecinit.conf ファイルの呼び出し	426ページの「2 つのシステム間のトラフィックの保護」
IPsec ポリシーによる Web サーバーの保護	ipsecinit.conf ファイルの編集と呼び出しによる保護トラフィックだけを使用可能	429ページの「IPsec ポリシーによる Web サーバーの保護」
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	431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新しいセキュリティアソシエーションの入力	437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。

▼ 2 つのシステム間のトラフィックの保護

この手順を行う前に、任意のアルゴリズムで AH 保護をすでに呼び出しているものとします。また、セキュリティアソシエーションを共有し (すなわち、2 システムの保護を必要とするのは 1 組の SA だけである)、各システムには IP アドレスが 1 つしかないという前提です。

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、セキュリティ的に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/hosts ファイルに追加します。次のコマンドを使用します。

   a. システム **1** では次のようになります。

```
# echo "system2_addr system2_name" >> /etc/hosts
```

   b. システム **2** では次のようになります。

```
# echo "system1_addr system1_name" >> /etc/hosts
```

   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ようなことなくシステム名を使用できます。

3. 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。

   a. システム **1** では次のようになります。

```
{saddr system1_name daddr system2_name} apply {auth_algs any sa shared}
{saddr system2_name daddr system1_name} permit {auth_algs any}
```

   b. システム **2** では次のようになります。

```
{saddr system2_name daddr system1_name} apply {auth_algs any sa shared}
{saddr system1_name daddr system2_name} permit {auth_algs any}
```

4. 次の操作でセキュリティアソシエーションを追加します。

a. たとえば MyKeyfile のように、選択したファイル名で、システムごとに読み取り専用 **(600** のアクセス権**)** ***keyfile*** を作成し、このファイルに次の行を入力します。

```
add ah spi random-number dst system1_name authalg algorithm_name \
      authkey random-hex-string-of-algorithm-specified-length
add ah spi random-number dst system2_name authalg algorithm_name \
      authkey random-hex-string-of-algorithm-specified-length
```

b. 次のコマンドを入力して、システムごとにセキュリティアソシエーションを有効にします。

```
# ipseckey -f keyfile
```

5. システムごとに次のどれかの操作をします。

a. 次のコマンドを入力して、ipsecinit.conf ファイルを呼び出します。

```
# ipsecconf -a /etc/inet/ipsecinit.conf
```

b. または両方のシステムをリブートします。

両方のシステムをリブートする場合、起動スクリプトにまず次のコマンド (手順 4 で使用) を入力します。

```
ipseckey -f keyfile
```

そのためには、次の操作をします。

c. 次のコマンドを入力して ***keyfile*** 名を ipseckey に変更します。

```
# cp keyfile /etc/inet/ipseckey
```

d. 次のコマンドを入力して ipseckey ファイルを読み取り専用にします。

```
# chmod 600 /etc/inet/ipseckey
```

e. 次のコードを組み込んだ起動スクリプト /etc/rc3.d/s99ipsec_setup を作成します。

```
if [ -f /etc/inet/ipseckeys -a -f /etc/inet/ipsecinit.conf ]; then
     /usr/sbin/ipseckey -f /etc/inet/ipseckeys
fi
```

以後のリブートでは、起動プロセスの終了前に、/etc/inet/ipseckeys ファイルが読み取られます。キーを変更する場合は、両方のシステムでファイルが変更されていることを確認してください。

▼ IPsec ポリシーによる Web サーバーの保護

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、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、セキュリティ的に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 名検査をしないときは、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 MyIPsecInitFile 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, so as not to require 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をアクセスするようになります。ただし先の手順で省略するようにリストしたトラフィックは例外です。

4. **2** つの操作のどちらかを実行します。

   a. 先の手順で作成したファイルを、/etc/inet/ipsecinit.conf にコピーし、次のコマンドでリブートします。

```
# cp filename /etc/inet/ipsecinit.conf
# reboot
```

   b. 作成したファイルを次のコマンドで呼び出します。

```
ipsecconf -a filename
```

注 - このファイルにはネームサービスが必要ないためにこの操作が可能です。また、ipsecconf を呼び出しても、既存の TCP 接続には IPsec ポリシーが適用されません。そのため、ipsecconf コマンドを実行すると警告が表示されます。

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 ipseckey(1M) でセキュリティアソシエーションを追加して、IPsec をリモートシステムで有効にしないと機能しません。

▼ 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し、そのネットワーク間のトラフィックを IPsec で保護する操作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、417ページの「仮想プライベートネットワークを使用可能にする」を参照してください。

1. システムコンソールで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、セキュリティ的に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 0
```

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. 次の手順で、必要に応じて **Solaris** システム上のほとんどの **(**すべてでなければ**)** 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か (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)、429ページの「IPsec ポリシーによる Web サーバーの保護」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 の編集がまだの場合は、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. 次の操作で、各システムごとに **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。

   a. 次のコマンドを入力します。

```
# ipseckey
```

   ipseckey コマンドモードが有効になります。

   b. ipseckey コマンドモードプロンプトで、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 very-random-hex-string-of-32-characters \
encrkey very-random-hex-string-of-16-characters
```

   c. **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。

   d. ipseckey コマンドモードプロンプトで、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 very-random-hex-string-of-32-characters \
encrkey very-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. ipseckey コマンドモードプロンプトで、**Ctrl-D** または **Exit** キーを入力してこのモードを終了します。

6. 次の手順で保護トンネル 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
```

保護トンネルが構築され、IP からみたもう 1 つの物理的インタフェースが追加されます。

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` 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実行して、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実行して、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。

    これで、in.rdisc がリブート時に開始しなくなります。

b. 次のコマンドを入力します。

```
# touch /etc/notrouter
```

起動シーケンスの初期にルーティングが打ち切られ、防備が強化されます。

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
```

d. /etc/rc3.d/S99ipsec_setup ファイルを編集して、最後に **if/then** 文に次の行を追加します。

```
ndd -set /dev/ip le1:ip_forwarding 1
ndd -set /dev/ip ip.tun0:ip_forwarding 1
ifconfig le0 private
in.routed
```

ファイルは次のようになります。

```
if [ -f /etc/inet/ipseckeys -a -f /etc/inet/ipsecinit.conf ]; then
     /usr/sbin/ipseckey -f /etc/inet/ipseckeys
     ndd -set /dev/ip le1:ip_forwarding 1
     ndd -set /dev/ip ip.tun0:ip_forwarding 1
     ifconfig le0 private
     in.routed
fi
```

11. システムごとに、次のコマンドを実行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 2 で呼び出すセキュリティアソシエーションは定期的に新しい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、次のように操作します。

▼ 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、セキュリティ的に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、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。
   a. 次のコマンドを入力します。

```
# ipseckey
```

   ipseckey コマンドモードが有効になります。

   b. ipseckey コマンドモードプロンプトで、次のコマンドを入力します。

```
> flush
```

3. 431ページの「仮想プライベートネットワークの構築」の手順 **5** を実行し、**SPI** とキーの値を変更して新しいセキュリティアソシエーションを設定します。

第 **20** 章

モデム関連ネットワークサービスのトピック

第 21 章	PPP の概要情報
第 22 章	PPP の計画情報
第 23 章	PPP のセットアップと障害追跡 (トラブルシューティング) の手順説明
第 24 章	PPP データベースとファイルの操作のリファレンス情報
第 25 章	UUCP の背景情報
第 26 章	UUCP のセットアップと障害追跡の手順説明
第 27 章	UUCP データベースファイルのリファレンス資料、UUCP 構成ファイル、UUCP シェルスクリプト、UUCP 障害追跡情報

第 21 章

PPP の概要

この章では、TCP/IP プロトコル群に含まれるデータリンクプロトコルの 1 つである Solaris ポイントツーポイントプロトコル (PPP) の概要を示します。仕様、最も典型的な PPP 構成の紹介、および PPP に関連した用語の定義について説明します。

Solaris PPP の概要

PPP を用いると、モデムと電話回線を使用して、物理的に離れた場所にあるコンピュータとネットワークを接続することができます。ユーザーは PPP を使用して、自宅や職場から、所属するサイトのネットワークに接続できます。また、PPP ソフトウェア、モデム、電話回線を組み合わせて、別々の場所にあるネットワーク同士を結ぶルーターとして使用することもできます。PPP は、このようなマシンとネットワークを構成するための方法を提供します。この章ではその方法を紹介します。

Solaris PPP の仕様

Solaris PPP は、標準化されたデータリンクレベルの PPP の非同期実装の 1 つです。PPP は TCP/IP プロトコル群に含まれているもので、多くのルーターシステムのベンダーや端末集線装置から提供されています。Solaris PPP には標準化されたカプセル化プロトコルが組み込まれているので、ネットワーク層プロトコルにとってデータグラムの転送が透過的になります。

Solaris PPP の主な特性には次のものがあります。

- RFC 1331 で定義されているインターネットポイントツーポイントプロトコルを実装
- CRC を使用したエラー検出機能を提供
- 全二重伝送をサポート

このプロトコルの主な機能には次のものがあります。

- IP が非同期シリアル回線を介してパケットを転送するためのインタフェース
- 要求時の接続確立
- 構成可能オプションのネゴシエーション
- 接続の切断 (自動ハングアップ)

PPP が使用する伝送機能

PPP は、Solaris ソフトウェアを実行するほとんどのマシンに備わっている CPU シリアルポートを使用した、RS-232-C(V.24) インタフェースをサポートします。さらに、PPP は、Solaris ソフトウェアを実行するマシンの製造元の多くが提供またはサポートしている、オプションの非同期シリアルポートでも動作します。PPP は、使用するマシンのシリアルポートで使用可能な最大のデータ速度をサポートします。マシンのシリアルハードウェアがサポートしている速度については、コンピュータシステムの製造元にお問い合わせください。

規格への適合性

PPP と、Solaris ソフトウェアに組み込まれているルーティング機能は、業界標準の規格に従って動作します。この規格は次のような機能をサポートしています。

- IP データグラムを転送する
- 転送するパケットを IP 互換にネットワーク化されたシステムから受け取る

- ローカルエリアネットワークメディア、たとえばEthernet、トークンリング、FDDI などを使用して IP 互換にネットワーク化されたシステムにパケットを配送する
- 標準化されたルーティングプロトコルを使用しているため、ユーザーは、多数の製造元が提供する PPP プロトコルをサポートする装置との間でパケットを交換できる

PPP ネットワークインタフェース

PPP を用いると、モデムなどのような非同期デバイスをネットワークインタフェースとして使用できるようになります。Solaris PPP では、2 つの仮想ネットワークインタフェース `ipdptp`*n* と `ipd`*n* を構成できます (*n* はインタフェースに割り当てるデバイス番号です)。

PPP ネットワークインタフェースは、仮想ネットワークインタフェースとみなされます。なぜなら、Ethernet インタフェースなどのようにネットワークハードウェアを含んでいないからです。さらに、PPP ネットワークインタフェースは特定のシリアルポートに関連付けられるものでもありません。PPP ネットワークインタフェースは、物理ネットワークインタフェースとともに `/devices` ディレクトリに入っています (物理ネットワークインタフェースについては、57ページの「ネットワークインタフェース」を参照してください)。

使用するネットワークインタフェースの種類は、設定したい PPP 通信リンクによって異なります。`ipdptp` インタフェースは、ポイントツーポイントリンクをサポートしています。`ipd` インタフェースは、ポイントツーマルチポイントリンク (「マルチポイントリンク」と呼ばれる) をサポートしています。

PPP によるネットワークの拡張

この節では、PPP に関連する通信の概念を紹介します。また、最も一般的な PPP 構成についても説明します。

ポイントツーポイント通信リンク

Solaris PPP の最も一般的な使用目的は、ポイントツーポイント通信リンクを設定することです。一般的なポイントツーポイント通信構成は、2 つのエンドポイントを通信リンクで接続したものです。この一般構成では、エンドポイントシステムはコンピュータでも端末でもよく、切り離された状態でも、ネットワークに物理的に接続していてもかまいません。通信リンクという用語は、2 つのエンドポイントシステムを接続するハードウェアとソフトウェアを指します。図 21–1 にこの概念を示します。

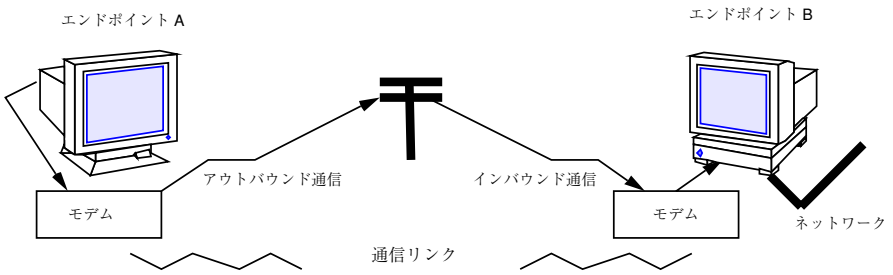


図 21–1　基本的なポイントツーポイントリンク

ダイヤルアウト操作とアウトバウンド通信

一方のエンドポイントが通信リンクの反対側のエンドポイントとの通信を望むとき、そのエンドポイントはダイヤルアウト操作を開始します。たとえば、エンドポイント B と通信する場合、その対等ホストであるエンドポイント A のユーザーは、`rlogin` *end-point-B* と入力します。すると、エンドポイント A は通信リンクを介してダイヤルアウトします。この場合、エンドポイント A はダイヤルアウトマシンとして機能することになります。`rlogin` コマンドは、モデムがエンドポイント B の電話番号をダイヤルすることを引き起こします。このコマンドが起動するエンドポイント A 動作と相手に渡す情報を、アウトバウンド通信といいます。

ダイヤルインとインバウンド通信

データが通信リンクを介してエンドポイント B に到達すると、エンドポイント B のシステムは着信データを受け取り、肯定応答信号をエンドポイント A に送って、通信を確立します。この場合、エンドポイント B は他のシステムからのダイヤルインを受け入れるので、ダイヤルインマシンとして機能することになります。通信の受信側に渡される情報と受信側が行う動作を、インバウンド通信といいます。

Solaris PPP がサポートするポイントツーポイント構成

Solaris PPP は、次の 4 つの種類のポイントツーポイント構成をサポートしています。

- ある場所のホストを、物理的に異なる場所にある別のホストに接続した構成(図 21–1)
- ダイヤルインサーバーとリモートホストを動的ポイントツーポイントリンクで接続した構成 (図 21–2)
- ネットワークを、物理的に離れた場所にある別のネットワークに接続した構成 (図 21–3)
- コンピュータを、離れた場所にあるネットワークに物理的に接続されているマルチポイントダイヤルインサーバーに接続したもの (図 21–4)

PPP リンクは、実質的にはローカルエリアネットワークと同じ種類の接続を提供しますが、ブロードキャスト機能だけはありません。次の各節では、上記の構成についてそれぞれ簡単に説明します。各構成の設定方法については、第 22 章で説明します。

2 つの単独ホストをポイントツーポイントリンクで接続

PPP を使用すると、異なる場所にある 2 つのスタンドアロンマシンを接続するポイントツーポイントリンクを設定できます。これにより、事実上、この 2 つのマシンだけからなるネットワークが作成されることになります。これはエンドポイントが 2 つしかなく、したがって最も単純なポイントツーポイント構成と言えます。図 21–1 に示した一般的な構成でも、このホストツーホスト構成が使用されています。

可搬マシンをダイヤルインサーバーに接続

従来は、標準的なダイヤル呼び出し接続または一時接続の場合、ネットワークに接続できるのは ASCII 端末だけでした。Solaris PPP を用いれば、個々のマシンを PPP リンクの 1 つのエンドポイントとして構成することによって、それらのマシンを物理的に離れた場所にあるネットワークの一部とすることができます。この可搬接続は、頻繁に旅行するユーザーや在宅勤務のユーザーを含むネットワークの場合に、特に便利です。

図 21–2 に示す可搬コンピュータは、それぞれネットワーク上のエンドポイントシステムへのポイントツーポイントリンクを持っています。ネットワーク上のエンドポイントシステムを、ダイヤルインサーバーと言います。

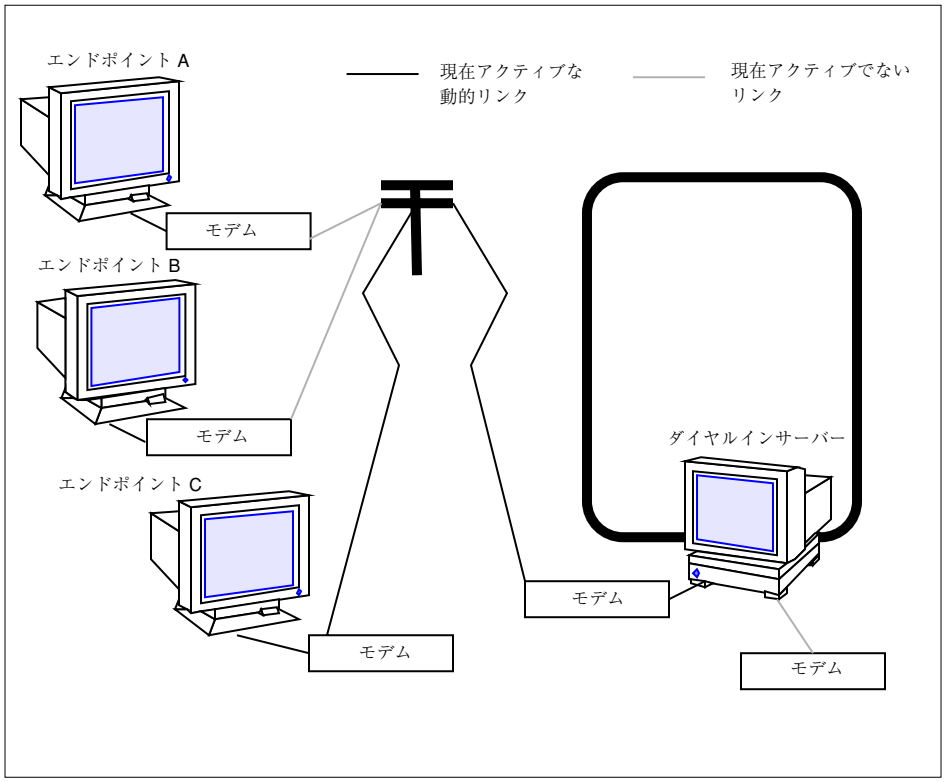


図 21–2　可搬コンピュータと動的リンクを持つダイヤルインサーバー

動的ポイントツーポイントリンクを持つダイヤルインサーバー

図 21–2 に示したネットワークのエンドポイントマシンは、動的ポイントツーポイントリンクを持つダイヤルインサーバーとして働きます。これをダイヤルインサーバーと呼ぶのは、リモートマシンがこのマシンにダイヤルインすることによってネットワークに入ることができるからです。サーバーは、あるマシンからダイヤルインの要求を受け取ると、必要時に提供するという方式でそのマシンに PPP リンクを割り当てます。

ダイヤルインサーバーは、動的ポイントツーポイントリンクまたはマルチポイントリンクを介してリモートホストと通信します。マルチポイントリンクについては、448ページの「マルチポイント通信リンク」で説明します。動的ポイントツーポイントには、ポイントツーポイント通信と同じ利点があります。つまり、リンク上で RIP を実行でき、ブロードキャストが使用可能になります。最も重要なのは、物理ネットワーク上の複数のマシンが、ダイヤルインサーバーとして機能すること

ができるという点です。これはバックアップサーバーを構成できることを意味し、したがってサーバーの重複が可能となり、管理が容易になります。図 21–2 の各マシンはネットワークエンドポイントとは直接通信できますが、互いに直接通信することはできません。ダイヤルインサーバーエンドポイントを仲介として、相互に情報を受け渡しする必要があります。

2 つのネットワークをポイントツーポイントリンクで接続

PPP を使用すると、2 つのネットワークをポイントツーポイントリンクで接続し、各ネットワーク上の 1 つのシステムをエンドポイントとして機能させることができます。これらのエンドポイントは、図 21–1 に示したのと実質的に同じ方法で、モデムと電話回線を使用して互いに通信します。ただし、この設定では、エンドポイント、モデム、PPP ソフトウェアは、各物理ネットワークのルーターとして働きます。この種類の構成方式を使用して、地理的に広い範囲にわたるインターネットワークを構築できます。

図 21–3 は、異なる場所にある 2 つのネットワークをポイントツーポイントリンクで接続した構成を示しています。

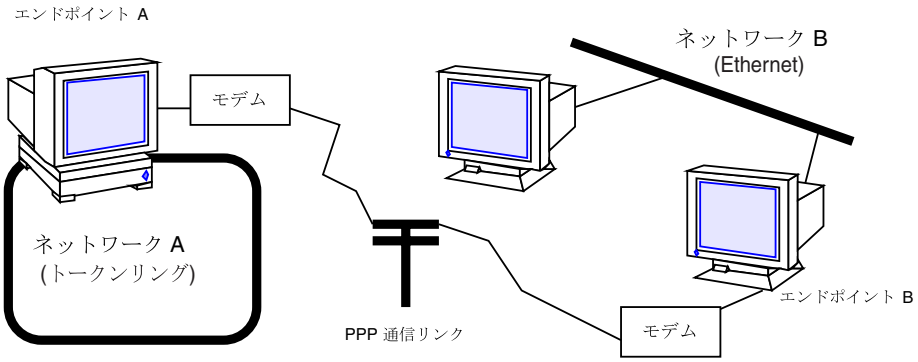


図 21–3　PPP リンクで接続された 2 つのネットワーク

この例では、エンドポイント A と B、それぞれのモデム、公衆電話回線、PPP ソフトウェアが、ネットワーク間のルーターとして働きます。これらのネットワークには、物理ネットワーク間のルーターとして機能する別のホストが存在することもあります。また、PPP ルーターとして機能するホストが追加のネットワークインタフェースボードを備えていて、同時に物理ネットワークのルーターとして機能する場合もあります。

マルチポイント通信リンク

Solaris PPP を使用して、マルチポイント通信リンクを設定できます。この種類の構成では、それぞれ個々のマシンが通信リンク上の 1 つのエンドポイントとして働きます。リンクの 1 つの端に複数のエンドポイントマシンが存在する場合もあります。これは、通信リンクの両端に 1 つずつしかエンドポイントがないポイントツーポイント構成とは異なります。

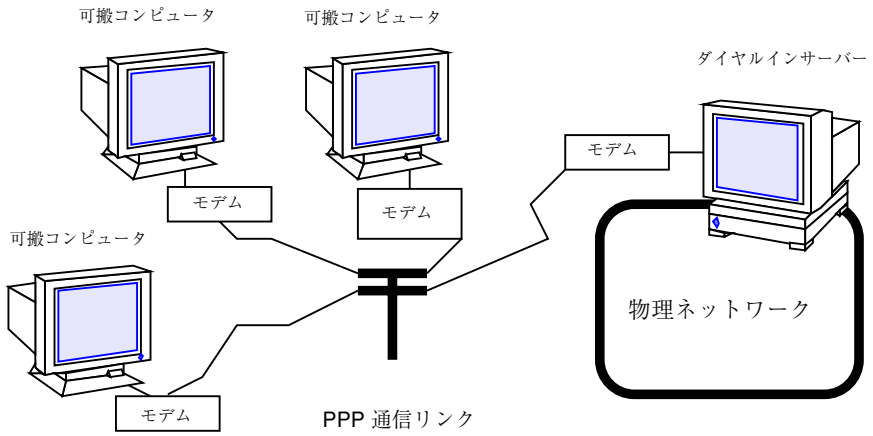


図 21–4　可搬コンピュータとマルチポイントダイヤルインサーバー

PPP がサポートするマルチポイント構成

PPP によって構成できるマルチポイントリンクには、次の 2 つの種類があります。

- ダイヤルインサーバーとリモートマシンとのマルチポイント接続 (図 21–4)
- 3 台以上の可搬コンピュータから成る論理ネットワーク、つまり仮想ネットワーク (図 21–5)

次の各節では、これらの構成の概略を説明します。各構成の設定方法については、第 22 章で説明します。

マルチポイントダイヤルインサーバー

図 21–3 では、地理的に離れた場所にある 3 台のコンピュータが、ネットワーク上のエンドポイントマシンへのポイントツーポイントリンクを介して、互いに通信します。しかし、ネットワークエンドポイントマシンは、マルチポイントリンクを介して可搬コンピュータと通信できるので、このマシンはマルチポイントダイヤルイ

ンサーバーとみなすことができます (461ページの「動的ポイントツーポイントリンクを持つダイヤルインサーバー」で説明したように、動的ポイントツーポイント接続を持つダイヤルインサーバーも設定できます)。

ダイヤルインサーバーは、マルチポイント PPP リンクの反対側にあるすべてのマシンと通信できます。図 21–4 の各マシンはマルチポイントダイヤルインサーバーとは直接通信できますが、各マシンどうしが直接通信することはできません。各マシンは、ダイヤルインサーバーを介して、互いに情報を受け渡しする必要があります。

仮想ネットワーク

PPP を使用して仮想ネットワークを設定できます。この設定では、モデム、PPP ソフトウェア、電話回線が、「仮想」ネットワークメディアとなります。Ethernet やトークンリングなどの物理ネットワークでは、コンピュータはケーブルで直接ネットワークメディアに接続されています。仮想ネットワークでは、現実のネットワークメディアは存在しません。

仮想ネットワーク上で各マシンをマルチポイント通信リンクにより接続した場合、マシンはどれも対等ホストとなります。各ホストは、モデムと電話回線を介して、他のエンドポイントマシンと通信できます。各コンピュータはダイヤルインマシンとしても機能するので、仮想ネットワーク上の対等ホストからのダイヤルインを受け入れることができます。

図 21–5 は、モデムと電話回線によって相互に接続されている可搬コンピュータで構成されている、仮想ネットワークを示しています。

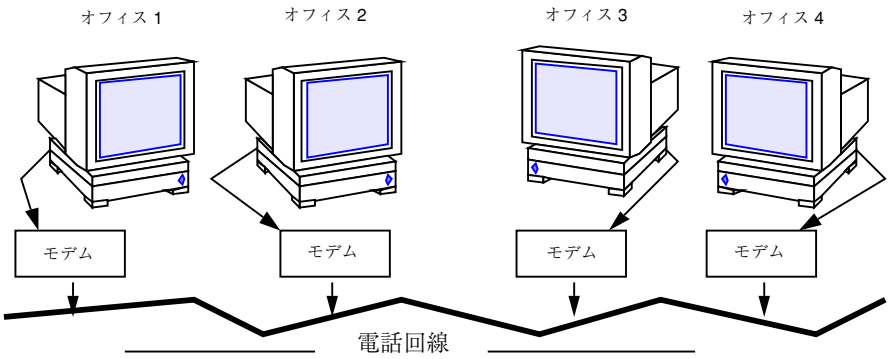


図 21–5　可搬コンピュータの仮想ネットワーク

各マシンはそれぞれ、仮想ネットワーク上の他のマシンから離れた場所にある別々のオフィスに設置されていますが、マルチポイント通信リンクを介して、他の対等ホストとの通信を確立できます。

PPP ソフトウェアの紹介

PPP のコンポーネントソフトウェアには次のものがあります。

- リンクマネージャ (/usr/sbin/aspppd)
- ログインサービス (/usr/sbin/aspppls)
- 構成ファイル (/etc/asppp.cf)
- ログファイル (/var/adm/log/asppp.log)
- FIFO ファイル (/tmp/.asppp.fifo)

PPP ソフトウェアのインストールが終わると、PPP 用の実行制御スクリプトである /etc/init.d/asppp ファイルが作成されています。このファイルは、実行制御ディレクトリ内の他のいくつかのファイルにリンクしています。

図 21–6 に、PPP の各ソフトウェアコンポーネントと、それぞれの相互作用を示します。

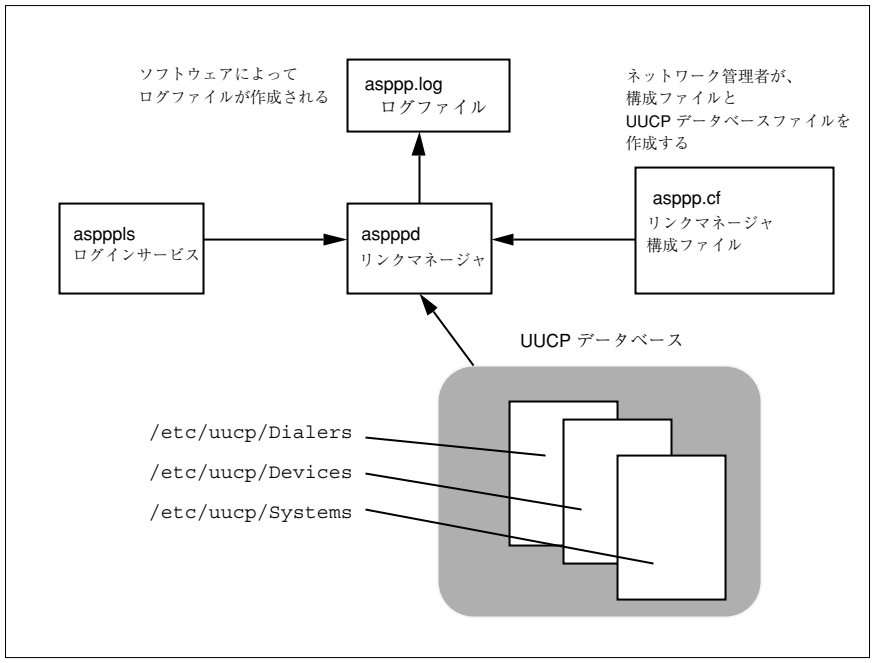


図 21–6　PPP のコンポーネントソフトウェア

リンクマネージャ

/usr/sbin/aspppd リンクマネージャは、ユーザーレベルのデーモンで、PPP サービスが必要となったときのリモートホストへの接続プロセスを自動化します。この自動化されたプロセスは、IP トラフィックを生じさせるようななんらかの動作が生じるたびに起動されます (たとえば、ユーザーがリモートマシンにログインしたり、NFS によりマウントされたファイルにアクセスしたりした場合)。リモートホストが接続を確立しようとすると、ローカルホストのリンクマネージャが接続を完了します。

リンクマネージャの詳細は、aspppd(1M) のマニュアルページを参照してください。

ログインサービス

/usr/sbin/aspppls ログインサービスは、ユーザーがダイヤル呼び出しを行い、ログインした後で、PPP を起動するログインシェルとして呼び出されます。このログインサービスの機能は、528ページの「UUCP ソフトウェア」で説明する

/usr/lib/uucp/uucico コマンドに似ています。マシンをダイヤルインサーバーとして構成するときは、ローカルホストへのダイヤルインが許されているすべての可搬コンピュータについて、/etc/passwd ファイルの中の対応するエントリのログインシェルに aspppls を指定する必要があります。

構成ファイル

asppp.cf ファイルは、ローカルホストの通信相手の各リモートエンドポイントに関する情報を、リンクマネージャに与えます。この情報は、構成ファイル内の path というセクションに定義します。また、path セクションは、使用する PPP インタフェースを定義し、さらにオプションとして、通信をどのように行うかについてのその他の属性 (セキュリティに関する事項など) も定義します。asppp.cf ファイルの各セクションについては、500ページの「基本構成ファイルの各部分」で詳しく説明します。例 21–1 に、変更されていない asppp.cf ファイルを示します。

例 **21–1** 未変更の状態の asppp.cf ファイル

```
#ident "@(#)asppp.cf 10 93/07/07 SMI"
#
# Copyright (c) 1993 by Sun Microsystems, Inc.
#
# Sample asynchronous PPP /etc/asppp.cf file
#
#

ifconfig ipdptp0 plumb mojave gobi private up

path
   inactivity_timeout 120    # Approx. 2 minutes
   interface ipdptp0
   peer_system_name Pgobi    # The name this system logs in with when
                             # it dials this server
                             # *OR* the entry we look up in
                             # /etc/uucp/Systems when we dial out.
```

ログファイル

リンクマネージャは、メッセージを生成し、それをログファイル /var/adm/log/asppp.log に記録します。このファイルに記録される詳細さのレベルは、aspppd の -d オプションか、構成ファイル内の debug_level キーワードにより制御されます。詳細については、524ページの「構成キーワード」と aspppd(1M) のマニュアルページを参照してください。

FIFO ファイル

PPP FIFO ファイル /tmp/.asppp.fifo は、aspppd と asppplsの間の通信に用いる名前付きパイプです。PPP ログインサービスがリンクマネージャに接続するためには、このファイルが /tmp に入っていなければなりません。/tmp/.asppp.fifo ファイルは、リンクマネージャが作成、管理、および削除を行います。

UUCP データベース

Solaris PPP は、コンポーネントソフトウェアの他に、/etc/uucp/Systems、/etc/uucp/Dialers、 /etc/uucp/Devices の 3 つの UUCP ファイルを利用して、通信リンクを確立します。ホストが PPP リンクを介してダイヤルアウトできるようにするには、これらのファイルを修正する必要があります。あるいは、/etc/uucp/Sysfiles を使用して、Systems、Devices、および Dialers ファイルに別の名前を指定することもできます。

これらの UUCP ファイルについての詳細は、第 25 章を参照してください。

コンポーネント間の相互作用

この節では、PPP の各種コンポーネントが、アウトバウンド接続とインバウンド接続についてどのような働きをするかを説明します。

アウトバウンド接続のシナリオ

PPP リンクの 1 つのエンドポイントのユーザーが、反対側のエンドポイントにある対等ホストの参加を必要とする活動を開始すると、アウトバウンド通信が始まります。ユーザーが rcp コマンドを入力して、リンクの反対側のホストからファイルをコピーしようとしたとすると、次に示すような動作が生じます。

1. rcp は、TCP/IP プロトコルスタックの各レベルを通してデータを送り出す
2. 仮想ネットワークインタフェース (ipd*n* または ipdptp*n*) が、IP パケットの形式でデータを受け取る
3. インタフェースは、アウトバウンド接続を開始するための接続要求を、aspppd リンクマネージャに送り出す

4. リンクマネージャは次のことを行う

   a. 接続要求が、/etc/asppp.cf 構成ファイル中で構成されているパスに対応していることを確認する

   b. UUCP データベースファイル (/etc/uucp/Systems、/etc/uucp/Devices、/etc/uucp/Dialers) を調べて、モデムと宛先システムに関する必要な情報を入手する

   c. 宛先ホストへの電話呼び出しをかけるか、適切な直結シリアル回線に接続する

5. 対等ホストへの物理リンクが確立される
6. リンクマネージャは PPP を構成して開始する
7. データリンク層が確立され、対等ホスト上の PPP モジュールが通信を開始する
8. リンクマネージャはリンクを介した IP を使用可能にする

リンクマネージャは次に、アイドルタイムアウト、回線の切断、エラー条件などのイベントが発生するまで、接続を監視します。これらのイベントのどれかが発生すると、リンクマネージャは対等ホストとの接続を切り離し、アイドル状態に戻ります。

インバウンド接続のシナリオ

インバウンド通信を開始するホストがログインすると、/usr/sbin/aspppls ログインサービスが呼び出され、次のイベントが発生します。

1. ログインサービスは、/tmp/.asppp.fifo ファイルを通してリンクマネージャに接続する
2. ログインサービスは、リンクの反対側のエンドポイントで使用するログイン名などの情報を、リンクマネージャに提供する
3. リンクマネージャはこのログイン名を使用して、対応する構成済みのパスを、構成ファイルの中から見つける
4. リンクマネージャは、PPP を構成し起動する
5. データリンク層が確立され、対等ホスト上の PPP モジュールが通信を開始する
6. リンクマネージャはリンクを介した IP を使用可能にする

リンクマネージャは次に、アイドルタイムアウト、回線の切断、エラー条件などのイベントが発生するまで、接続を監視します。これらのイベントのどれかが発生すると、リンクマネージャは対等ホストとの接続を切り離し、アイドル状態に戻ります。

PPP のセキュリティ

構成に含まれているすべてのマシンに PPP をインストールしたあと、PPP リンクに関する 1 レベルまたは 2 レベルのセキュリティを付加できます。

第 1 のレベルのパスワード認証プロトコル (PAP) は、最小限のセキュリティです。認証が確認されるかまたは接続が切断されるまで、パスワードを暗号化しない状態で回線上に送り出します。

第 2 レベルのセキュリティであるチャレンジハンドシェーク認証プロトコル (CHAP) は、ポイントツーポイントリンクの反対側にある対等ホストの識別情報を、定期的に検査します。認証者、つまり接続またはチャレンジ (Challenge) を開始するシステムが、対等ホストにチャレンジメッセージを送ります。これに対する応答が、リンクを介さずに渡されている「シークレット」と照合され、両者の値が一致すれば認証が確認されます。一致しない場合は、接続は切断されます。PPP のセキュリティを付加する方法については、485ページの「PAP または CHAP セキュリティのための `asppp.cf` の編集」で説明します。

第 22 章

PPP 構成の計画

PPP ソフトウェアを構成する前に、必要なハードウェアとソフトウェアの準備を整え、構成に必要な情報を収集する必要があります。この章では、構成の前に行う必要のある作業について説明します。主に次のような作業があります。

- 457ページの「構成に応じた要件の決定」
- 464ページの「PPP リンク用の IP アドレス指定の決定」
- 467ページの「ルーティングに関する考慮事項」
- 467ページの「PPP のハードウェア要件」
- 468ページの「PPP 構成前のチェックリスト」

この章の末尾には、PPP リンクを構成する前に、上記の点を確認するためのチェックリストがあります (表 22–1 を参照)。

構成に応じた要件の決定

Solaris PPP は、次のようなさまざまな構成をサポートしています。

- ポイントツーポイントリンクを介した、リモートコンピュータ対ネットワーク
- ポイントツーポイントリンクを介した、リモートコンピュータ対リモートコンピュータ
- ポイントツーポイントリンクを介した、ネットワーク対ネットワーク
- 1 つまたは複数の動的ポイントツーポイントリンクを介した、ダイヤルインサーバー対複数のリモートコンピュータ

- マルチポイントリンクを介した、ダイヤルインサーバー対複数のリモートコンピュータ
- 仮想ネットワークを形成する複数のリモートコンピュータ。すべてがマルチポイントリンクを介して通信を行う

これらの構成については、443ページの「PPP によるネットワークの拡張」で紹介しました。

この節では、構成を始める前に確認しておかなければならない情報と、行なっておかなければならない作業について構成別に説明します。 設定したい構成に該当する節をお読みください。

検討を要する事項は次のとおりです。

- ネットワークインタフェース
- アドレス指定方式
- ネームサービスを使用するかどうか
- ダイヤルインとダイヤルアウトのサポート
- ルーティングの要件

リモートコンピュータ対ネットワークの構成

リモートコンピュータ対ネットワークは、最も一般的な非同期 PPP 構成です。この構成を使用するのは、リモートオフィスやユーザーの自宅にあるマシンが、ポイントツーポイントリンクを介してダイヤルアウトし、ネットワーク上のダイヤルサーバーに接続する場合です。

- ネットワークインタフェース – このポイントツーポイントリンクは、`ipdptp`*n* 仮想ネットワークインタフェースを使用します。ネットワークへのダイヤルアウトを行うすべてのリモートマシンの構成ファイルの中で、このネットワークインタフェースを指定しておく必要があります。
- アドレス指定方式 – 構成ファイルには、リンクを介して通信するマシンのホスト名または IP アドレスが含まれていなければなりません。リモートホストについては、既存のホスト名と IP アドレスを使用するのが普通です。詳細は、464ページの「PPP リンク用の IP アドレス指定の決定」を参照してください。
- ネームサービス – リモートホスト用のネームサービスとしては、NIS と NIS+ はお勧めできません。これらのサービスは、予想外のときに大量のネットワークトラフィックを生じることがあります。この種類の構成の場合は、DNS ネーム

サービスの方が効率的です。DNS は、『*Solaris* ネーミングの管理』の説明に従って、各リモートホストごとに設定してください。DNS を使用しない場合は、PPP は、リモートマシン上の `/etc/inet/hosts` ファイルを使用します。

- ダイヤルインとダイヤルアウトのサポート – 通常、リモートホストはダイヤルアウト通信だけを実装しています。リモートホストは、他のマシンからの直接的なダイヤルインは受け入れません。したがって、ダイヤルアウト通信をサポートできるようにするには、各マシンの UUCP ファイルを更新する必要があります。その方法については、479ページの「UUCP データベースの編集」で説明します。
- ルーティングの要件 – Solaris TCP/IP プロトコルスタックの一部として RIP が組み込まれているので、リモートホストではデフォルトにより RIP が実行されます。性能の改善のために必要なら、RIP を止めて、代わりに静的ルーティングを使用します。詳細は、160ページの「ルーティングプロトコル」と、481ページの「RIP をオフにする」を参照してください。

リモートホスト対リモートホストの構成

ホスト対ホストの構成を確立するのは、物理的に異なる位置にある 2 つのリモートホスト間のポイントツーポイント通信を確立する場合です。この構成は、リモートオフィスにある 2 つのスタンドアロンマシンの間で情報を交換したい場合に便利です。物理ネットワークは関与しません。

- ネットワークインタフェース – この基本的なポイントツーポイントリンクは、`ipdptp`*n* 仮想ネットワークインタフェースを使用します。両方のエンドポイントの構成ファイルの中で、このインタフェースを指定しておく必要があります。
- アドレス指定方式 – 構成ファイルには、リンクを介して通信するマシンのホスト名または IP アドレスが含まれていなければなりません。一次ネットワークインタフェースに割り当てられている既存のホスト名と IP アドレスがある場合は、それらを使用します。既存のものがない場合は、エンドポイント用の IP アドレスを作成します。詳細は、464ページの「PPP リンク用の IP アドレス指定の決定」を参照してください。
- ネームサービス – 2 つの対等ホストが通信し合うだけなので、本当のネームサービスは必要ありません。両方の対等ホスト上にある `/etc/inet/hosts` ファイルを使用して、アドレスが解決されます。

- ダイヤルインとダイヤルアウトのサポート – 両方のマシンが、ダイヤルイン操作とダイヤルアウト操作を行う必要があります。したがって、両方のエンドポイントの UUCP データベースと /etc/passwd を修正する必要があります。
- ルーティングの要件 – Solaris TCP/IP プロトコルスタックの一部として RIP が組み込まれているので、リモートホストではデフォルトにより RIP が実行されます。性能の改善のために必要なら、RIP を止めて、代わりに静的ルーティングを使用します。詳細は、160ページの「ルーティングプロトコル」と、481ページの「RIP をオフにする」を参照してください。

ネットワーク対ネットワークの構成

ネットワーク対ネットワークの PPP 構成を使用するのは、物理的に離れた場所にある 2 つのネットワークを連結してインターネットワークを構築したい場合です。その場合は、モデムと PPP ソフトウェアが、ネットワークを相互に接続するルーターとして働きます。

- ネットワークインタフェース – ポイントツーポイントリンクは、ipdptp*n* 仮想ネットワークインタフェースを使用します。2 つのネットワークを連結する両方のエンドポイントマシンの構成ファイルの中で、ipdptp*n* を指定しておく必要があります。
- アドレス指定方式 – 構成ファイルには、リンクを介して通信するマシンのホスト名または IP アドレスが含まれていなければなりません。この種類の構成については 2 種類のアドレス指定が考えられます。これについては、464ページの「PPP リンク用の IP アドレス指定の決定」に説明があります。
- ネームサービス – この種類の PPP リンクでは、NIS ネームサービスと NIS+ ネームサービスを使用できます。しかし、各ネットワークがそれぞれ別個のドメインであることが必要です。DNS を使用する場合は、2 つのネットワークが同じドメインに属していてもかまいません。詳細は、『*Solaris* ネーミングの管理』を参照してください。ローカルファイルをネームサービスとして使用する場合は、両方のエンドポイントマシン上にある /etc/inet/hosts ファイルを使用して、アドレスが解決されます。このファイルには、リンクを介した通信ができる、各ネットワーク上のすべてのホストのホスト名と IP アドレスが含まれている必要があります。
- ダイヤルインとダイヤルアウトのサポート – 両方のネットワークエンドポイントマシンが、ダイヤルイン操作とダイヤルアウト操作を行う必要があります。したがって、両方のエンドポイントの UUCP と /etc/passwd ファイルを修正する必要があります。

- ルーティングの要件 – 通常、ネットワーク対ネットワークのリンクのエンドポイントは、RIP を実行することによりルーティング情報を交換します。この構成の場合は、RIP を使用禁止にしないでください。

動的ポイントツーポイントリンクを持つダイヤルインサーバー

動的ポイントツーポイントリンクは、リモートホストからアクセスするネットワークエンドポイントとして機能する、ダイヤルインサーバー用に使用できる 2 つの種類の構成のうちの 1 つです。この構成方式では、サーバーは、動的に割り当てられたポイントツーポイントリンクを介してリモートホストに接続します。ダイヤルインサーバーは、必要時提供の方式で動的リンクを使用して、サービス対象のリモートホストとの通信を確立します。

- ネットワークインタフェース – 動的ポイントツーポイントリンクは、`ipdptp*` 仮想ネットワークインタフェースを使用します。アスタリスクはワイルドカード文字です。このアスタリスクの働きにより、リンクが動的に割り当てられます。構成ファイルの中に、このインタフェースを指定しておく必要があります。
- アドレス指定方式 – 構成ファイルには、リンクを介して通信するマシンのホスト名または IP アドレスが含まれていなければなりません。詳細は、464ページの「PPP リンク用の IP アドレス指定の決定」を参照してください。
- ネームサービス – NIS と NIS+ はリモートホスト用としてはお勧めしませんが、リモートホスト対ネットワークの構成でのダイヤルインサーバーは、それが物理的に接続されているネットワーク上の NIS クライアントとなることができます。NIS がサーバーの物理ネットワーク上にある場合は、リモートホストのホスト名と IP アドレスによって NIS マップを更新してください。DNS は、ダイヤルインサーバーとそのリモートホストのどちらにも使用できます。DNS とネームサービスの一般的な事項については、『*Solaris* ネーミングの管理』を参照してください。ローカルファイルをネームサービスとして使用する場合は、PPP はダイヤルインサーバーの `/etc/inet/hosts` ファイルを使用して、アドレスを解決します。
- ダイヤルインサポート – 動的ポイントツーポイントダイヤルインサーバーの `/etc/passwd` ファイルを更新する必要があります。動的リンクサーバーは、直接にはリモートホストへのダイヤルアウトをしません。
- ルーティングの要件 – Solaris TCP/IP プロトコルスタックの一部として RIP が組み込まれているので、リモートホストではデフォルトにより RIP が実行されます。性能の改善のために必要なら、RIP をオフにして、代わりに静的ルーティン

グを使用します。詳細は、160ページの「ルーティングプロトコル」と、481ページの「RIP をオフにする」を参照してください。

マルチポイントダイヤルインサーバー

マルチポイントリンクは、リモートマシンからアクセスするネットワークエンドポイントとして機能するダイヤルインサーバー用に使用できる、2 つの種類の構成のうちの 1 つです。この構成では、ダイヤルインサーバーは、同じマルチポイントリンクを介して複数のリモートホストを接続します。458ページの「リモートコンピュータ対ネットワークの構成」で説明したように、リモートホストは、常にポイントツーポイントリンクを介してダイヤルインサーバーに接続されます。

この構成を使用するのは、リモートホストとダイヤルインサーバーから成る独立したネットワークを定義したい場合です。

- ネットワークインタフェース – マルチポイントリンクは、`ipd`*n* 仮想ネットワークインタフェースを使用します。ダイヤルインサーバーの構成ファイルの中に、このインタフェースを指定しておく必要があります。
- アドレス指定方式 – 構成ファイルには、リンクを介して通信するマシンのホスト名または IP アドレスが含まれていなければなりません。詳細は、464ページの「PPP リンク用の IP アドレス指定の決定」を参照してください。マルチポイントリンク上のホストのための独立したネットワークを作成する必要があります。詳細は、466ページの「PPP リンクへのネットワーク番号の割り当て」を参照してください。
- ネームサービス – NIS と NIS+ はリモートホスト用としては勧められませんが、リモートホスト対ネットワークの構成でのダイヤルインサーバーは、それが物理的に接続されているネットワーク上の NIS クライアントとなることができます。NIS がサーバーの物理ネットワーク上にある場合は、リモートホストのホスト名と IP アドレスによって NIS マップを更新してください。DNS は、ダイヤルインサーバーとそのリモートホストのどちらにも使用できます。DNS とネームサービスの一般的な事項については、『*Solaris* ネーミングの管理』を参照してください。ローカルファイルをネームサービスとして使用する場合は、PPP はダイヤルインサーバーの `/etc/inet/hosts` ファイルを使用して、アドレスを解決します。
- ダイヤルインとダイヤルアウトのサポート – マルチポイントダイヤルインサーバーは、PPP 仮想ネットワークと、サーバーが接続している物理ネットワークとの間のネットワークルーターとして働きます。サーバーは、PPP ネットワークを宛先とする IP トラフィックを物理ネットワークから受け取るたびに、リモート

ホストに対してダイヤルアウトします。したがって、マルチポイントダイヤルインサーバーは、ダイヤルインサポートとダイヤルアウトサポートの両用として構成し、UUCP と /etc/passwd ファイルを更新する必要があります。

- ルーティングの要件 – ipd*n* インタフェースは RIP をサポートしません。RIP を使用禁止にする必要はありません。

仮想ネットワーク上のホスト

仮想ネットワーク構成を使用するのは、電話回線、モデム、PPP ソフトウェアを使用して、物理的に離れた場所にある 3 台以上のコンピュータを 1 つの仮想ネットワークにしたい場合です。

- ネットワークインタフェース – この種類の構成はマルチポイントリンクを必要とし、マルチポイントリンクは ipd*n* 仮想ネットワークインタフェースを使用します。このインタフェースは、各エンドポイントシステムを、仮想ネットワークの反対側のエンドポイントに接続します。
- アドレス指定方式 – 構成ファイルには、リンクを介して通信するマシンのホスト名または IP アドレスが含まれていなければなりません。詳細は、464ページの「PPP リンク用の IP アドレス指定の決定」を参照してください。仮想ネットワークにはネットワーク番号を割り当てる必要があります。詳細は、465ページの「一意な IP アドレスとホスト名の作成」を参照してください。
- ネームサービス – 仮想ネットワークでは、NIS と NIS+ を実行できます。しかし、これはリンクの性能を低下させることがあります。DNS の方が効率的です。これらのネームサービスの設定方法については、『*Solaris* ネーミングの管理』を参照してください。ローカルファイルをネームサービスとして使用する場合は、仮想ネットワークを形成するすべてのマシンのホスト名と IP アドレスにより、各マシンの /etc/inet/hosts を更新する必要があります。
- ダイヤルインサポートとダイヤルアウトサポート – 仮想ネットワーク内のすべてのマシンを、ダイヤルイン操作とダイヤルアウト操作の両用として構成し、UUCP と /etc/passwd ファイルを更新する必要があります。
- ルーティングの要件 – ipd*n* インタフェースは RIP をサポートしません。RIP を使用禁止にする必要はありません。

PPP リンク用の IP アドレス指定の決定

PPP リンクを介した通信ができるようにするには、リンクの一端にあるマシンが、リンクの反対側にある対等ホストのホスト名と IP アドレスを認識している必要があります。PPP 構成は、特定のアドレス指定スキーマを必要とすることがよくあります。この節では、各アドレス指定スキーマと、それぞれをどのような場合に使用するかについて説明します。

IP アドレスの指定

各エンドポイントマシンでは、次の場所にアドレス指定情報を指定します。

- /etc/asppp.cf 構成ファイル
- /etc/inet/hosts ファイル
- NIS+、NIS、DNS データベースのどれか (該当する場合)

ローカルマシンの asppp.cf ファイルを編集するときに、リンク上に配置する各エンドポイントマシンについて、ホスト名と、場合によっては IP アドレスを指定する必要があります。たとえば、各エンドポイントの IP アドレスまたはホスト名を、構成ファイル内の ifconfig セクション内の引数として入力する必要があります。

```
ifconfig ipdptp0 plumb 192.99.44.01 192.99.44.02 up
```

/etc/asppp.cf の形式については、480ページの「構成ファイルの編集」を参照してください。

さらに、通信を可能にするには、/etc/inet/hosts を編集することにより、リモートエンドポイントの IP アドレスとホスト名を、ローカルエンドポイントの hosts データベースに追加する必要があります。この手順については、111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。

アドレス指定スキーマのタイプ

PPP 用のアドレス指定スキーマはいくつかあり、構成に応じて選択することができます。asppp.cf ファイルと hosts データベースを編集する前に、使用する構成に適切なアドレス指定スキーマを決める必要があります。アドレス指定スキーマには次のものがあります。

- ローカル /etc/inet/hosts ファイル内で一次ネットワークインタフェースに割り当てられているのと同じ IP アドレスを PPP 用に使用する
- 各 PPP エンドポイントに一意な IP アドレスを割り当てる
- PPP リンクが作成したネットワークに新しいネットワーク番号を割り当てる

一次ネットワークインタフェースと同じ IP アドレスの使用

この方式は、ポイントツーポイントリンクの場合にのみ使用できます。このアドレス指定スキーマでは、各エンドポイントについて一次ネットワークインタフェースのアドレスを指定します (一次ネットワークインタフェースについての詳細は、第 2 章を参照してください)。このようなエンドポイントには次のようなものがあります。

- PPP リンクを介して通信する 2 つのスタンドアロンマシン (既存の IP アドレスを持っている場合)
- PPP リンクを介して通信する 2 つのネットワークエンドポイント
- ポイントツーポイントリンクによりネットワークダイヤルインサーバーに接続されているリモートホスト
- 動的割り当てポイントツーポイントリンクによりリモートホストに接続されているダイヤルインサーバー

ローカルエンドポイントの /etc/inet/hosts ファイルを編集するときに、一次ネットワークインタフェースの IP アドレスと、リンクの反対側の対等ホストのホスト名と IP アドレスを入力します。

一意な IP アドレスとホスト名の作成

この方式では、PPP ネットワークインタフェースに、一意なホスト名と IP アドレスを割り当てます (インタフェースを hostname-ppp と名付けるとよいでしょう)。このアドレス指定スキーマは次のものに使用します。

- マルチポイントダイヤルインサーバーとして使用されるネットワーク上のエンドポイントマシン
- 仮想ネットワーク上のマシン
- 専用 IP アドレスを使用して、動的に割り当てられた PPP リンクを介してダイヤルインサーバーと通信するリモートホスト (これは、動的リンク構成の場合の必須条件ではない)

- Ethernet やトークンリングなどの物理ネットワークのルーターとしても構成されているマシン
- スタンドアロン対スタンドアロンの構成において、既存の IP アドレスを持っていないマシン (PPP インタフェースが一次ネットワークインタフェースになる)

asppp.cf 構成ファイル中に、PPP ネットワークの一意なアドレスとホスト名を指定する必要があります。

新しいホスト名と IP アドレスを作成するには、140ページの「hosts データベース」の説明に従って、単にその名前とアドレスを /etc/inet/hosts ファイルに追加するだけです。

PPP リンクへのネットワーク番号の割り当て

PPP 構成用に新しいネットワーク番号を作成するのは、次のものが構成に含まれる場合です。

- PPP マルチポイントリンクを介して通信するコンピュータの仮想ネットワーク (必須)
- マルチポイントダイヤルインサーバーとそのリモートホスト (必須)
- 2 つのネットワーク間の PPP リンク、特にネットワークエンドポイントマシンの一方または両方が物理ネットワークのルーターでもある場合 (省略可能)

(ネットワーク番号については、第 5 章を参照してください)。

PPP リンクは、物理ネットワークメディアを含まないため、仮想ネットワークとなります。すべてのエンドポイントマシンの networks データベースに、その仮想ネットワークのネットワーク番号と、リンクするネットワークのネットワーク番号を入力する必要があります。

例 22–1 は、PPP を使用したインターネットワーク用の /etc/inet/networks ファイルの例を示します。

例 22–1 PPP を使用したインターネットワーク用の /etc/inet/networks ファイル

```
kalahari      192.9.253
negev         192.9.201
nubian-ppp    192.29.15
```

このファイルで、kalahari と negev は 2 つのローカルエリアネットワークで、nubian-ppp は PPP リンクの名前です。

ルーティングに関する考慮事項

Solaris TCP/IP ネットワークでは、デフォルトにより RIP ルーティングプロトコルが実行されます。ほとんどの場合、ポイントツーポイントリンクでは、RIP をそのまま実行させておくのが妥当です。しかし、リンクの性能に問題がある場合は、 ポイントツーポイントリンク上で RIP を使用しないようにした方がよい場合もあります。

注 - マルチポイントリンクでは RIP は起動されません。したがって、マルチポイントリンクの場合は静的ルーティングを設定する必要があります。その方法については、160ページの「ルーティングプロトコル」を参照してください。

RIP を無効にする方法については、481ページの「RIP をオフにする方法」を参照してください。

PPP のハードウェア要件

基本的な PPP 構成には、コンピュータ、モデム、RS-232 電話回線が含まれます。しかし、構成を行う前に、選択したハードウェアが PPP をサポートするものであるかどうかを確認しておく必要があります。この節では、PPP で必要なハードウェアについて説明します。

- モデム – PPP を実行するには、各エンドポイントマシンが、少なくとも 9600 bps 以上の双方向接続をサポートするモデムを備えている必要があります。このようなモデムは、V.32 または V.32bis の仕様を満たしています。
- シリアルポート選択 (ダイヤルインサーバーの場合のみ) – ほとんどの CPU では、シリアルポート A とシリアルポート B のどちらでも、PPP 用として構成できます。ダイヤルインサーバーでポートを初期化するには、Solaris シリアルポートマネージャを使用します。適切なポートを選択する方法については、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』を参照してください。追加のシリアルカードをインストールしてある場合は、そのシリアルポートも PPP 接続用に使用できます。
- ディスク容量 – PPP をインストールするには、`/usr` 内に 300K バイトの空き領域が必要です。

注 - 64 ビットの PPP をインストールするには、/usr 内に 600K バイトの空き領域が必要です。

PPP 構成前のチェックリスト

このチェックリストは、PPP の構成の準備を整えるために使用します。構成プロセスに着手する前に収集する必要のある情報と、行う必要のある作業を列記してあります。

表 22–1 PPP 構成前のチェックリスト

/usr に使用可能な空き領域が 300K バイトありますか。	はい/いいえ
64 ビット の PPP をインストールする場合、/usr にさらに 300K バイトの空き領域がありますか。	はい/いいえ
/ (ルート) に使用可能な空き領域が 4K バイトありますか。	はい/いいえ
各エンドポイントのモデムが、V.32 または V.32bis 以上をサポートしていますか。	はい/いいえ
ダイヤルインサーバーでシリアルポートマネージャを使用して、モデム用のシリアルポートを指定しましたか。	はい/いいえ
各エンドポイントマシンに Solaris PPP をインストールしてあることを確認しましたか (PPP をインストールしていない場合は、pkgadd プログラムまたは admintool ソフトウェアマネージャを使用してインストールできます。インストール方法については、『*Solaris 8* のインストール (上級編)』を参照してください)。	はい/いいえ
各エンドポイントで別のバージョンの PPP が実行されていないことを確認しましたか (そのようなバージョンがある場合は、それぞれのマニュアルの説明に従って不使用にしてください)。	はい/いいえ
PPP リンクに関与するすべてのコンピュータについて、使用する IP アドレスを決定しましたか。	はい/いいえ
すべてのマシンのホスト名と IP アドレスをリストしてください。	_____ _____ _____ _____

表 22–1 PPP 構成前のチェックリスト 続く

ダイヤルインサーバーの名前と IP アドレスを記入してください (該当する場合)。	_____
使用するネットワークインタフェースの名前を記入してください。	_____

第 23 章

PPP の管理

この章では PPP を構成するための手順および情報、あまり一般的には使用されない PPP リンクを設定する手順、およびの障害追跡の方法をいくつか示します。次のトピックについて説明します。

PPP 作業マップ

この節では PPP を構成したり、一度インストールした PPP を保守したり、PPP に関する障害追跡を行なったりするための各種のマップを示します。

表 23–1 PPP 構成作業マップ

作業	説明	参照ページ
PPP がすべてのマシンにインストールされていることを確認する	pkginfo を使用して、PPP が PPP リンク内のすべてのマシンにインストールされていることを確認する	475ページの「インストールを確認する方法」
リモートマシンのホストのデータベースを構成する	/etc/inet/hosts ファイルに IP アドレスとホスト名を追加することにより、リモートマシンのホストのデータベースを構成する	477ページの「リモートマシンの hosts データベースの構成方法」
ダイアルインサーバーのホストデータベースを構成する	/etc/hosts ファイルおよび /etc/inet/networks ファイルにエントリを追加することにより、ダイアルインサーバーのホストデータベースを構成する	478ページの「ダイヤルインサーバーの hosts データベースの構成方法 」
/etc/asppp.cf ファイルを編集して、起動時にエントリが認識されるようにエントリを追加する	リモートエンドポイントとの通信を確立し維持するために必要な情報を /etc/asppp.cf ファイルに加える	481ページの「asppp.cf 構成ファイルの編集方法」
RIP をオフにする	/etc/gateways ファイルに norip を加えて RIP をオフにする	481ページの「RIP をオフにする方法」
リモートホストを更新する	IP アドレスとホスト名を /etc/inet/hosts ファイルに加えて、リモートホストを更新する	484ページの「リモートホストの更新方法」
ダイアルインサーバーを更新する	/etc/inet/hosts ファイルを更新して、サーバーに接続されている各ホストに使用するエントリを加える	484ページの「ダイヤルインサーバーの更新方法」
PAP/CHAP サポートを加える	require_authentication のキーワードと will_do_authentication のキーワードを /etc/asppp.cf ファイルに追加して、リンク上の各マシンが PAP または CHAP に応答するかどうかに関するセキュリティを確立する	486ページの「PAP または CHAP のインストール方法」

表 23–2 PPP 保守作業マップ

作業	説明	参照ページ
PPP を手動で起動する	/etc/init.d/asppp start コマンドを使用して PPP を起動します。通常、PPP は自動的にスタートするため、このコマンドを使用する必要はない	488ページの「手動で PPP を起動する方法」
PPP が作動中であることを確認する	ps コマンドと ping コマンドを使用して、PPP が作動中であるかどうかを確認する	488ページの「PPP が実行中であることを確認する方法」
PPP を停止する	/etc/init.d/asppp stop コマンドを使用して PPP を停止する	489ページの「PPP の停止方法」
インタフェースの状態を確認する	ifconfig コマンドを使用して、回線の現在の状態を監視する	490ページの「インタフェースの状態を確認する方法」
接続状態を確認する	ping コマンドを使用して接続が完了していることを確認する	491ページの「接続状態を確認する方法」
インタフェースの活動を確認する	netstat コマンドを使用して、パケットが送受されていることを確認する	492ページの「インタフェースアクティビティを確認する方法」
ローカルルーティングテーブルを確認する	netstat コマンドを使用してローカルルーティングテーブルを表示する	492ページの「ローカルルーティングテーブルを確認する方法」
in.routed を使用してルートを加える	動的ルーティングを行っているときに、in.routed コマンドを使用してルートを加える	493ページの「in.routed を使用してルートを追加する方法」

表 23–3 PPP 障害追跡の作業マップ

作業	説明	参照ページ
診断機能を設定する	障害追跡のために PPP 診断機能を設定する方法	496ページの「マシンに対する診断の設定方法」

構成プロセスの概要

第 22 章で述べたプリインストール作業が終われば、次は、PPP の構成にとりかかることができます。

PPP については次のことを行う必要があります

1. PPP ソフトウェアのインストール (まだインストールしてない場合)
2. 関与するすべてのマシンの `/etc/inet/hosts` ファイルの編集
3. すべてのダイヤルアウトマシンの UUCP データベースファイルの編集
4. ダイヤルインマシンの `/etc/passwd` ファイルと `/etc/shadow` ファイルの編集
5. リンク上の各マシンの `/etc/asppp.cf` ファイルの編集
6. リンク上の各マシンでのリンクマネージャ `aspppd` の起動
7. PPP が正常に実行されていることの確認

上記の作業 1 ～ 4 は順番どおりに進めなくてもかまいませんが、PPP 構成ファイルの編集の前に、すべて完了しておく必要があります。

この章の各節では、PPP の構成のための手順について説明します。

PPP ソフトウェアのインストール

Solaris インストールプログラムを実行するときに配布ソフトウェア全体を選択すると、PPP ソフトウェアは自動的に組み込まれます。配布ソフトウェア全体を選択しなかった場合は、PPP を個別のパッケージとしてインストールできます。

▼ インストールを確認する方法

次の手順に進む前に、PPP リンクに含めるすべてのマシンに、Solaris バージョンの PPP をインストールしてあることを確認する必要があります。

1. スーパーユーザーになります。

2. リンクに含める各エンドポイントについて、次のように入力します。

```
# pkginfo | grep ppp
```

32 ビット PPP がインストールされている場合は、次のパッケージ名が表示されます。

```
SUNWapppr       PPP/IP Asynchronous PPP daemon configuration files
SUNWapppu       PPP/IP Asynchronous PPP daemon and PPP login service
SUNWpppk        PPP/IP and IPdialup Device Drivers
```

64 ビット PPP がインストールされている場合は、次のパッケージ名が表示されます。

```
SUNWapppr       PPP/IP Asynchronous PPP daemon configuration files
SUNWapppu       PPP/IP Asynchronous PPP daemon and PPP login service
SUNWpppk        PPP/IP and IPdialup Device Drivers
SUNWpppkx       PPP/IP and IPdialup Device Drivers (64-bit)
```

3. **PPP** がインストールされていないエンドポイントシステムがある場合は、pkgadd プログラムまたは admintool ソフトウェアマネージャを使用してインストールしてください。

注 - pkgadd を使用して PPP をインストールする場合は、上記の順序でパッケージをインストールする必要があります。

---

pkgadd プログラムと admintool ソフトウェアマネージャについての詳細は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』を参照してください。

PPP 構成例

この節と以後の各節では、最も一般的な PPP 構成、つまりリモートホストとそのダイヤルインサーバーをサポートするファイルを編集する方法を紹介します。図 23–1 は、この章で例として使用する構成を示しています。この例は、3 台のリモートマシン (nomada、nomadb、nomadc) と、ダイヤルインサーバー nubian で構成されるネットワーク 192.41.43 を表しています。このネットワークは、ダイヤルインサーバー nubian が直接接続しているローカルエリアネットワーク 192.41.40 とは別個のネットワークです。ネットワーク 192.41.40 は、ネームサービスとして NIS を実行しています。

各リモートホストについて示されている IP 番号は、それぞれの PPP ネットワークインタフェースのアドレスです。しかし、ダイヤルインサーバーは、自己の一次ネットワークインタフェースの IP アドレスである 192.41.40.45 の他に、PPP インタフェース用として特別に作成された IP アドレスである 192.41.43.10 も持っています。

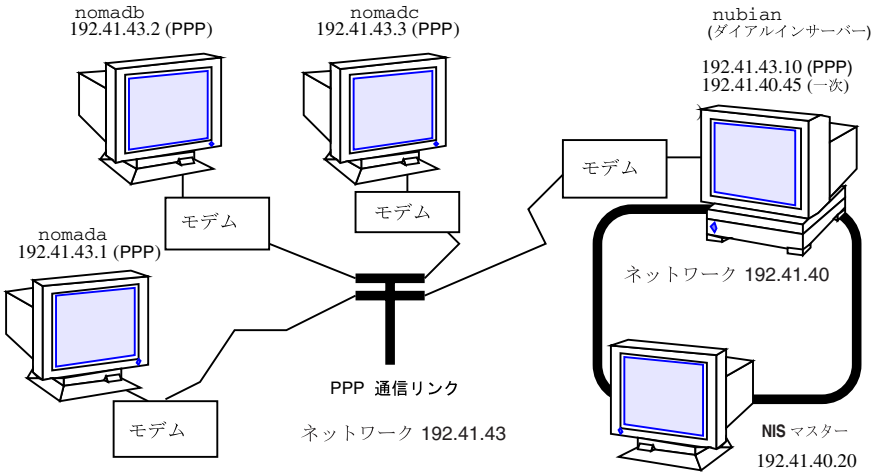


図 23–1　リモートホストとマルチポイントダイヤルインサーバーのネットワーク例

/etc/inet/hosts ファイルの編集

構成に含まれるすべてのマシンに PPP がインストールされていることを確認したら、次に、各マシンの /etc/inet/hosts ファイルを編集します。PPP リンクの反対側にあって、ローカルマシンが通信する必要のあるすべてのマシンについて、hosts データベースにホスト情報を追加する必要があります。

注 - 物理ネットワーク上でどのネームサービスを使用しているかに関係なく、/etc/inet/hosts を更新する必要があります。これは、ブートプロセスの中で、PPP の方がネームサービスデーモンより前に起動されるからです。

▼ リモートマシンの hosts データベースの構成方法

1. スーパーユーザーになります。

2. /etc/inet/hosts ファイルを編集して次の手順を行います。

   a. リンクの反対側にあるダイヤルインサーバー用の **PPP** 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たエントリを追加します。

   図 23–1 では、ダイヤルインサーバー nubian の PPP ネットワークインタフェースの IP アドレスが入ったエントリが、nomada の /etc/inet/hosts ファイルに存在する必要があります。nomadb と nomadc の /etc/inet/hosts ファイルにも、このエントリが必要です。

   b. ダイヤルインサーバーの物理ネットワーク上にあって、リモートホストからのリモートログインが可能な各マシンの **IP** アドレスが入ったエントリを追加します。

   たとえば、 nomadc の /etc/inet/hosts ファイルは次のようになります。

```
# Internet host table
#
127.0.0.1         localhost      loghost
192.41.43.3       nomadc
192.41.43.10      nubian-ppp
192.41.40.20      nismaster
```

3. ネットワークで使用中のネームサーバーがある場合、リモートホストのホスト名と **IP** アドレスでそのネームサーバーのデータベースを更新します。

マルチポイントダイヤルインサーバーの hosts データベース

マルチポイントダイヤルインサーバーは、一次ネットワークインタフェースのローカル IP アドレスのほかに、PPP インタフェース用の一意な IP アドレスも持っていなければなりません。ダイヤルインサーバー用の hosts データベースを構成するために必要な手順は、次のとおりです。

▼ ダイヤルインサーバーの hosts データベースの構成方法

1. スーパーユーザーになります。

2. **PPP** インタフェースの **IP** アドレスが入ったエントリを、ダイヤルインサーバーの /etc/inet/hosts ファイルに追加します。

   たとえば、図 23–1 に示すダイヤルインサーバー nubian の /etc/hosts ファイルは、次のようになります。

```
# Internet host table
#
127.0.0.1           localhost       loghost
192.41.43.10        nubian-ppp
192.41.40.45        nubian
```

3. サーバーの物理ネットワークでネームサービスが使用されていない構成の場合は、次のようにします。

   a. サービス対象となる各リモートホストに関するエントリを、サーバーの /etc/inet/hosts ファイルに追加します。

   b. 物理ネットワーク上にあって、リモートマシンとの通信が許可されているすべてのマシンの /etc/inet/hosts ファイルに、リモートホストについてのエントリを追加します。

4. サーバーとそのリモートホストからなるネットワークの新しいネットワーク番号を、ダイヤルインサーバーの /etc/inet/networks ファイルに追加します。

   詳細は、466ページの「PPP リンクへのネットワーク番号の割り当て」を参照してください。

UUCP データベースの編集

マシンが PPP リンクを介してダイヤルアウトできるようにするには、そのマシンの UUCP データベース内の以下のファイルを編集する必要があります。

- /etc/uucp/Devices
- /etc/uucp/Dialers
- /etc/uucp/Systems

これらのファイルの編集が必要なのは、PPP ダイヤルアウトマシンとして機能するリモートホストの場合です。また、ダイヤルインサーバーがリモートホストへのダイヤルアウトを行う場合も (マルチポイントダイヤルインサーバーの場合の必須条件)、そのダイヤルインサーバーにある上記のファイルを編集する必要があります。これらのファイルについては、第 25 章で詳しく説明します。

/etc/passwd ファイルの修正

ダイヤルインサーバーを構成するには、/etc/passwd ファイルと /etc/shadow ファイルも編集する必要があります。

ダイヤルインサーバーへのログインを許可されている各リモートホストの各ユーザーについて、そのサーバーの /etc/passwd ファイルにエントリを追加する必要があります。リモートホストがダイヤルインサーバーを呼び出す場合、自分自身の UUCP データベースを読み、ユーザー名かユーザー ID をサーバーに渡すことで呼び出しを開始します。すると、サーバーは、/etc/passwd ファイルのユーザー情報に照らして確認します。

そのユーザーのパスワードが認証されると、サーバーは、PPP ホスト用の特別なシェルである /usr/sbin/asppplsにそのユーザーをログインさせます。サーバー

は、この情報を /etc/passwd ファイルのログインシェルエントリから入手します。たとえば、図 23–1 の例の場合、ダイヤルインサーバー nubian の /etc/passwd ファイルには、次のようなエントリが入っています。

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nomada:x:121:99:R. Burton:/:/usr/sbin/aspppls
nomadb:x:122:99:T. Sherpa:/:/usr/sbin/aspppls
nomadc:x:123:99:S. Scarlett:/:/usr/sbin/aspppls
```

/etc/passwd パスワードについての詳細は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』を参照してください。

注 - /etc/passwd ファイル中の情報に加えて、/etc/shadow ファイルもサーバーへのダイヤルインを許可されている各エンドポイントマシンで使用するログイン名のパスワードに更新します。詳細は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』を参照してください。

/etc/asppp.cf 構成ファイルの編集

/etc/asppp.cf 構成ファイルは、エンドポイントマシン上にある PPP リンクマネージャに、リンクの反対側にあるマシンに関する情報、またはマルチポイントリンク (または動的ポイントツーポイントリンク) の反対側にあるマシンに関する情報を提供します。このマシンがブートすると、リンクマネージャはこの情報を使用して、リモートエンドポイントとの通信を確立し維持します。

構成ファイルの編集

asppp.cf を編集するときは、次の点に注意してください。

- 構成ファイル内では、キーワードとキーワードとの間を空白(ブランク、タブ、改行) で区切る
- コメントとして使用する文字列の前には # 記号を付ける。# からその次の改行までの文字はすべてコメントとみなされ無視される

キーワードをファイルに設定するためのフォーマットに対して他の条件は無効となります。

▼ asppp.cf 構成ファイルの編集方法

1. エンドポイントマシンの 1 つでスーパーユーザーになります。
2. /etc ディレクトリに移動します。
3. 汎用 asppp.cf ファイルを編集して、このマシンの **PPP** リンクを定義する情報を追加します。
4. アクセス権が必ず 600 に設定されるように、ファイルを保存します。
5. 残りの各エンドポイントで /etc ディレクトリに移動し、上記の手順 **2** と **3** を繰り返します。

RIP をオフにする

/etc/gateways ファイルにより、ポイントツーポイントリンク上の RIP を無効にすることができます。このファイルはご使用のオペレーティングシステムには入っていません。テキストエディタで作成する必要があります。

▼ RIP をオフにする方法

1. スーパーユーザーになります。
2. /etc/gateways を編集して、次のエントリを追加します。

```
norip ipdptpn
```

ipdptp*n* は、使用されているポイントツーポイントの PPP インタフェースのデバイス名を表します。

詳細は、in.routed(1M) のマニュアルページを参照してください。

PPP のセキュリティの付加

構成に含まれるすべてのマシンに PPP をインストール後、asppp.cf を修正することによって、PPP リンクについての PAP または CHAP レベルのセキュリティを付加できます。485ページの「PAP または CHAP セキュリティのための asppp.cf の編集」を参照してください。

動的割り当て PPP リンクの構成

動的ポイントツーポイントリンクを持つダイヤルインサーバーを使用するサイトでは、ポイントツーポイント通信の利点を最大限に活用することができます。この構成タイプについては、第 21 章で概説しました。この構成では、必要時に動的にポイントツーポイントリンクを割り当てる少なくとも 1 つのダイヤルインサーバーと、リモートホストとの間で通信が行われます。この節では、図 23–2 に示す構成例に基づいて説明を進めます。

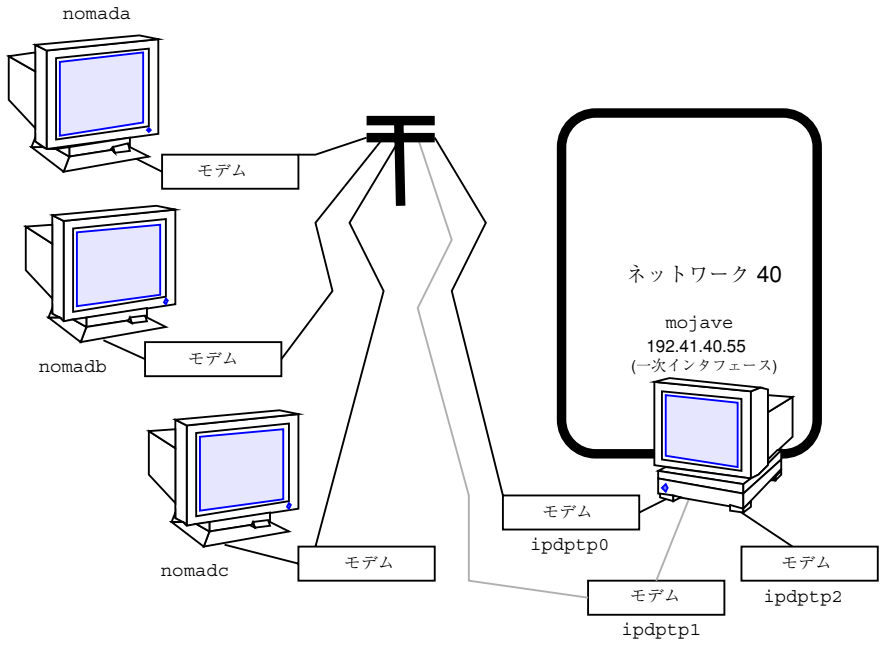


図 23–2　リモートホストと動的リンクダイヤルインサーバーのネットワーク

各リモートホストは、標準のポイントツーポイントリンクを使用してダイヤルインサーバーと通信します。しかし、図 23–1 に示したマルチポイントダイヤルインサーバーとは違って、ダイヤルインサーバー mojave は、動的ポイントツーポイントリンクを介して呼び出し側ホストに接続されます。リモートホストのどれかが接続を確立しようとすると、サーバーが使用可能なリンクを割り当てます。

動的リンクの基本概念は、接続確立のたびにサーバーがクライアントに IP アドレスを供給するというものです。接続を確立すると、使用可能な IP インタフェースをサーバーがクライアントに割り当てます。その後、接続が継続している間、インタフェースのリモート IP アドレスがクライアントの IP アドレスになります。接続を終了すると、使用可能なインタフェースのプールに IP インタフェースが戻され、別の接続に使用できる状態になります。

動的リンクの構成には、リモートホスト対マルチポイントダイヤルインサーバーの場合と同じ一般的な手順を用います。この手順については、474ページの「構成プロセスの概要」に説明があります。ただし、動的ポイントツーポイントリンクには独自の必要条件がいくつかあり、そのため構成に関係するファイルに対する修正の方法も少々異なります。

▼ リモートホストの更新方法

リモートマシンの hosts データベース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ます。

2. リンクの反対側にある各ダイヤルインサーバーについて、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を /etc/inet/hosts ファイルに追加します。

   たとえば、図 23–2 では、nomada、nomadb、および nomadc の /etc/inet/hosts ファイルには、ダイヤルインサーバー mojave の一次ネットワークインタフェースの IP アドレスが入ります。

3. ダミー **IP** アドレスを追加します。

   この IP アドレスが使用されるのは、PPP の起動時だけです。

   nomadc の /etc/inet/hosts ファイルは、次のように表示されます。

```
# Internet host table
#
127.0.0.1          localhost        loghost
192.41.40.55       mojave
1.2.3.4            dummy
```

4. ダイヤルインサーバーの物理ネットワーク上にあって、リモートホストからリモートログインできるすべてのマシンの **IP** アドレスを、/etc/inet/hosts ファイルに追加します。

5. 物理ネットワーク上にあるネームサーバーのデータベースを、リモートホストのホスト名と **IP** アドレスに更新します。

▼ ダイヤルインサーバーの更新方法

ダイヤルインサーバーの hosts データベースには、PPP 固有のアドレスを追加する必要はありません。動的割り当てリンクは、サーバーのネットワークインタフェースを使用する必要があります。したがって、ダイヤルインサーバーの hosts データベースを構成するには、次の作業を行います。

1. スーパーユーザーになります。

2. サービス対象の各リモートホストについて、サーバーの /etc/inet/hosts ファイルにエントリを追加します。

3. 物理ネットワーク上のすべてのマシンの /etc/inet/hosts ファイルに、それぞれが通信することのできるリモートホストに関するエントリを追加します。

PAP または CHAP セキュリティのための asppp.cf の編集

asppp.cf ファイルを編集することによってセキュリティを設定し、リンクの各部分が、パスワード認証プロトコル (PAP) またはチャレンジハンドシェーク認証プロトコル (CHAP) に応答するかどうかを指定できます。PAP と CHAP については、455ページの「PPP のセキュリティ」で説明してます。asppp.cf ファイルを編集するには、一連のキーワードを追加します。この節では、認証システムはリンクまたはチャレンジを開始するシステムであり、これは多くの場合サーバーです。対等システムはリンクの反対側にあるシステムであり、これは多くの場合クライアントです。

追加するキーワードは、require_authentication と will_do_authentication です。認証システムつまりサーバーは通常、認証を要求し、対等システムつまりクライアントは通常、認証を行います。

表 23–4 認証システムのキーワードと関連の文字列

require_authentication pap	require_authentication chap
pap_peer_id	chap_peer_secret
pap_peer_password	chap_peer_name

表 **23–5** 対等システムのキーワードと関連の文字列

will_do__authentication pap	will_do_authentication chap
pap_id pap_password	chap_secret chap_name

▼ PAP または CHAP のインストール方法

1. サーバー上のスーパーユーザーになります。

2. /etc/asppp.cf ファイルを編集します。

3. リンク上の各マシンについて require_authentication キーワードを追加して、**PAP** セキュリティと **CHAP** セキュリティのどちらを使用するかを指定します。

   a. 各 **pap** キーワードについて、関連の pap_peer_id と pap_peer_password 文字列を追加します。

   b. 各 **chap** キーワードについて、関連の chap_peer_secret と chap_peer_name 文字列を追加します。
   これらのキーワードは明示的に指定することも、パスのデフォルト値を使用することもできます。各キーワードによって指定される内容については、表 24–1 を参照してください。また、例 23–1 は、/etc/asppp.cf ファイルの例を示します。

4. will_do_authentication キーワードを使用して、リンク上で **PAP** セキュリティまたは **CHAP** セキュリティを使用する各リモートホストについて、リモートホストの /etc/asppp.cf ファイルにエントリを追加します。

   a. 各 **pap** キーワードについて、関連の pap_id と pap_password 文字列を追加します。

   b. 各 **chap** キーワードについて、関連の chap_secret と chap_name 文字列を追加します。

PAP と CHAP の例

例 23–1 は、PAP と CHAP の認証を必要とするサーバー mojave 用の asppp.cf ファイルを示しています。対等システムは、nomada (PAP) と nomadb (CHAP) です。

例 23–1 サーバー mojave 用のコード例

```
ifconfig ipdptp0 plumb mojave nomada up
ifconfig ipdptp1 plumb mojave nomanb up
path
      peer_system_name tamerlane
      require_authentication pap  #tells nomada that mojave
                                  #requires pap authentication
      pap_peer_id desert
      pap_peer_password oasis
path
      peer_system_name lawrence
      require_authentication chap  #tells nomadb that mojave
                                   #requires chap authentication
      chap_peer_name another\sdesert
      chap_peer_secret secret\soasis\swith\007bell
```

例 23–2 に示された mojave のリモートホスト nomada は、PAP と CHAP の両方を認証しようしています。

例 23–2 リモートホスト nomada 用のコード例

```
ifconfig ipdptp0 plumb tamerlane mojave up
path
      interface ipdptp0
      peer_system_name mojave
      will_do_authentication chap pap #nomada tells mojave
                                      #that it will do chap and
                                      #pap authentication
      pap_id desert
      pap_password oasis
      chap_name desert\srain
      chap_secret %$#@7&*(+|'P'12
```

例 23–3 に示された mojave のリモートホスト nomadb は、CHAP を認証しようしています。

例 **23–3**　リモートホスト nomadb 用のコード例

```
ifconfig ipdptp0 plumb nomadb mojave private up
path
     interface ipdptp0
     peer_system_name mojave
     will_do_authentication chap   #nomadb tells mojave that it
                                   #will do chap authentication
     chap_name another\sdesert
     chap_secret secret\soasis\swith\007bell
```

通常、CHAP と PAP の両方が構成ファイルに組み込まれていて、サーバーが認証を要求し、リモートホストが認証を行うのが、理想的な形です。ただし、この順序は逆も可能なためリモートホストの方が認証を要求することも可能です。CHAP シークレットは安全な手段で送付する必要があります。これは一般的に手動での解放を含みます。

新規の PPP リンクの起動と停止

PPP は、ブート時に自動的に起動されるようにすることも、コマンド行から手動で起動することもできます。

▼ 手動で PPP を起動する方法

通常は必要ありませんが、PPP を手動で起動することができます。

1. スーパーユーザーになります。
2. 次のように入力します。

```
# /etc/init.d/asppp start
```

▼ PPP が実行中であることを確認する方法

1. スーパーユーザーになります。
2. ps コマンドを実行します。

```
# ps -e | grep aspppp
```

grep の結果の出力に aspppd デーモンがリストされれば、PPP が実行中です。

3. 結果が表示されたら、リモート **PPP** リンクに到達できるかどうかを確認するために、次のように入力します。

```
# ping remote-host 300
```

この例の ping では、タイムアウト値が 5 分 (300 秒) に設定されています。このコマンドに対しては、「*remote-host* is alive」に類似した出力が表示されます。これとは異なる出力、たとえば「*remote-host* unreachable」などと表示された場合は、経路の構成が失敗したことを意味します。

4. ログファイルを調べて、構成にエラーがないかどうか検査します。

```
# tail /var/adm/log/asppp.log
```

構成時にエラーが見つかった場合は、asppp.log にエラーメッセージが記録されています。

障害追跡と問題解決については、490ページの「共通の確認事項」 を参照してください。

▼ PPP の停止方法

ネットワーク上での PPP 操作を停止するには、次の手順を実行します。

1. スーパーユーザーになります。

2. 次のように入力します。

```
# /etc/init.d/asppp stop
```

共通の確認事項

この節では、ご使用の PPP 設定の動作を確認するために行う必要があると思われるいくつかの共通の確認事項について説明します。

注 - これらの確認を行うためにはスーパーユーザーになる必要があります。

ハードウェアの検査

すべてのモデムケーブルと電源ケーブルがしっかりと接続されていることを確認します。PPP に問題が生じたときは、常に、モデム、ケーブル、シリアルカード、および電話回線を最初に検査してください。

▼ インタフェースの状態を確認する方法

PPP を起動したあとは、PPP インタフェース名だけを引数として指定した `ifconfig` を使用して、回線の現在の状態が監視できます。例 23–4 に示すのは、実行中の PPP リンクについての `ifconfig` のサンプル出力です。

注 - 特権 (root) ユーザーが `ifconfig` コマンドを発行した場合は、上記のようにマシンのアドレスが出力に表示されます。

---

1. スーパーユーザーになります。
2. 次のように入力します。

```
ifconfig ipdptp0
```

例 **23–4** ポイントツーポイントリンクに関する `ifconfig` の出力

```
nomadb# ifconfig ipdptp0
ipdptp0: flags=28d1<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST,UNNUMBERED> mtu 1500
```

(続く)

```
inet 129.144.111.26 --> 129.144.116.157 netmask ffff0000
ether 0:0:0:0:0:0
```

標準と動的のどちらのポイントツーポイントリンクの場合も、例 23–5 に示すような出力が得られます。

例 **23–5** マルチポイントリンクに関する ifconfig の出力

```
nubian# ifconfig ipd0

ipd0: flags=c1<UP,RUNNING,NOARP> mtu 1500
        inet 129.144.201.191 netmask ffffff00
        ether 0:0:0:0:0:0
```

ifconfig に UP と RUNNING が表示されない場合は、PPP が正しく構成されていないことを示します。ifconfig の詳細は、124ページの「ifconfig コマンド」と、ifconfig(1M) のマニュアルページを参照してください。

▼ 接続状態を確認する方法

ping コマンドを使用して、接続が up 状態であるか、または確立可能であるかを検査します。たとえば、次のような単純な往復テストを考えてみてください。

1. スーパーユーザーになります。
2. 次のように入力します。

```
# ping elvis
```

elvis はリモートホスト上の PPP インタフェースの名前です。結果の表示が次のとおりであったとします。

```
elvis is alive
```

この場合は、elvis との間でパケットを送受信できます。この結果が得られなかったとすれば、ローカルホストとリモートホストの間のどこかに、ルーティン

グに関する問題があります。ping についての詳細は、122ページの「ping コマンド」と、ping(1M) のマニュアルページを参照してください。

▼ インタフェースアクティビティを確認する方法

パケットが正しく送受信されているかどうかを検査するには、netstat コマンドを使用します。

1. スーパーユーザーになります。
2. 次のように入力します。

```
# netstat -i
```

126ページの「netstat コマンド」と netstat(1M) のマニュアルページを参照してください。

▼ ローカルルーティングテーブルを確認する方法

ローカルルーティングテーブルを表示するには、netstat コマンドを使用します。

1. スーパーユーザーになります。
2. 次のように入力します。

```
# netstat -r
```

次に出力例を示します。

```
Routing tables
Destination     Gateway    Flags  Ref    Use    Interface
------------ ------------ ----- ------ ------- ----------
sahara       deserted     UGH    0           0  ie1
karakum      labia        UGH    0           0  ie1
frodo        bilbo        UGH    1       12897  ipdptp0
route7       route7       UGH    0           0  ie0
eastgate     route71      UGH    0         158  ie0
backbone     pitstopbb    U      1       16087  ie1
```

(続く)

```
dresdenpc      route1       UG      0           0  ie1
loopback       localhost    U       2      113436  lo0
swan-bb        pitstop      U       406    146044  ie0
dallas2        route7       UG      0           0  ie0
trainingpc     route62      UG      0           0  ie1
```

到達可能なネットワークごとに、ルーティングテーブルエントリが存在することを確認します。特に、Interface の欄に示される PPP デバイスが、Gateway の欄に示される適切なホスト名と適合している必要があります。同様に、Gateway エントリは、Destination の欄の正しいエントリと適合している必要があります。

この条件が満たされていない場合は、静的ルーティングを使用しているのであれば、適正な静的ルートを追加します。

in.routed を使用してルートを追加する方法

in.routed によって動的ルーティングを使用しているときは、次の手順を行います。

1. スーパーユーザーになります。
2. 次のように入力して、in.routed が実行中であることを確認します。

```
# ps -e | grep route
```

それでもまだルーティングテーブルが正しくない場合は、スーパーユーザーになって次の手順に進みます。

3. ps -e から入手したプロセス **ID** を kill の引数として指定して、in.routed を終了します。たとえば、**1384** がプロセス **ID** であるとすれば、次のように入力します。

```
# kill 1384
```

4. 次のようにしてルーティングテーブルをフラッシュします。

```
# /usr/sbin/route -f
```

5. in.routed を再起動します。

```
# /usr/sbin/in.routed
```

アクセス権の検査

rsh を使用しようとして、Permission denied というメッセージが出力された場合は、リモートシステムの /etc/hosts.equiv ファイルまたは /.rhosts ファイルに、送信側システムのホスト名が含まれていないか、行 + が含まれていません。

パケットフローの検査

次にパケットフローを検査します。snoop コマンドを使用して、ネットワークからパケットや、各パケットの内容を観察します。例 23–6 に、snoop からの出力例を示します。

例 **23–6** snoop からの出力例

```
# snoop -d ipdptp0
Using device ipdptp0 (promiscuous mode)
corey -> pacifica7    RLOGIN C port=1019
       hugo -> ponc3          RPC R XID=22456455 Success
       ponc3 -> hugo          NFS C WRITE FH=1B29 at 32768

    commlab3 -> commlab4      TELNET R port=34148
    commlab4 -> commlab3      IP  D=129.144.88.3 S=129.144.88.4 LEN=46, ID=41925
    commlab3 -> commlab4      TELNET R port=34148
    commlab4 -> commlab3      ICMP Echo request
    commlab3 -> commlab4      ICMP Echo reply
    commlab4 -> commlab3      FTP C port=34149
    commlab4 -> commlab3      FTP C port=34149
    commlab3 -> commlab4      FTP R port=34149
    commlab4 -> commlab3      FTP C port=34149
```

出力の最初の行の Using device ipdptp0 に含まれている ipdptp0 というデバイス名は、ポイントツーポイント接続を示しています。

注 - snoop を使用して回線の状態を検査するには、リンクが up 状態にあり、トラフィックがある程度生成されている必要があります。

snoop は、ネットワークからパケットを取り込んで、その内容を表示します。snoop は、パケットフィルタモジュールとストリームバッファーモジュールの両方を使用して、ネットワークから効率的にパケットを取り込みます。取り込んだパケットは、受け取ると同時に表示することも、あとで見るためにファイルに保存しておくこともできます。

snoop は、単一行要約形式と複数行詳細形式のどちらでも、パケットを表示できます。要約形式の場合は、最高レベルのプロトコルに関するデータだけが表示されます。たとえば、NFS パケットについては NFS に関する情報だけが表示されます。その下位にある RPC、UDP、IP、Ethernet フレームの情報は抑止されますが、詳細形式オプションのどれかを選択した場合は表示されます。

snoop コマンドの詳細は、snoop(1M) のマニュアルページを参照してください。

PPP 診断機能を使用した障害追跡

モデム接続を正常に確立した後でリンクに問題がある場合は、PPP レベルの診断機能を使用した障害追跡を行うことができます。PPP レベルの診断機能は、リンクの動作状況に関する詳細情報を報告するので、どこに障害があるのかを突き止めるのに役立ちます。

診断情報を入手するには、debug_level 8 の行を asppp.cf ファイルの path セクションに追加します (データ通信に関する詳しい知識がある場合は、デバッグレベル 9 を用いれば、きわめて詳細な情報が得られます)。次に、PPP 診断機能を呼び出す構成ファイル例を示します。

```
ifconfig ipdptp0 plumb nomada nubian-ppp up
path
   interface ipdptp0
   peer_system_name nubian-ppp   #The name in the /etc/uucp/Systems file
   inactivity_timeout 300        #Allow five minutes before timing out
   debug_level 8                 #Start up PPP diagnostics for this link
```

asppp.cf ファイルについては、480ページの「/etc/asppp.cf 構成ファイルの編集」を参照してください。

▼ マシンに対する診断の設定方法

監視したいホストについて診断を設定するには、次の手順を行います。

1. スーパーユーザーになります。
2. /etc ディレクトリに移動します。
3. 現在の asppp.cf ファイルを編集して、path セクションに下記を追加します。

```
debug_level 8
```

4. アクセス権が必ず 600 に設定されるように、ファイルを保存します。
5. 現在の aspppd デーモンを終了し、再起動します。

```
# kill PID
# aspppd
```

PID は aspppd のプロセス ID です。

PPP は、/var/adm/log/asppp.log に診断情報を書き込みます。

第 24 章

PPP リファレンス

この章では PPP を使用するためのリファレンスを提供しています。次のトピックについて説明します。

UUPC データベース

マシンが PPP リンク上でダイアルできるようにするためには、その UUPC データベース内の次のファイルを編集する必要があります。

- /etc/uucp/Devices
- /etc/uucp/Dialers
- /etc/uucp/Systems

PPP の /etc/uucp/Devices の更新

/etc/uucp/Devices ファイルには、そのホストが使用するか、または認識していなければならない、すべての通信デバイスについてのエントリが含まれている必要があります。たとえば、あるマシンが US Robotics V.32bis モデムを PPP リンクの一部として使用している場合、/etc/uucp/Devices ファイルに次のようなエントリが入っていなければなりません。

```
# Use these if you have a USrobotics V.32bis modem on Port B.
ACUEC   cua/b - 9600 usrv32bis-ec
ACUEC   cua/b - 19200 usrv32bis-ec
ACUEC   cua/b - 38400 usrv32bis-ec
```

各 PPP エンドポイントマシンの Devices ファイル中に、それぞれのモデムを記述しているエントリがあることを確認してください。/etc/uucp/Devices についての詳細は、556ページの「UUCP /etc/uucp/Devices ファイル」を参照してください。

PPP の /etc/uucp/Dialers の更新

/etc/uucp/Dialers ファイルには、PPP エンドポイントマシンに接続しているモデムとの会話を記述するエントリが含まれている必要があります。たとえば、US Robotics V.32bis モデムを PPP リンクとして使用する場合、このエントリは次のようになります。

```
usrv32bis-ec =,-,  "" \dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2\r\c OK\r
\EATDT\T\r\c CONNECT\s14400/ARQ STTY=crtscts
```

このエントリ内の最初のパラメータ、usrv32bis は /etc/uucp/Devices ファイル内の最後のパラメータに対応します。このエントリの残りの部分には、モデムが送る文字、モデムが受け取ると予期している文字などが記述されています。表 27–5 に、Dialers ファイルの中で使用する制御コードの定義を示してあります。

リンク上の各ダイヤルアウトエンドポイントに接続しているモデムについて、Dialers ファイル内にエントリが 1 つずつあることを確認してください。特定のモデムの会話が正しいかどうか確信がない場合は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』および、そのモデムの操作マニュアルの説明を参照してください。

PPP の /etc/uucp/Systems の更新

/etc/uucp/Systems ファイルには、ローカルホストがダイヤルアウトできる各マシンについてのエントリが入っています。各エントリには、リモートホストの電話番号や、回線速度などの情報が入っています。たとえば、図 23–1 に示したホスト nomadb では、ダイヤルインサーバーについてのエントリは次のような内容になります。

```
nubian-ppp  Any ACUEC 38400 5551212 "" P_ZERO ""
\r\n\c login:-\r\n\c-login:-\r\n\c-login:-
EOT-login: bnomad password: Secret-Password
```

最初のフィールドに示されているのはサーバーのホスト名である nubian-ppp で、これは、asppp.cf ファイルのキーワード peer_system_name に使用されます。ACUEC と 38400 はデバイスと速度を示し、これは、/etc/uucp/Devices ファイルからエントリを選択するために使用されます。その後の部分には、nomadb がダイヤルインするマシンの電話番号、nomadb がログインするために使用するログイン名などの情報があります。Systems ファイルに指定する必要のあるパラメータについては、547ページの「UUCP /etc/uucp/Systems ファイル」で詳しく説明します。

構成内の各リモートホストには、ダイヤルインサーバーについてのエントリを追加する必要があります。/etc/uucp/Systems ファイルには、そのホストが UUCP 通信でダイヤルアウトする他のマシンについてのエントリや、他の PPP ダイヤルインサーバーについてのエントリを一緒に入れることができます。

ダイヤルインサーバーがリモートホストに直接ダイヤルアウトを行う場合は、それらのリモートホストのそれぞれを記述するエントリを Systems ファイルに追加する必要があります。

/etc/asppp.cf 構成ファイル

/etc/asppp.cf 構成ファイルは、エンドポイントマシン上にある PPP リンクマネージャに、リンクの反対側にあるマシンに関する情報、またはマルチポイントリンク (または動的ポイントツーポイントリンク) の反対側にあるマシンに関する情報を提供します。このマシンがブートすると、リンクマネージャはこの情報を使用して、リモートエンドポイントとの通信を確立し維持します。

基本構成ファイルの各部分

基本的な asppp.cf 構成ファイルには、少なくとも 2 つのメインセクションが含まれていなければなりません。それは、1 個の ifconfig 行と、少なくとも 1 つの path セクションです。これに加えて defaults セクションも含めることができます。このセクションは、エンドポイントについてデフォルト値を設定したい場合に使用します (defaults セクションで使用するキーワードの説明については、524 ページの「構成キーワード」を参照してください)。

例 24–1 に示す基本構成ファイルは、ダイヤルインサーバーとの間にポイントツーポイントリンクを確立するリモートホスト用として作成されたものです。

例 24–1 基本構成ファイル

```
ifconfig ipdptp0 plumb nomada nubian-ppp up
   path
      interface ipdptp0
      peer_system_name nubian-ppp      # The name in the /etc/uucp/Systems file
      inactivity_timeout 300           # Allow five minutes before timing out
```

asppp.cf ファイルの ifconfig セクション

asppp.cf ファイルには、次の構文の ifconfig セクションを含める必要があります。

ifconfig *interface-number* plumb *local-machine remote-machine* up

各フィールドについて説明します。

- ifconfig – リンクマネージャに、ifconfig コマンドを実行し、PPP インタフェースの構成を始めるよう指示します。
- *interface-number* – PPP インタフェースを識別します。ポイントツーポイントリンクの場合は ipdptp*n*、マルチポイントリンクの場合は ipd*n* (*n* はインタフェースの番号で置き換えます)。
- plumb – IP がインタフェースを認識できるようにする、ifconfig のオプション。
- *local-machine* – ローカルエンドポイントの名前を指定します。これには、ローカルホスト名か IP アドレスを使用できます。
- *remote-machine* – リモートエンドポイントの名前を指定します。これには、リモートホスト名か IP アドレスを使用できます。

- up – 記述したインタフェースに up のマーク付けをする、ifconfig のオプション。

リンクマネージャはまず、ローカルホストで ifconfig コマンドを実行して、ipdptp0 ポイントツーポイントインタフェースを構成します。ipdptp0 の中の 0 は、インタフェースのデバイス番号を示します。plumb オプションは、IP が ipdptp0 インタフェースを認識するのに必要な各種の操作を行います。nomada はローカルホストの名前です。nubian-pppは、nomada がポイントツーポイントリンクを介して接続するダイヤルインサーバーの名前です。ifconfig オプション up は、ipdptp0 インタフェースに up のマークを付けます。

注 - ifconfig についての詳細は、490ページの「インタフェースの状態を確認する方法」と、ifconfig(1M) のマニュアルページを参照してください。

---

asppp.cf ファイルの path セクション

構成ファイルの path セクションは、リモートエンドポイントの名前と、エンドポイントマシン間を結ぶインタフェースの名前を、リンクマネージャに指示します。path セクションには、少なくとも下記の行が必要です。

```
path
    interface interface-number
    peer_system_name endpoint-name
```

interface キーワード

このキーワードは PPP インタフェースを定義します (ipdptp*n* か ipd*n* のどちらか)。例 24–1 では、path セクションに次の情報があります。

```
interface ipdptp0
peer_system_name nubian-ppp
```

この interface キーワードは、ローカルエンドポイント nomada が、この path セクションの記述に従ってリモートエンドポイントと通信するのに使用するポイントツーポイントインタフェースが ipdptp0 であることを表します。このキーワードは、peer_system_name をインタフェースに結び付けています。

peer_system_name キーワード

リモートホストなどのようなダイヤルアウトマシンでは、peer_system_name キーワードは、リモートエンドポイントのホスト名を引数としてとります。これは、/etc/uucp/Systems の中で指定されたリモートエンドポイントの名前です。この名前は、対応する ifconfig 行のホスト名と同じでなくてもかまいません。

注 - ダイヤルインサーバーの場合は、peer_system_name キーワードへの引数の値は異なります。詳細は、例 24–1 を参照してください。

例 24–1 では、peer_system_name は、このリンクの反対側にあるリモートエンドポイントが、ダイヤルインサーバー nubian-ppp であることを示しています。リンクマネージャは、asppp.cf ファイルを読んだあとで、/etc/uucp/Systems ファイルの中で nubian-ppp についてのエントリを見つけます (Systems ファイルには、リモートエンドポイントとの通信を設定する方法や、そのマシンの電話番号などが含まれているということを思い出してください。499ページの「PPP の /etc/uucp/Systems の更新」を参照してください)。

inactivity_timeout キーワード

inactivity_timeout キーワードは省略可能です。このキーワードは、指定した時間が経過するまでの期間は、リンクが未使用状態であっても構わないことをリンクマネージャに指示します。その期間が経過すると、リンクマネージャは自動的にリンクを切り離します。デフォルトの時間は 2 分です。未使用期間として別の時間を指定したい場合でない限り、inactivity_timeout を使用する必要はありません。

その他のキーワード

asppp.cf ファイルには、上記以外にも、エンドポイントマシンによる通信の方法を定義するためのキーワードがいくつかあります。これらのキーワードについては、524ページの「構成キーワード」に詳しい説明があります。

マルチポイントダイヤルインサーバーの構成ファイル

マルチポイントダイヤルインサーバーの asppp.cf ファイルの場合も、基本的なセクションはポイントツーポイントリンクの場合と同じで、1 個の ifconfig セクションと、少なくとも 1 つの path セクションのほかに、必要に応じて指定する defaults セクションがあります。

例 24–2 は、例 24–1 に示したダイヤルインサーバー nubian の構成ファイルです。

例 24–2 マルチポイントダイヤルインサーバーの構成ファイル

```
ifconfig ipd0 plumb nubian-ppp up

path
   interface ipd0
   peer_system_name tamerlane  # The user name this remote
                               # machine logs in with when it
                               # dials this server
    peer_ip_address nomada
                               # nomada is a remote machine that
                               # dials in to this server

# nomadb is another remote machine that dials in to nubian

path
   interface ipd0
   peer_system_name lawrence
   peer_ip_address nomadb

# nomadc is another remote machine that dials in to nubian

path
   interface ipd0
   peer_system_name azziz
   peer_ip_address nomadc
```

マルチポイントダイヤルインサーバーの ifconfig セクション

マルチポイントダイヤルインサーバーの場合の ifconfig セクションは、ポイントツーポイントリンクの場合とはやや構文が異なります。構文は次のとおりです。

```
ifconfig ipdn plumb server-name up
```

最も大きな相違点は、ifconfig への引数として宛先エンドポイントを指定しないという点です。代わりに、リンクマネージャは、asppp.cf ファイルの path セクションからこの情報を拾いだします。

例 24–2 では、リンクマネージャは、まずダイヤルインサーバーで ifconfig コマンドを実行して、マルチポイントインタフェース ipd0 を構成します。ipd0 の中の 0 は、インタフェースのデバイス番号を示します。plumb オプションは、IP が ipd0 インタフェースを認識するために必要な各種の操作を行います。ifconfig オプション up は、ipd0 インタフェースに up のマークを付けます。

注 - サブネットを使用する場合は、ifconfig 行に netmask + パラメータの指定が必要です。

マルチポイントダイヤルインサーバーの path セクション

asppp.cf ファイルの path セクションは、リモートエンドポイントの名前と、エンドポイントマシンをリンクするインタフェースの名前を、リンクマネージャに指示します。ただし、マルチポイントダイヤルインサーバーでは、複数の path セクションを設けることができます。また、キーワードへの引数のいくつかは、マルチポイントリンクでは使い方が異なります。

```
path
    interface interface-number
    peer_system_name endpoint-username
    peer_ip_address endpoint-hostname
```

path セクションは、ダイヤルインサーバーが接続を確立する相手となる各可搬エンドポイントについて、1 つずつ定義する必要があります。

interface キーワード

マルチポイントダイヤルインサーバーの場合は、interface キーワードは PPP インタフェース ipd*n* を定義します。このインタフェースを介してサーバーと通信するすべてのエンドポイントについて、同じ PPP インタフェースを path セクションに指定する必要があります。

peer_system_name キーワード

ダイヤルインマシンの場合の peer_system_name キーワードは、ダイヤルアウトマシンの場合と引数が少々異なります。ダイヤルインサーバーの場合は、この引数は、リモートホストがサーバーとの通信を確立しようとするときに使用するログイン名です。このユーザー名は、すでにサーバーの /etc/passwd ファイル内に存在しているものでなければなりません。ログインサービスは、この名前を読み取ると、/etc/passwd ファイルと /etc/shadow ファイルの中のユーザー名とを検証して、通信を可能にします。

次に示す、例 24–2 の抜粋を見てください。

```
path
     interface ipd0
     peer_system_name scarlett
     peer_ip_address nomadc
```

ここでは、peer_system_name への引数は scarlett です。これは、nomadc が nubian-ppp にログインするときに、scarlett というログイン名を使用することを示しています。

peer_ip_address キーワード

peer_ip_address キーワードは、マルチポイントリンクの場合は必須です。このキーワードは、引数としてリモートエンドポイントのホスト名または IP アドレスを受け取ります。上記の例では、peer_ip_address キーワードの引数はホスト名 nomads です。

その他のキーワード

asppp.cf ファイルには、上記以外にもエンドポイントマシンによる通信の方法を定義するためのキーワードがいくつかあります。これらのキーワードについては、524ページの「構成キーワード」に詳しい説明があります。

PPP の障害追跡

モデムの接続を正常に確立したあとでリンクに問題が発生した場合、PPP レベル診断機能を使用して障害追跡を行うことができます。PPP レベル診断機能は、リンクの動きに関する詳細な情報を報告するため、問題がどこに発生しているかを発見するのに役立ちます。

診断出力の分析

PPP が正常に実行されているときに、asppp.log ファイルには、通常の出力のほかに診断情報が含まれています。この節では、診断メッセージの意味について説明します。ここに該当する出力がない場合は、RFC 1331 を参照してください。

ホストとモデムの設定

ローカルホストがモデムに構成情報を送り、モデムがリモートホストにダイヤルしようとしたときに発生するメッセージについて説明します。これらの初期の動作は、実際には UUCP デーモンが取り扱います。これらの動作は、非同期 PPP 通信の UUCP 部分と考えることができます (UUCP についての詳細は、第 25 章を参照してください)。

次の 2 つのメッセージは、セッションの始めに常に表示されます。これは、aspppd デーモンが正常に起動されたことを示します。

```
11:53:33 Link manager (1057) started 04/14/94
11:53:33 parse_config_file: Successful configuration
```

次の行は、パケットがローカルホストの ipdptp0 インタフェースに送られたことを示しています。これは、ダイヤルアウトが正常に行われたかどうかを判断するのに役立ちます。たとえば、リモートマシンの ping を試みたときに、asppp.log 内にこのメッセージがないとすれば、ルーティングの問題が原因でパケットが失われていると考えられます。

次に、UUCP は、/etc/uucp/Systems ファイル内のチャットスクリプトの中にある Ppac7 に一致するエントリを探します。そして、デバイスタイプが ACUTEC であるエントリが見つかったことを報告します (Systems ファイルについての詳細は、547ページの「UUCP /etc/uucp/Systems ファイル」を参照してください)。

```
11:53:46 process_ipd_msg: ipdptp0 needs connection
conn(Ppac7)
Trying entry from '/etc/uucp/Systems' - device type ACUTEC.
```

UUCP は次に、/etc/uucp/Devices ファイルから ACUTEC ダイヤラに関するダイヤル情報を探します。この情報が見つかると、UUCP は、ローカルホストの該当するシリアルポートをオープンし、その速度を 9600 に設定します (/etc/uucp/Devices についての詳細は、556ページの「UUCP /etc/uucp/Devices ファイル」を参照してください)。

```
Device Type ACUTEC wanted
Trying device entry 'cua/a' from '/etc/uucp/Devices'.
processdev: calling setdevcfg(ppp, ACUTEC)
fd_mklock: ok
fixline(8, 9600)
gdial(tb9600-ec) calle
```

UUCP は、/etc/uucp/Dialers ファイルの中に tb9600 というエントリを確認して、次のメッセージを送り出します。

```
Trying caller script 'tb9600-ec' from '/etc/uucp/Dialers'
expect: (''')
```

ホストは 2 秒間待ってから、モデムのレジスタを設定します。下記のログに示される情報は、個々のモデムに固有のものです。これは /etc/uucp/Dialers ファイルからの情報をもとにしています。

```
got it
sendthem (DELAY)
APAUSE
APAUSE
APAUSE
T&D2E1V1X1Q0S2=255S12=255S50=6S58=2^M<NO CR>)
```

次の行は、モデムとホストマシンとの間のダイアログです。expect (OK^M) は、モデムが「了解」を送ることを予期していることを意味します。2 行目の終わりの got it という語句は、ホストがモデムから「了解」メッセージを受け取ったことを意味します。

```
expect: (OK^M)
AAAT&D2E1V1X1Q0S2=255S12=255S50=6S58=2^M^M^JOK^Mgot it
```

次にホストは下記の文字列をモデムに送り、実際にはモデムがダイヤリングを行います。2 行目の電話番号は、/etc/uucp/Systems ファイル内のリモートホストに関するエントリから検索されます。

```
sendthem (ECHO CHECK ON
A^JATTDDTT99003300887744^M^M<NO CR>)
```

expect で始まる行は、ローカルホストが、モデムから 9600 bps の速度であるという応答を受け取ることを予期していることを意味します。その次の行は、モデムが応答したことを示しています。

```
expect: (CONNECT 9600)
^M^JCONNECT 9600got it
```

次の行は、リンク上でハードウェアフロー制御が開始されたことを示しています。ホストは、フロー制御情報を /etc/uucp/Dialers ファイルから入手します。

```
STTY crtscts
```

次に示す一連のメッセージは、ローカルホストが、リモートホストから標準的な UNIX ログインプロンプトが送られてくるのを待っていることを示しています。

```
getty ret 8
expect: (''''')
got it
sandiest (^J^M)
expect: (login:)
```

次のメッセージは、ローカルホストがリモートからのログインプロンプトを受け取ったことを示します。ローカルホストは、リモートホストについての /etc/uucp/Systems エントリ内のチャットスクリプトから、該当するログインシーケンスを検索します。このシーケンスは Ppong^M で、リモートホストがログインするために必要です。

```
^M^J^M^Jlogin:got it
sendthem (Ppong^M)
```

次のメッセージでは、ローカルホストは、リモートホストからの ssword プロンプトを待ちます。このプロンプトを受け取ると、ローカルホストは、リモートホストに関する /etc/uucp/Systems エントリ内のチャットスクリプトから検索したパスワードを送ります。

```
expect: (ssword:)
login: Ppong^M^JPassword:got it
```

次のメッセージは、ダイヤリングとモデム接続が正常に完了したことを示しています。

```
sendthem (ppptest1^M)
call cleanup(0)^M
```

ローカルホストとリモートホストの間の通信

この時点で、ローカルホストとリモートホストの間のリンクが確立され、PPP 通信が開始されます。

セッションのこの部分の最初のいくつかの行は、構成要求 (Config-Req) です。これは、リモートホストに送られる最初の PPP パケットです。構成要求は、リンク制御プロトコル (LCP) パケットの一例です。このパケットは、構成を設定することを要求し、エンドポイントマシン間の PPP リンクを設定します。例 24–3 は、サンプルの構成要求を示します。

例 **24–3** 構成要求

```
11:54:20 004298 ipdptp0 SEND PPP ASYNC 29 Octets LCP Config-Req
ID=4c LEN=24 MRU=1500 ACCM=00000000 MAG#=69f4f5b2 ProtFCOMP
AddrCCOMP
```

次に、構成要求について説明します。

- `11:54:20` – タイムスタンプフィールド。パケットが送られた時刻を示す
- `004298` – パケットの番号
- `ipdptp0` – 使用するネットワークインタフェース
- `SEND PPP ASYNC` – モデムが非同期 PPP を送信していることを示す
- `29 Octets` – ホストが送ったデータの量
- `LCP` – 送信するパケットタイプ
- `ID=4c` – パケットに関連付けられている識別子。これは実際にはパケットの一部
- `LEN=24` – パケットの LCP 部の長さ

残りの項目は、ホスト間でのネゴシエーションを必要とするオプションのリストです。

- `MRU=1500` – 最大受信単位 (MRU)。呼び出し側ホストがリモートホストから受信できる最大パケットサイズ
- `ACCM=00000000` – 非同期文字マップ (ACCM)。送信でエスケープする制御文字をリモートホストに知らせるために送られるマスク
- `MAG#=69f4f5b2` – マジックナンバフィールド。ループバック検出メカニズムに使用される
- `ProtFCOMP AddrCCOMP` – フレームヘッダーの特定の部分 (プロトコルフィールド、アドレスフィールド) の圧縮をリモートホストに要求する

次に示す一連のメッセージは、無効な PPP パケットを報告しています。これらのパケットは、実際には UNIX テキストを送信しようとしているリモートホストから送られてきたものです。これは PPP に問題があることを示すものではありません。

```
11:54:20 004299 ipdptp0 RECEIVE {Invalid ppp packet}PPP ASYNC 7
Octets [BAD FCS] {Unrecognized protocol:    1}

11:54:20 004299 ipdptp0 RECEIVE PPP ASYNC 73 Octets [BAD FCS]
```

(続く)

```
{Unrecognized protocol:    880a}
```

次のパケットでは、ローカルホストはリモートホストからの構成要求を受け取り、さらに別の構成要求を送ります。これら 2 つのパケットは、ID フィールド以外の部分はどちらも同じです。2 つのパケットは ID フィールドにより区別されます。

```
11:54:21 004301 ipdptp0 RECEIVE PPP ASYNC 29 Octets LCP Config-
Req  ID=35 LEN=24 MRU=1500 ACCM=00000000 MAG#=a8562e5f ProtFCOMP
AddrCCOMP

11:54:21 004302 ipdptp0 SEND PPP ASYNC 29 Octets LCP Config-Req
ID=4d LEN=24 MRU=1500 ACCM=00000000 MAG#=69f4f5b2 ProtFCOMP
AddrCCOMP
```

次のパケットでは、ローカルホストは、リモート要求に対する確認として、構成肯定応答 (Config-ACK) を送ります。

```
11:54:21 004303 ipdptp0 SEND PPP ASYNC 29 Octets LCP Config-ACK
ID=35 LEN=24 MRU=1500 ACCM=00000000 MAG#=a8562e5f ProtFCOMP
AddrCCOMP
```

ローカルホストは、リモートホストからの構成要求 (Config-Req) を受け取ります。

```
11:54:21 004304 ipdptp0 RECEIVE PPP ASYNC 29 Octets LCP Config-
Req  ID=36 LEN=24 MRU=1500 ACCM=00000000 MAG#=a8562e5f ProtFCOMP
AddrCCOMP
```

次のパケットでは、ローカルホストはリモートホストから送られてきた第 2 のパケットを確認し、リモートホストの肯定応答を受け取ります。

```
11:54:21 004305 ipdptp0 SEND PPP ASYNC 29 Octets LCP Config-ACK
ID=36 LEN=24 MRU=1500 ACCM=00000000 MAG#=a8562e5f ProtFCOMP
```

```
AddrCCOMP

11:54:21 004306 ipdptp0 RECEIVE PPP ASYNC 29 Octets LCP Config-
ACK  ID=4d LEN=24 MRU=1500 ACCM=00000000 MAG#=69f4f5b2 ProtFCOMP
AddrCCOMP
```

次のパケットでは、ローカルホストは IP 伝送に関するパラメータについてのネゴシエーションを行います。LEN=16 はパケットサイズを表します。VJCOMP は、Van Jacobson のヘッダー圧縮を示しています。IPADDR の後にあるのは呼び出し側ホストの IP アドレスです。

```
11:54:21 004307 ipdptp0 SEND PPP ASYNC 21 Octets IP_NCP Config-
Req  ID=4e LEN=16 VJCOMP MAXSID=15 Sid-comp-OK IPADDR=192.9.68.70
```

次のパケットは、ローカルホストがリモートホストから、IP アドレスを含む IP 構成を受け取ったことを示しています。

```
11:54:22 004308 ipdptp0 RECEIVE PPP ASYNC 21 Octets IP_NCP
Config-Req  ID=37 LEN=16 VJCOMP MAXSID=15 Sid-comp-OK
IPADDR=192.9.68.71
```

ローカルホストは次の ACK をリモートホストに送り、リモートホストからの ACK を受け取ります。

```
11:54:22 004309 ipdptp0 SEND PPP ASYNC 21 Octets IP_NCP Config-
ACK  ID=37 LEN=16 VJCOMP MAXSID=15 Sid-comp-OK IPADDR=192.9.68.71

11:54:22 004310 ipdptp0 RECEIVE PPP ASYNC 21 Octets IP_NCP
Config-ACK  ID=4e LEN=16 VJCOMP MAXSID=15 Sid-comp-OK
IPADDR=192.9.68.70
```

次の最初のメッセージは、リンク上で IP が起動されたことを示しています。第 2 のメッセージは、ローカルホストがリンクを介して IP トラフィックを送信していることを示しています。

```
11:54:22 start_ip: IP up on interface ipdptp0, timeout set for
120 seconds

11:54:24 004311 ipdptp0 SEND PPP ASYNC 89 Octets IP_PROTO
```

次の最初のメッセージでは、ローカルホストはリモートホストからの IP トラフィックを受け取ります。そのあとのメッセージは、アイドルタイムアウトが原因でインタフェースが切り離されたことを示しています。

```
11:54:25 004312 ipdptp0 RECEIVE PPP ASYNC 89 Octets IP_PROTO
11:56:25 process_ipd_msg: interface ipdptp0 has disconnected
11:56:25 disconnect: disconnected connection from  ipdptp0
```

次のメッセージからは、終了シーケンスを開始します。最初のメッセージは、リモートホストが IP 層を終了するためのパケットを送ったことを示しています。第 2 のメッセージは、終了要求に対するローカルホストの肯定応答です。

```
11:56:25 004313 ipdptp0 RECEIVE PPP ASYNC 9 Octets IP_NCP Term-
REQ  ID=38 LEN=4

11:56:25 004314 ipdptp0 SEND PPP ASYNC 9 Octets IP_NCP Term-ACK
ID=38 LEN=4
```

ローカルホストは、LCP 層の終了要求を受け取ります。第 2 のメッセージはその要求に対する肯定応答であり、その結果正常なシャットダウンが行われます。

```
11:56:25 004315 ipdptp0 RECEIVE PPP ASYNC 9 Octets LCP Term-REQ
ID=39 LEN=4

11:56:25 004316 ipdptp0 SEND PPP ASYNC 9 Octets LCP Term-ACK
ID=39 LEN=4
```

次のメッセージはリンクが閉じられたことを示しています。

```
11:56:29 004317 ipdptp0 PPP DIAG CLOSE
```

動的に割り当てられた PPP リンク

動的ポイントツーポイントリンクを持つダイアルインサーバーを使用すると、サイトでポイントツーポイント通信の持つメリットがすべて利用できます。第 21 章では、この構成タイプについて説明しています。このタイプは、ポイントツーポイントリンクを必要に応じて動的に割り当てる、1 台以上のダイアルインサーバーと通信するリモートホストから構成されています。この節全体で次に示す構成例を使用します。

動的割り当てリンクの場合のアドレス指定に関する必要事項

動的割り当て PPP リンクを使用する各マシンについて、/etc/inet/hosts ファイルにホスト情報を追加する必要があります。PPP エンドポイントの IP アドレスについては次の規則があります。

- ダイヤルインサーバーの場合は、そのサーバーの一次ネットワークインタフェースの IP アドレス (たとえば le0、smc0 など) を、動的リンクのアドレスとして使用する必要があります。
- 動的リンクでは、各リモートホストに IP アドレスを割り当てる (静的リンクの場合) 必要はありません。ただし、サーバー上のポイントツーポイント IP インタフェースのそれぞれにリモート IP アドレスを割り当てる必要があります。使用可能な IP インタフェースの数は、サーバーに接続されたモデムの数と一致しま

す。たとえば、モデムが 3 つある場合、ポイントツーポイント IP インタフェースと IP アドレスが 3 つずつ必要です。

- クライアント上で ifconfig コマンドを正しく実行するには、ダミーの IP アドレスを入れなければなりません。PPP が起動すると、このアドレスはクライアントの IP インタフェースに割り当てられたローカル IP アドレス用のプレースホルダとして機能します。

注 - IP インタフェースに割り当てられるリモート IP アドレスに制限はありません。ただし、明確にするため、同じサブネットに属する IP アドレスだけを入れるのが最適です。

動的リンクの場合の hosts データベースの更新

動的リンク構成に含まれるすべてのマシンで、hosts データベースを更新する必要があります。

その他のファイルに関する考慮事項

次に行う手順として、/etc/passwd ファイルと /etc/shadow ファイルを編集します。動的リンク構成の場合も、リモートホスト対マルチポイントダイヤルインサーバー構成の場合と同じ手順で、これらのファイルを編集します。/etc/passwd ファイルと /etc/shadow ファイルについての詳細は、479ページの「/etc/passwd ファイルの修正」を参照してください。

動的リンクの場合の asppp.cf の編集

動的リンク構成用の asppp.cf 構成ファイルには、リモートホストに関する情報と、PPP リンクに使用するインタフェースに関する情報が含まれていなければなりません。ダイヤルインサーバーがブートした後、リモートエンドポイントからサーバーが呼び出されるたびに、リンクマネージャはこの情報を使用して通信を確立します。

動的リンクを持つリモートホスト

リモートホスト用の asppp.cf 構成ファイルは、500ページの「基本構成ファイルの各部分」で説明したファイルと同じですが、パラメータ *negotiate_address* が追加されている点が異なります。

```
ifconfig ipdptp0 plumb dummy mojave up
path
    interface ipdptp0
    peer_system_name mojave-ppp
    connectivity_timeout 300
    negotiate_address on
```

negotiate_address パラメータは、ローカル IP アドレスの割り当てがネゴシエーションによって取得されて動的に割り当てられているかを示します。設定が「on」の場合、サーバーから供給された IP アドレスが、接続中にクライアントのローカルアドレスとして使用されます。

動的リンクを持つダイヤルインサーバー

ダイヤルインサーバーが着信パケットを受信すると、リンクマネージャは構成ファイルの path セクションを読んで、リモートエンドポイントを識別し、使用するインタフェースを決定します。例 24–4 に示す構成ファイルには、インタフェースキーワードは含まれていません。代わりに、リンクマネージャは、defaults セクションに設定されているインタフェース情報を使用します。

動的割り当てリンクを持つダイヤルインサーバー用の asppp.cf 構成ファイルは、例 24–4 のようになります。

例 24–4 動的割り当てリンクを持つサーバー用の構成ファイル

```
ifconfig ipdptp0 plumb mojave clienta down
ifconfig ipdptp1 plumb mojave clientb down
ifconfig ipdptp2 plumb mojave clientc down

# This means grab whatever interface is available (not in use)
defaults
     interface ipdptp*

# Each path specifies a machine that might dial up / log
# in to this server
```

(続く)

```
path
    peer_system_name tamerlane   # nomada uses the login name
                                 # tamerlane

path
    peer_system_name lawrence    # nomadb uses the name lawrence
                                 # for login

path
    peer_system_name nomadc
```

動的リンクを持つサーバー用の ifconfig セクション

動的割り当てリンクを持つダイヤルインサーバー用の ifconfig セクションの構文は、次のとおりです。

ifconfig ipdptp*n* plumb *server-name client-address* down

例 24–4 には、3 つの ifconfig 行があり、それぞれポイントツーポイントインタフェースを初期化しています。

```
ifconfig  ipdptp0  plumb  mojave  clienta  down
ifconfig  ipdptp1  plumb  mojave  clientb  down
ifconfig  ipdptp2  plumb  mojave  clientc  down
```

動的リンクを持つサーバー用の defaults セクション

動的割り当てリンクを構成するときに、asppp.cf ファイルに defaults セクションを含めることができます。このセクションでは、その後に asppp.cf ファイル内に *keyword* が現れたときに、*keyword* に代入するデフォルトの値を設定します。defaults セクションの構文は次のとおりです。

```
default
      keyword
```

例 24–4 では、キーワード interface を使用して ipdptp* をインタフェースとして定義することにより、動的リンクを指定しています。ワイルドカードを示すアスタリスクは、ifconfig セクションで定義されている任意の使用可能な ipdptp インタフェースを使用するよう、リンクマネージャに指示しています。したがって、

サーバー mojave のリンクマネージャは、ipdptp0、ipdptp1、ipdptp2 のうち、down として構成されている最初のインタフェースを使用します。

動的リンクを持つサーバー用の path セクション

動的リンクを持つサーバー用の構成ファイルには、そのサーバーとの接続の確立が許されているすべてのリモートホストについての path セクションが含まれていなければなりません。path セクションの構文は次のとおりです。

```
path
    peer_system_name endpoint-username
```

interface キーワードは、path セクションの中で定義されていません。これは、この値が defaults セクションで定義されているからです。 この場合の peer_system_name キーワードと peer_ip_address キーワードの意味は、マルチポイントサーバー用の構成ファイルの場合と同じです。詳細は、504ページの「マルチポイントダイヤルインサーバーの path セクション」を参照してください。

その他のキーワード

asppp.cf ファイルでは、上記の他に、エンドポイントがどのように通信するかを定義するためのキーワードをいくつか指定できます。これには、524ページの「構成キーワード」で説明するセキュリティキーワードも含まれます。

仮想ネットワークの構成

仮想ネットワークは、それぞれ離れた場所にあるいくつかのスタンドアロンコンピュータを、互いに PPP マルチポイントリンクで接続したものです。仮想ネットワークの概念については、449ページの「仮想ネットワーク」で紹介しました。この節では、仮想ネットワークを構成する方法について説明します。

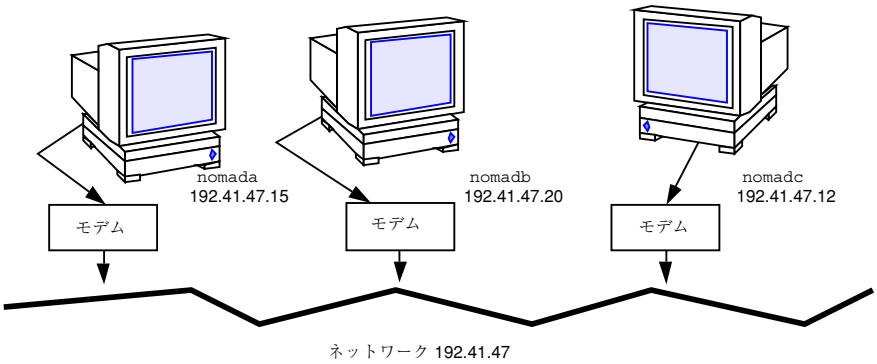


図 24–1　仮想ネットワーク例

図 24–1 に示すネットワークは、3 つの単独コンピュータで構成されています。ネットワークの各メンバーは、マルチポイント PPP リンクを介して他のメンバーに接続しています。したがって、このようなネットワークを作成するには、ネットワーク管理者 (リモートロケーションの他のネットワーク管理者である場合もあります) は、関与する各ホストでマルチポイント PPP リンクを構成する必要があります。

マルチポイントリンクの構成には、マルチポイントダイヤルインサーバーの場合と同じ一般的な手順を用います。この手順については、474ページの「構成プロセスの概要」に説明があります。ただし、仮想ネットワークには独自の必要条件がいくつかあり、それに従ってネットワーク内の各ホストを構成する必要があります。

仮想ネットワークの場合のアドレス指定に関する必要事項

仮想ネットワーク内の各マシンについて、`/etc/hosts` ファイルにホスト情報を追加する必要があります。PPP エンドポイント用に使用する IP アドレスを入力するときは、次の規則に従ってください。

- ポイントツーポイントリンクには PPP 固有の IP アドレスを指定する。物理ネットワーク内でまだ構成されていないマシンの場合は、PPP リンク用の IP アドレスを作成する必要がある。このアドレスが、ホストの一次ネットワークインタフェースになる
- 仮想ネットワークのネットワーク番号を作成する。詳細は、466ページの「PPP リンクへのネットワーク番号の割り当て」を参照

hosts データベースと networks データベースの更新

最初に行う手順としては、仮想ネットワークに関する情報によって、hosts データベースと networks データベースを更新します。

仮想ネットワークの場合の /etc/inet/hosts ファイル

各マシンの /etc/inet/hosts ファイルには、このホストからアクセスできるすべてのネットワークメンバーに関するアドレス指定情報が含まれている必要があります。たとえば、図 24–1 に示したネットワーク内の各ホストは、次のような情報を持っている必要があります。

```
# Internet host table
#
127.0.0.1           localhost loghost
192.41.47.15        nomada
192.41.47.20        nomadb
192.41.47.12        nomadc
```

仮想ネットワークの場合の /etc/inet/networks ファイル

仮想ネットワークは一意な IP アドレスを必要とするので、このアドレスを networks データベースに入力する必要があります。たとえば、図 24–1 に示したネットワークの番号は 192.41.47 です。さらに、このネットワーク上のホストが他のネットワークと通信する必要がある場合は、このネットワークを InterNIC のアドレス指定機関に登録する必要があります。networks データベースの編集方法については、156ページの「networks データベース」を参照してください。

仮想ネットワーク上の各ホストは、ネットワークのアドレスが入ったエントリを、/etc/inet/networks ファイル中に持っている必要があります。たとえば、ネットワーク 192.41.47 の各ホストは、/etc/inet/networks の中に次のようなエントリを持っている必要があります。

```
# Internet networks
#
# arpanet   10          arpa
# ucb-ether 46          ucbether
#
# local networks
loopback    127
ppp         192.41.47   #remote sales offices
```

その他のファイルの構成

次に行う手順としては、UUCP データベース、/etc/passwd ファイル、/etc/shadow ファイルを編集します。仮想ネットワーク内のマシンについてこれらのファイルを編集する方法は、マルチポイントダイヤルインサーバー構成の場合と同じです。UUCP 関係の情報については、479ページの「UUCP データベースの編集」を、passwd ファイルについては、479ページの「/etc/passwd ファイルの修正」を参照してください。

仮想ネットワークの場合の asppp.cf 構成ファイル

仮想ネットワーク上のローカルマシン用の構成ファイルには、そのネットワーク内にあってローカルホストからアクセスできるすべてのリモートホストに関する情報が含まれている必要があります。さらに、仮想ネットワーク上のマシンは、どれもダイヤルインとダイヤルアウトの両方の機能を備えたものとして構成されていなければなりません。ローカルホストマシンがブートされると、リンクマネージャは asppp.cf ファイルを読んで通信を確立します。

例 24–5 は、仮想ネットワーク 192.41.47 の nomada 用として設定した構成ファイルです。

例 24–5 nomada 用の構成ファイル

```
# /etc/asppp.cf for hosta

ifconfig ipd0 plumb nomada netmask + up
defaults
   interface ipd0
path
   peer_ip_address  nomadb
   peer_system_name lawrence     # name machine logs in with
path
   peer_ip_address nomadc
   peer_system_name azziz
```

例 24–6 は、仮想ネットワーク 192.41.47 の nomadb 用として設定した構成ファイルです。

例 **24–6**　nomadb 用の構成ファイル

```
# /etc/asppp.cf for nomadb

ifconfig ipd0 plumb nomadb netmask + up
defaults
   interface ipd0
path
   peer_ip_address   nomada
   peer_system_name  tamerlane  # name the machine logs in with
path
   peer_ip_address   nomadc
   peer_system_name  azziz
```

PAP と CHAP のキーワードに関する規則

- サーバーまたはクライアントのどちらも、認証を要求することも認証を行うこともできる。
- PAP と CHAP の両方が存在する場合は、認証システムはまず CHAP を試みる。失敗するとリンクは終了する。認証システムは PAP を試みない。
- PAP と CHAP の認証キーワードのデフォルトはオフ。キーワードの構文は次のとおりです。

```
require_authentication     off | pap[chap] | chap[pap]

will_do_authentication     off | pap[chap] | chap[pap]
```

- pap_id と pap_password キーワードまたは pap_peer_id と pap_peer_password キーワードに対する値を、関連の path に指定しなかった場合は、それぞれの値は NULL 文字列に設定されます。
- 該当する path について、chap_name、 chap_secret、 chap_peer_secret、chap_peer_name キーワードと値を指定する必要があります。
- **peername** – ポイントツーポイントリンクの認証システムの対にあるシステムの名称です。次に指定する構文を持つ文字列の形をとります。
- **string** – 空白が埋め込まれていない単一のトークンです。特殊な文字を埋め込むためには標準の ANSI C \ エスケープシーケンスを使用することができます。空白文字には \s 使用します。文字列の始めにあるポンド記号はエスケープして

(\#)、コメントとして解釈されないようにする必要があります。NULL (\0) はこの文字列を切り捨てます。

表 24–1 PAP と CHAP のキーワードの定義

キーワード	値の定義
`require_authentication` *keywords*	対等システムがそれ自身を認証することを指定する。`pap` か `chap` のどちらかがある場合は、対等システムは認証に参加するか、または接続を終了する必要がある。デフォルト値は off
`pap_peer_id` *peername*	現在のパスについて認証される必要のある対等システムの名前を指定する。*peername* 文字列の長さは 1 オクテット以上。長さがゼロの文字列を指示するには、このキーワードを省略する
`pap_peer_password` *string*	対等システムのパスワードを 1 オクテット以上の長さで指定する。長さがゼロの文字列を指示するには、このキーワードを省略する
`chap_peer_secret` *string*	対等システムが送る応答を生成するためにチャレンジ値とともに使用されるシークレットを指定する。形式は 1 オクテット以上の長さで、少なくとも 16 オクテット以上が望ましい
`chap_peer_name` *peername*	パケットを伝送する対等システムの識別情報を指定する。名前には、NULL と、CR/LF で終わる文字列は使用できない。名前は、対等システムからの応答パケットの一部として受信されるもので、1 オクテット以上の長さからなる
`will_do_authentication` *keywords*	システムが、指定した認証プロセスに認証された対等システムとして参加する意志があるかどうかを指定する。`pap` と `chap` の両方が存在する場合は、システムはどちらの認証プロトコルにも参加する意志を持つことになる。デフォルト値は off
`pap_id` *peername*	応答パケットに入れて認証システムに送るシステムの名前を指定する。長さがゼロの文字列を指示するには、このキーワードを省略する
`pap_password` *string*	応答パケットに入れて認証システムに送るシステムのパスワードを指定する。長さがゼロの文字列を指示するには、このキーワードを省略する
`chap_secret` *string*	認証システムに送る応答を生成するために、受信したチャレンジ値とともに使用するシークレットを入れる。形式は 1 オクテット以上の長さで、少なくとも 16 オクテット以上が望ましい
`chap_name` *peername*	システムの識別情報を指定する。名前は、NULL または CR/LF で終わるものであってはならない。この名前は、応答パケットに入れて認証システムに送られる

構成キーワード

この節では、asppp.cf 構成ファイルで使用できる構成キーワードと、それぞれについて定義する必要のある値について説明します。これらのキーワードのほとんどは必須ではありません。必須のものについてはその旨を示しています。キーワードについての詳しい説明は、RFC 1331、1332、1333、および 1334 を参照してください。

表 24–2 は、すべての asppp.cf ファイルに含まれていなければならない必須キーワードの一覧です。

表 24–2 asppp.cf の必須キーワード

キーワード	値の定義
ifconfig *parameters*	*parameters* に指定する値で ifconfig コマンドを実行するようリンクマネージャに指示する。詳細は、500ページの「asppp.cf ファイルの ifconfig セクション」、503ページの「マルチポイントダイヤルインサーバーの ifconfig セクション」、および ifconfig(1M) のマニュアルページを参照
path	この (現行の) パスの属性としてグループ化するトークンシーケンスの始まりを指定する。現行パスを形成する属性の集合は、後続の path キーワード、defaults キーワード、ファイルの終わり文字のどれかが生じた時点で終了する
interface (ipdptp*n*、 ipdptp* または ipd*n*)	ネットワーク内の各インタフェースについて、ipdptp (静的ポイントツーポイント)、ipdptp* (動的ポイントツーポイント)、ipd (マルチポイント) のどれかのデバイスを指定する。ipdptp*n* と ipd*n* の場合は、このキーワードは、*n* で定義される特定のインタフェースを現行パスに関連付ける。*n* は 0 または正の整数でなければならない。この数は、path セクションに定義されているインタフェースと、ifconfig セクションに指定されているインタフェースが一致するようにする ipdptp** インタフェースの場合は、* は、インタフェースが、down として構成されているどのポイントツーポイントインタフェースにも一致することを示す

表 24–2 asppp.cf の必須キーワード 続く

キーワード	値の定義
peer_system_name *hostname*、 peer_system_name *username*	ダイヤルアウトマシンでは、ローカルマシンから呼び出したいリモートエンドポイントのホスト名 (*hostname*) を指定する。この名前は、/etc/uucp/Systems ファイルの中のシステム名と同じである。リモートシステム名を現行パスに関連付ける。この名前は、/etc/uucp/Systems ファイルから、アウトバウンド接続に関する、モデムと対等システムに固有の情報を見つけるために使用される ダイヤルインマシンでは、そのダイヤルインマシンにログインするときにリモートマシンが使用するユーザー名 (*username*) を指定する。*username* と、接続の獲得に使用されたログイン名との突き合わせによって、適正なパスが決定される
peer_ip_address *hostname*、 peer_ip_address *ip-address*	宛先ホストアドレスを指定する。これは、マルチポイントリンクの場合に限り必要とされる。このアドレスは現行パスに関連付けされる。パスがポイントツーポイントインタフェースを示している場合は、この値は無視される。アドレスの形式は、ドット付き 10 進数、16 進数、シンボルのどれでもよい

表 24–3 に、PPP 構成をさらに進んで定義するために使用できる、asppp.cf の省略可能キーワードを示します。

表 24–3 asppp.cf の省略可能キーワード

キーワード	値の定義
debug_level *0–9*	ログファイルに書き込むデバッグ情報の量を定義する 0 ～ 9 の整数。数値が大きいほど出力の量が多くなる
defaults	次の path キーワードか、EOF 文字が現れるまでの後続のすべてのトークンシーケンスをデフォルトの属性に設定して、その間に定義されるパスに適用することを指示する
default_route	現行パスに対応する IP 層が完全に稼動状態にあるときに、このパスの対等 IP アドレスをデフォルトの宛先としてルーティングテーブルに追加するよう、リンクマネージャに指示する。IP 層がシャットダウンされると、このルートは削除される
inactivity_timeout *seconds*	現行パスの接続が、終了しないでアイドル状態のままでいられる最大秒数を指定する。タイムアウトなしの場合は 0 を指定する。デフォルトは 120 秒

表 24–3 asppp.cf の省略可能キーワード 続く

キーワード	値の定義
ipcp_async_map *hex-number*	現行パスの非同期制御文字マップを指定する。*hex-number* は、マップを形成する 4 オクテットの自然 (ビッグエンディアン) 形式を示す。デフォルト値は 0x FFFFFFFF
ipcp_compression (vj または off)	IP 圧縮を使用可能にするかどうかを指定する。デフォルトは、Van Jacobson 圧縮アルゴリズム (vj)
lcp_compression (on または off)	PPP アドレスフィールド、制御フィールド、プロトコルフィールドの圧縮を使用可能にするかどうかを指定する。デフォルトは on
lcp_mru *number*	必要な最大受信ユニットパケットサイズの値を指定する。number はサイズを指定するオクテット数。デフォルトは 1500
negotiate_address (on または off)	ローカル IP アドレス割り当てをネゴシエーションにより入手し動的に割り当てるかどうかを指示する。これを使用可能にした場合は、ローカルアドレスは PPP リンクのリモート側から渡される。このようにして渡された場合、0.0.0.0 を除くどのようなローカルアドレスでも、インタフェースの初期構成に使用できる。デフォルトはネゴシエーションなし (off)
peer_ip_address *hostname*、peer_ip_address *ip-address*	宛先ホストアドレスを指定する。このキーワードはポイントツーポイントリンクの場合に限りオプション。*address* は現行パスに関連付けされる。アドレスの形式は、ドット付き 10 進数、16 進数、シンボルのどれでもよい
version *n*	構成ファイルの内容が形式バージョン *n* に対応することを指定する。このキーワードを使用する場合は、ファイルの最初のキーワードとする必要がある。このキーワードがないときは、バージョンは 1 とみなされる。このマニュアルでは、バージョン 1 形式の定義を構成ファイルに使用している

第 25 章

UUCP の概要

この章では UNIX 間コピープログラム (UUCP) およびデーモンについて説明します。次のトピックについて説明します。

UUCP は、コンピュータが相互にファイルの転送、メールの交換を実現します。また、UUCP を使用して Usenet のような大規模ネットワークにコンピュータを接続することも可能です。

Solaris 環境には、HoneyDanBer UUCP とも呼ばれる基本ネットワークユーティリティ (BNU) バージョンの UUCP が備えられています。UUCP という用語はシステムを形成するすべてのファイルとユーティリティを意味するものであり、uucp プログラムはそのシステムの一部にすぎません。UUCP のユーティリティには、コンピュータ間でファイルをコピーするためのもの (uucp と uuto) から、リモートログインやリモートコマンド実行のためのもの (cu と uux) まで、さまざまなものがあります。

UUCP のハードウェア構成

UUCP は次のハードウェア構成をサポートしています。

直接リンク	2 つのマシンのシリアルポート間を RS-232 ケーブルで結ぶことにより、他のコンピュータとの間の直接リンクを作成できる。2 つのコンピュータが常時互いに通信を行い、両者の間の距離が 15m 以内の場合は、直接リンクを使用すると便利。この制限距離は、短距離モデムを使用することである程度延長できる
電話回線	高速モデムなどの自動呼び出し装置 (ACU) を使用すれば、通常の電話回線を介して他のコンピュータと通信できる。モデムは、UUCP が要求する電話番号をダイヤルする。受信側のモデムは、着信に応答できなければならない
ネットワーク	UUCP は、TCP/IP またはその他のプロトコルファミリが機能するネットワークを介しても通信できる。コンピュータがネットワーク上でホストとして確立されていれば、そのネットワークに接続されている他のどのホストとも通信できる

この章では、UUCP ハードウェアをすでに設置、構成してあるものとして説明を進めます。モデムを設定する必要がある場合は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』と、モデムに付属のマニュアルを参照してください。

UUCP ソフトウェア

Solaris インストールプログラムを実行するときに全体ディストリビューションを選択していれば、UUCP ソフトウェアは自動的に組み込まれています。あるいは、`pkgadd` を使用して UUCP を単独で追加することもできます。UUCP のプログラムは、デーモン、管理プログラム、ユーザープログラムの 3 種類に分類されます。

UUCP デーモン

UUCP システムのデーモンには、`uucico`、`uuxqt`、`uusched`、`in.uucpd` の 4 つがあります。これらのデーモンは、UUCP のファイル転送とコマンド実行を取り扱います。必要な場合は、これらのデーモンをシェルから手動で実行することもできます。

uucico	リンクに使用するデバイスを選択し、リモートコンピュータへのリンクを確立し、必要なログインシーケンスとアクセス権の検査を行い、データを転送し、ファイルを実行し、結果をログに記録し、転送の完了を mail によりユーザーに通知する。uucico は、UUCP ログインアカウント用の「ログインシェル」として働く。ローカル uucico デーモンはリモートマシンを呼び出して、セッションの間、リモート uucico デーモンと直接通信する 必要なファイルすべてを作成したら、uucp、uuto、および uux プログラムが uucico デーモンを実行してリモートコンピュータに接触する。uusched と Uutry は、どちらも uucico を実行する (詳細は、uucico(1M) のマニュアルページを参照)
uuxqt	リモート実行要求を処理する。uuxqt は、スプールディレクトリを検索して、リモートコンピュータから送られた実行ファイル (名前は常に X.*file*) を見つける。X.*file* が見つかると、uuxqt はそのファイルをオープンして、実行に必要なデータファイルのリストを取得する。次に、必要なデータファイルが使用可能でアクセスできるかどうかを確認する。ファイルが使用可能であれば、uuxqt は Permissions ファイルを調べて、要求されたコマンドを実行する権限があるかどうかを確認する。uuxqt デーモンは、cron により起動される uudemon.hour シェルスクリプトから実行される (詳細は、uuxqt(1M) のマニュアルページを参照)
uusched	スプールディレクトリ内でキューに入っている作業をスケジュールする。uusched は、cron から起動される uudemon.hour シェルスクリプトによって、ブート時に最初に実行される (詳細は、uusched(1M) のマニュアルページを参照)。uusched は uucico デーモンを起動する前に、リモートコンピュータを呼び出す順序をランダム化する
in.uucpd	ネットワークを介した UUCP 接続をサポートする。リモートホスト上の inetd は、UUCP 接続が確立されるたびに in.uucpd を呼び出す。次に、uucpd がログイン名を要求する。呼び出し側ホストの uucico は、これに対してログイン名を応答しなければならない。次に in.uucpd は、不要な場合を除いてパスワードを要求する (詳細は、in.uucpd(1M) のマニュアルページを参照)

UUCP 管理プログラム

ほとんどの UUCP 管理プログラムは、/usr/lib/uucp にあります。基本データベースファイルの多くは、/etc/uucp に入っています。ただし、uulog だけは例外で、これは /usr/bin にあります。uucp ログイン ID のホームディレクトリは /usr/lib/uucp です。su または login を使用して管理プログラムを実行するときには、uucp ユーザー ID を使用してください。このユーザー ID は、プログラムとスプールデータファイルを所有しています。

uulog	指定したコンピュータのログファイルの内容を表示する。ログファイルは、このマシンが通信する各リモートコンピュータごとに作成される。ログファイルには、uucp、uuto、uux の使用が記録される (詳細は、uucp(1C) のマニュアルページを参照)
uucleanup	スプールディレクトリをクリーンアップする。これは通常、cron によって起動される uudemon.cleanup シェルスクリプトから実行される (詳細は、uucleanup(1M) のマニュアルページを参照)
Uutry	呼び出し処理機能をテストし、簡単なデバッグを行うことができる。uucico デーモンを呼び出して、このマシンと指定されたリモートコンピュータとの間の通信リンクを確立する (詳細は、Uutry(1M) のマニュアルページを参照)
uucheck	UUCP のディレクトリ、プログラム、およびサポートファイルの有無を検査する。また、/etc/uucp/Permissions ファイルの所定の部分に、明らかな構文エラーがないかどうかも検査する (詳細は、uucheck(1M) のマニュアルページを参照)

UUCP ユーザープログラム

UUCP のユーザープログラムは /usr/bin にあります。これらのプログラムを使用するのに、特別な権限は必要ありません。

cu	このマシンをリモートコンピュータに接続して、ユーザーが両方のマシンに同時にログインできるようにする。cu を使用すれば、接続したリンクを切断することなく、どちらのマシンでもファイルを転送したり、コマンドを実行したりできる (詳細は、cu(1C) のマニュアルページを参照)
uucp	あるマシンから別のマシンへファイルをコピーする。uucp は作業ファイルとデータファイルを作成し、転送するジョブをキューに入れ、uucico デーモンを呼び出す。このデーモンは、リモートコンピュータへの接続を試みる (詳細は、uucp(1C) のマニュアルページを参照)
uuto	ローカルマシンから、リモートマシン上の公共スプールディレクトリ /var/spool/uucppublic/receive にファイルをコピーする。uucp はリモートマシン上のアクセス可能な任意のディレクトリにファイルをコピーするのに対して、uuto は所定のスプールディレクトリにファイルを格納し、リモートユーザーに uupick を使用してそのファイルを取り出すよう指示する (詳細は、uuto(1C) のマニュアルページを参照)
uupick	uuto を使用してコンピュータにファイルが転送されてきたときに、/var/spool/uucppublic/receive からファイルを取得する (詳細は、uuto(1C) のマニュアルページを参照)

uux	リモートマシン上でコマンドを実行するために必要な作業ファイル、データファイル、および実行ファイルを作成する (詳細は、uux(1C) のマニュアルページを参照)
uustat	要求された転送 (uucp、uuto、uux) の状態を表示する。また、キューに入っている転送を制御する手段も提供する (詳細は、uustat(1C) のマニュアルページを参照)

UUCP データベースファイル

UUCP の構成の主要部分の 1 つに、UUCP データベースを形成するファイルの構成があります。これらのファイルは /etc/uucp ディレクトリにあります。マシン上で UUCP または PPP を設定するには、これらのファイルを編集する必要があります。UUCP データベースファイルには次の内容があります。

Config	変数パラメータのリストが入っている。これらのパラメータは、ネットワークを構成するために手動で設定できる
Devconfig	ネットワーク通信を構成するために使用される
Devices	ネットワーク通信を構成するために使用される
Dialcodes	Systems ファイルのエントリの電話番号フィールド内で使用できるダイヤルコード省略名が入っている。これは必須ではないが、UUCP の他に PPP でも使用できる
Dialers	リモートコンピュータとの接続を確立するときに、モデムとのネゴシエーションを行うために必要な文字列が入っている。これは、UUCP の他に PPP でも使用される
Grades	ジョブのグレードと、ジョブの各グレードに対応するアクセス権を定義する。これらは、リモートコンピュータに対するジョブをキューに入れる際に、ユーザーが指定できる
Limits	このマシンで同時に実行できる uucico、uuxqt、および uusched の最大数を定義する
Permissions	このマシンにファイルを転送したり、コマンドを実行しようとしているリモートホストに与えられるアクセス権のレベルを定義する

Poll	このシステムがポーリングするマシンと、ポーリングする時刻を定義する
Sysfiles	uucico と cu が、Systems、Devices、および Dialers ファイルとして、別のファイルや複数のファイルを使用する時に、その割り当てを行う
Sysname	TCP/IP ホスト名の他に、各マシンに固有の UUCP 名を定義できる
Systems	uucico デーモン、cu、および PPP が、リモートコンピュータへのリンクを確立するために必要とする情報が入っている。この情報には、リモートホスト名、リモートホストに対応する接続デバイス名、そのホストに接続できる日時、電話番号、ログイン ID、パスワードが含まれる

サポートデータベースの一部とみなすことのできるファイルが他にもいくつかありますが、これらは、リンクの確立とファイルの転送に直接には関係しません。

UUCP データベースファイルの構成設定

UUCP データベースは、531ページの「UUCP データベースファイル」に示したファイルから構成されます。ただし、基本的な UUCP 構成に関係する重要なファイルは次に示すものだけです。

- /etc/uucp/Systems
- /etc/uucp/Devices
- /etc/uucp/Dialers

PPP は UUCP データベースの一部を使用するので、PPP を構成する予定がある場合は、少なくともこれらのデータベースファイルだけは理解しておく必要があります。これらのデータベースを構成してしまえば、その後の UUCP の管理はきわめて簡単です。通常、Systems ファイルを最初に編集し、次に Devices ファイルを編集します。/etc/uucp/Dialers ファイルは、普通はデフォルトのままで使用できますが、デフォルトファイルに含まれていないダイヤラを追加する予定がある場合は編集が必要になります。基本的な UUCP 構成と PPP 構成には、さらに次のファイルを加えることもできます。

- /etc/uucp/Sysfiles
- /etc/uucp/Dialcodes
- /etc/uucp/Sysname

これらのファイルは互いに関係しながら機能するので、1 つでも変更する場合は、全部のファイルの内容を理解しておくことが必要です。あるファイルのエントリに変更を加えた場合に、別のファイル内の関連エントリに対しても変更が必要になることがあります。531ページの「UUCP データベースファイル」に挙げたその他のファイルは、上記のファイルほど緊密な相互関係を持っていません。

注 - PPP が使用するファイルはこの節で説明するものだけであり、他の UUCP データベースファイルは使用しません。

第 26 章

UUCP の管理

この章では、ご使用のマシンに合わせてデータベースファイルを変更したあと、UUCP 操作を起動する方法について説明します。この章には、Solaris 環境が動作するマシンで UUCP を構成し保守するための、手順と障害の解明についての情報が記載されています。

UUCP 管理の作業マップ

表 26–1 で、この章で説明する手順の記載場所と各手順について簡単に説明します。

表 26–1 作業マップ: UUCP 管理

作業	説明	参照個所
リモートマシンにユーザーシステムへのアクセスを許可する	/etc/passwd ファイルを編集し、ユーザーのシステムへのアクセスを許可するマシンを識別するようエントリを追加する	536ページの「UUCP ログインの追加方法」
UUCP を起動する	UUCP の起動用に提供されたシェルスクリプトを使用する	538ページの「UUCP の起動方法」
UUCP を TCP/IP ネットワーク上で有効にする	/etc/inetd.conf ファイルと /etc/uucp/Systems ファイルを編集し、TCP/IP 用の UUCP を起動する	540ページの「TCP/IP 用 UUCP の起動方法」
UUCP に起こりがちな問題を解決する	モデムまたは ACU の異常を確認するための診断ステップ。 送信に関するデバッグを行うための診断ステップ	543ページの「モデムまたは ACU の障害確認方法」 543ページの「送信に関するデバッグ方法」

UUCP のログインの追加

リモートマシンからの UUCP (uucico) 着信要求が正しく取り扱われるように、各リモートマシンはローカルシステム上にログインを持っていなければなりません。

▼ UUCP ログインの追加方法

ユーザーのシステムへのアクセスをリモートマシンに許可するには、次の手順を行なって /etc/passwd ファイルにエントリを追加する必要があります。

1. **/etc/passwd ファイルを編集し、システムにアクセスを許可するマシンを識別するためのエントリを追加します。**

   UUCP 接続でのシステムへのアクセスを許可するリモートマシンについて、一般的に、次のように、エントリを /etc/passwd ファイルに入力します。

```
Ugobi:*:5:5:gobi:/var/spool/uucppublic:/usr/lib/uucp/uucico
```

リモートマシンのログイン名は慣例的に、そのマシン名の前に大文字の U を付けたものです。8 文字を超える名前は使用できないので、一部を短縮した名前や省略名を使用しなければならない場合もあります。

例に示したエントリは、Ugobi からのログイン要求に /usr/lib/uucp/uucico が応答することを示しています。ホームディレクトリは /var/spool/uucppublic です。パスワードは /etc/shadow ファイルから取得されます。パスワードとログイン名は、リモートマシンの UUCP 管理者と協議して決める必要があります。リモート側の管理者は、ログイン名と暗号化されていないパスワードを含む正しいエントリを、リモートマシンの Systems ファイルに追加する必要があります。

2. 他のシステムの **UUCP** 管理者と、ローカルマシン名を調整します。

   同様に、ローカルマシン名とパスワードについて、UUCP を介して通信する相手方のすべてのマシンの UUCP 管理者と協議する必要があります。

UUCP の起動

UUCP には、次に示す 4 つのシェルスクリプトが付属しています。これらのスクリプトは、リモートマシンをポーリングし、転送を再スケジュールし、古いログファイルと成功しなかった転送を処理します。

- uudemon.poll
- uudemon.hour
- uudemon.admin
- uudemon.cleanup

UUCP を円滑に運用するには、これらのスクリプトを定期的に実行する必要があります。Solaris の全体インストールを行なった場合は、これらのスクリプトを実行するための crontab ファイルが、インストールプロセスの一環として自動的に /usr/lib/uucp/uudemon.crontab の中に作成されます。全体インストールでない場合は、UUCP パッケージをインストールするときにこのファイルが作成されます。

UUCP シェルスクリプトは手動でも実行できます。次に示すのは、uudemon.crontab のプロトタイプです。このファイルは、マシンの運用の都合に合わせて適宜変更することができます。

```
#
#ident  "@(#)uudemon.crontab    1.5     97/12/09 SMI"
#
# This crontab is provided as a sample. For systems
# running UUCP edit the time schedule to suit, uncomment
# the following lines, and use crontab(1) to activate the
# new schedule.
#
#48 8,12,16 * * * /usr/lib/uucp/uudemon.admin
#20 3 * * * /usr/lib/uucp/uudemon.cleanup
#0 * * * * /usr/lib/uucp/uudemon.poll
#11,41 * * * * /usr/lib/uucp/uudemon.hour
```

注 - デフォルトでは、UUCP の操作は無効にされています。UUCP を有効にするには、タイムスケジュールを編集し、uudemon.crontab ファイルの適切な行のコメントを解除してください。

▼ UUCP の起動方法

uudemon.crontab ファイルは、次の手順に従って起動します。

1. スーパーユーザーになります。
2. /usr/lib/uucp/uudemon.crontab ファイルを編集し、必要に応じてエントリを変更します。
3. 次のように入力します。

```
crontab < /usr/lib/uucp/uudemon.crontab
```

uudemon.poll シェルスクリプト

デフォルトの uudemon.poll シェルスクリプトは、1 時間に 1 回 /etc/uucp/Poll ファイルを読み取ります。Poll ファイル内のマシンのどれかに

対するポーリングがスケジュールされると、作業ファイル (C.*sysnxxxx*) が /var/spool/uucp/*nodename* ディレクトリに入れられます。*nodename* は、そのマシンの UUCP ノード名です。

このシェルスクリプトは、1 時間に 1 回ずつ uudemon.hour の前に実行されるようにスケジュールされているので、uudemon.hour が呼び出されたときには、作業ファイルが存在しています。

uudemon.hour シェルスクリプト

デフォルトの uudemon.hour シェルスクリプトは次のことを行います。

- uusched プログラムを呼び出し、スプールディレクトリを検索して未処理の作業ファイル (C.) を見つけ、それらの作業ファイルをリモートマシンに転送するためにスケジュールする
- uuxqt デーモンを呼び出し、スプールディレクトリを検索して、ローカルコンピュータに転送済みで、転送時に処理されなかった実行ファイル (X.) を見つける

デフォルトでは、uudemon.hour は 1 時間に 2 回実行されます。リモートマシンへの呼び出しの失敗の頻度が高いと予測される場合は、このスクリプトの実行頻度を増やすこともできます。

uudemon.admin シェルスクリプト

デフォルトの uudemon.admin シェルスクリプトは次のことを行います。

- p オプションと q オプション付きで uustat コマンドを実行する。q は、キューに入っている作業ファイル (C.)、データファイル (D.)、実行ファイル (X.) の状態を報告する。p は、ロックファイル (/var/spool/locks) 中に列挙されているネットワークプロセス用のプロセス情報を表示する
- 結果の状態情報を、mail により uucp 管理ログインに送る

uudemon.cleanup シェルスクリプト

デフォルトの uudemon.cleanup シェルスクリプトは次のことを行います。

- /var/uucp/.Log ディレクトリから個々のマシンに関するログファイルを取り出し、それらをマージし、他の古いログ情報とともに /var/uucp/.Old ディレクトリに入れる

- 7 日以上経過している作業ファイル (C.)、7 日以上経過しているデータファイル (D.)、2 日以上経過している実行ファイル (X.) を、スプールファイルから削除する
- 配達できなかったメールを送信元に戻す
- その日に収集した状態情報の要約を、メールにより UUCP 管理ログイン (uucp) に送る

TCP/IP を介した UUCP の実行

TCP/IP ネットワーク上での UUCP を実行するには、この節で説明するようにいくつかの変更が必要になります。

▼ TCP/IP 用 UUCP の起動方法

1. /etc/inetd.conf ファイルを編集し、次のエントリがコメントマーク (#) で始まっていないことを確認します。

```
uucp stream tcp nowait root /usr/sbin/in.uucpd in.uucpd
```

2. /etc/uucp/Systems ファイルを編集し、対象のエントリが次のフィールドを持っていることを確認します。

   System-Name Time TCP Port networkname Standard-Login-Chat

   典型的なエントリは次のようになります。

```
rochester Any TCP - ur-seneca login: Umachine password: xxx
```

   networkname フィールドには、TCP/IP ホスト名を明示的に指定できます。これは、一部のサイトにとっては重要な点です。上の例に示したサイトの UUCP ノード名 rochester は、TCP/IP ホスト名 ur-seneca と違っています。rochester という TCP/IP ホスト名を持ち、UUCP を実行する別のマシンがあっても問題はありません。

   Systems ファイル内の Port フィールドには - を指定します。これは、uucp と指定するのと同じです。ほとんどの場合、*networkname* はシステム名と同じで、Port フィールドは - となります。これは、services データベースから標準 uucp ポートを使用することを意味します。in.uucpd デーモンは、認証のため

にリモートマシンがログインとパスワードを送ることを想定しているので、getty や login と同様に、ログインとパスワードを要求します。

3. /etc/inet/services ファイルを編集し、次のように **UUCP** 用のポートを設定します。

```
uucp 540/tcp uucpd # uucp daemon
```

このエントリを変更する必要はありません。ただし、マシンがネームサービスとして NIS または NIS+ を実行する場合は、/etc/services の /etc/nsswitch.conf エントリを変更して、まず files、次に nis または nisplus が検査されるようにする必要があります。

UUCP のセキュリティと保守

UUCP の設定が終われば、その後の保守は簡単です。この節では、セキュリティ、保守、および障害追跡に関連する UUCP の作業について説明します。

UUCP のセキュリティの設定

デフォルトの /etc/uucp/Permissions ファイルは、UUCP リンクに関する最大限のセキュリティを提供します。デフォルトの Permissions ファイルには、エントリは入っていません。

定義する各リモートマシンについて、次に示す追加パラメータを設定できます。

- ローカルマシンからファイルを受け取る方法
- 読み取り権と書き込み権が与えられるディレクトリ
- リモート実行に使用できるコマンド

典型的な Permissions のエントリは次のようになります。

```
MACHINE=datsun LOGNAME=Udatsun VALIDATE=datsun
COMMANDS=rmail REQUEST=yes SENDFILES=yes
```

このエントリでは、(システム内のどこかからではなく、通常の UUCP ディレクトリとの間での) ファイルの送信と受信が可能となり、ログイン時に UUCP ユーザー名の認証が行われます。

日常の UUCP の保守

UUCP の保守に必要な作業の量はさほど多くはありません。538ページの「UUCP の起動方法」で述べたように、crontab ファイルを正しい場所に配置してあることを確認する以外に注意する必要があるのは、メールファイルと公共ディレクトリが大きくなるという点だけです。

UUCP に関連する電子メール

UUCP のプログラムとスクリプトが生成する電子メールメッセージは、すべてユーザー ID uucp に送られます。管理者がユーザー uucp として頻繁にログインしていないと、メールが蓄積されている (このためディスク空間を浪費している) ことに気付かない場合があります。この問題を解決するには、/etc/mail/aliases の中に別名を 1 つ作り、root か自分自身、そして他の UUCP 保守責任者に、電子メールをリダイレクトします。aliases ファイルを変更したあとで、newaliases コマンドを実行するのを忘れないようにしてください。

UUCP 公共ディレクトリ

ディレクトリ /var/spool/uucppublic は、UUCP がデフォルトでファイルをコピーできる場所として、すべてのシステムに対して提供されているディレクトリです。すべてのユーザーが、/var/spool/uucppublic への移動、その中のファイルの読み書きを行う権限を持っています。しかし、スティッキビットが設定されているため、このディレクトリのモードは 01777 です。したがって、ユーザーには、このディレクトリにコピーされ uucp に所有されているファイルを削除することはできません。このディレクトリからファイルを削除できるのは、root または uucp としてログインした UUCP 管理者だけです。このディレクトリ内に無秩序にファイルが蓄積するのを防ぐために、定期的に整理する必要があります。

このような定期的な整理がユーザーにとって面倒な場合は、スティッキビットを削除するよりも、各ユーザーに uuto と uupick を使用するよう奨励してください。スティッキビットはセキュリティのために設定されています (uuto と uupick の使い方については、uuto(1C) のマニュアルページを参照してください)。このディレクトリのモードの制限の度合を強めて、たとえば特定のユーザーグループだけに使用を

限定することもできます。だれかがディスク空間を使い切ってしまうことが望ましくないのであれば、そのディスクへの UUCP アクセスを拒否することもできます。

UUCP の障害追跡

ここでは、UUCP に関する一般的な問題を解決するための手順について説明します。

▼ モデムまたは ACU の障害確認方法

モデムや ACU で、適正に動作していないものがないかどうかを、いくつかの方法で検査できます。

1. 次のコマンドを実行し、接続障害の回数と理由を表示します。

```
uustat -q
```

2. 特定の回線を介した呼び出しを行い、その試行に関するデバッグ情報を表示します。

   この回線は、/etc/uucp/Devices ファイルの中で direct として定義されていなければなりません (回線が自動ダイヤラに接続している場合は、コマンド行の終わりに電話番号を追加するか、デバイスを direct として設定する必要があります)。次のように入力します。

```
cu -d -l line
```

   line は /dev.cua/a です。

▼ 送信に関するデバッグ方法

特定のマシンに接続できない場合は、Uutry と uucp を使用して、そのマシンに対する通信を検査できます。

1. 次のように入力し、接続を調べます。

```
/usr/lib/uucp/Uutry -r machine
```

machine には、接続に問題のあるマシンのホスト名を指定します。このコマンドは次のことを行います。

a. デバッグ機能を指定して転送デーモン (uucico) を起動する。root としてログインしていれば、さらに多くのデバッグ情報が得られる

b. デバッグ出力を /tmp/***machine*** に送る

c. 次のように入力すると、デバッグ出力を端末に表示する

```
tail –f
```

出力を終了するには Control-c キーを押します。この出力を保存したい場合は、/tmp/*machine* から出力内容をコピーします。

2. Uutry を使用しても問題の原因が分からない場合は、次のように入力して、ジョブをキューに入れてみます。

```
uucp -r file machine\!/dir/file
```

file には転送したいファイル、*machine* には転送先のマシンを指定します。*/dir/file* には、相手のマシンのどこにファイルを転送するかを指定します。r オプションを指定すると、ジョブはキューに入りますが、転送は開始されません。

3. 次のように入力します。

```
Uutry
```

それでも問題が解決できないときは、ご購入先へお問い合わせください。デバッグ出力を保存しておいてください。これは問題の診断に役立ちます。

Uutry で –x *n* オプションを使用して、デバッグのレベルを増減することも考えてみてください。*n* はデバッグレベルを指定します。Uutry のデフォルトのデバッグレベルは 5 です。

デバッグレベル 3 では、接続がいつどのように確立されたかについての基本的な情報は提供されますが、転送自体について提供される情報は多くはありません。これに対して、デバッグレベル 9 では、転送処理に関するすべての情報が網羅されます。デバッグは転送の両端で行われるという点に注意してください。比較的大きいテキストについて 5 より高いレベルのデバッグを行いたい場合は、相手サイトの管理者に連絡して、デバッグを行う時期について同意を得てください。

UUCP /etc/uucp/Systems ファイルの検査

特定のマシンと接続しようとすると障害が発生する場合は、Systems ファイルの中の情報が最新のものであることを確認してください。マシンに関する次の情報が、最新でない可能性があります。

- 電話番号
- ログイン
- パスワード

UUCP エラーメッセージの検査

UUCP のエラーメッセージには、ASSERT と STATUS の 2 つの種類があります。

- プロセスが異常終了した場合は、ASSERT メッセージが /var/uucp/.Admin/errors に記録されます。この種類のメッセージには、ファイル名、sccsid、回線番号、およびテキストが含まれています。この種類のメッセージが出るのは、通常、システムに問題がある場合です。
- STATUS エラーメッセージは /var/uucp/.Status ディレクトリに格納されます。このディレクトリ内には、ローカルコンピュータが通信しようとした各リモートマシンについて、それぞれファイルが作られます。これらのファイルには、試行した通信と、その通信が成功したかどうかについての状態情報が入っています。

基本情報の検査

次のコマンドを使用して、基本的なネットワーク情報を検査するために使用できます。

- uuname コマンドは、ローカルマシンが接続できるマシンのリストを表示したい場合に使用します。

- uulog コマンドは、特定のホストのログディレクトリの内容を表示するために使用します。
- uucheck –v コマンドは、uucp が必要とするファイルとディレクトリが存在しているかどうかを検査するために使用します。また、Permissions ファイルも検査して、設定してあるアクセス権に関する情報を出力します。

第 27 章

UUCP リファレンス

この章では、UUCP を使用する場合のリファレンス情報について説明します。次の各項目について説明します。

UUCP /etc/uucp/Systems ファイル

/etc/uucp/Systems ファイルには、uucico がリモートコンピュータとの通信リンクを確立するために必要な情報が入っています。これは、UUCP を構成するときに編集しなければならない最初のファイルです。

Systems ファイルの中の各エントリは、このホストが通信するリモートコンピュータを表します。1 つのホストについて複数のエントリがある場合もあります。付加的なエントリは、順番に試される代替通信パスを表します。さらに、UUCP のデフォルト状態では、/etc/uucp/Systems ファイルに含まれていないコンピュータがこのホストにログインできないようになっています。

Sysfiles ファイルを使用して、Systems ファイルとして使用されるファイルをいくつか定義できます。詳細は、569ページの「UUCP /etc/uucp/Sysfiles ファイル」で Sysfiles ファイルの説明を参照してください。

Systems ファイルのエントリの形式は次のとおりです。

System-Name	*Time*	*Type*	*Speed*	*Phone*	*Chat-Script*

例 27–1 に、Systems ファイルのフィールドの例を示します。

例 27–1 /etc/uucp/Systems のフィールド

```
System-Name Time Type  Speed Phone   Chat-Script
Arabian     Any  ACUEC 38400 111222  Login: Puucp ssword:beledi
```

UUCP System-Name フィールド

このフィールドには、リモートコンピュータのノード名が入ります。TCP/IP ネットワークでは、これは、マシンのホスト名でも、/etc/uucp/Sysname ファイルによって UUCP 通信用として特別に作成した名前でもかまいません。547ページの「UUCP /etc/uucp/Systems ファイル」を参照してください。例 27–1 では、System-Name フィールドにはリモートホスト arabian に関するエントリが含まれています。

UUCP Time フィールド

このフィールドには、リモートコンピュータを呼び出すことのできる曜日と時刻を指定します。Time フィールドの形式は次のとおりです。

```
daytime[;retry]
```

day の部分には、次のエントリのいくつかを含むリストを指定できます。

表 27–1 Day フィールド

Su Mo Tu We Th Fr Sa	個々の曜日
Wk	任意の平日
Any	任意の日
Never	このホストはこのリモートコンピュータの呼び出しをいっさい行わない。呼び出しはリモートコンピュータ側から行う必要がある。それを受けて、このホストは受動モードで稼動する

例 27–1 では、Time フィールドに Any が示されています。これは、ホスト arabian をいつでも呼び出せるということです。

time の部分には、24 時間表記で表した時間の範囲を指定します (たとえば、午前 8 時 00 分から午後 12 時 30 分までなら 0800-1230)。*time* の部分を指定しなかった場合は、どのような時刻にでも呼び出しができるものとみなされます。

0000 の前後にまたがる時間範囲も指定できます。たとえば、0800-0600 は、午前 6 時から午前 8 時までの間を除くすべての時間帯で呼び出し可能であることを示します。

UUCP retry サブフィールド

retry サブフィールドには、試行が失敗してから次の再試行までの間に最小限必要な時間 (分単位) を指定できます。デフォルトの待ち時間は 60 分です。サブフィールド区切り文字はセミコロン (;) です。たとえば、Any;9 は、呼び出しはいつでもできるが、失敗したときは次の再試行までに少なくとも 9 分は待たなければならないことを意味します。

retry エントリを指定しなかった場合は、待ち時間倍加アルゴリズムが使用されます。これは、UUCP がデフォルトの待ち時間から始めて、失敗した試行の回数が増えるほど待ち時間を長くしていくことを意味します。たとえば、最初の再試行待ち時間が 5 分であるとします。応答がない場合は、次の再試行は 10 分後となります。次の再試行は 20 分後というようになり、最大再試行時間の 23 時間に達するまで増加します。*retry* を指定した場合は、常にその値が再試行待ち時間となります。指定がなければ待ち時間倍加アルゴリズムが使用されます。

UUCP Type フィールド

このフィールドには、リモートコンピュータとの通信リンクを確立するために使用するデバイスタイプを指定します。このフィールドで使用するキーワードは、Devices ファイル中のエントリの最初のフィールドと突き合わされます。

例 27–2　Type フィールドと /etc/uucp/Devices ファイル

```
File Name System-Name  Time  Type     Speed  Phone     Chap-Script
Systems   arabian      Any   ACUEC, g 38400  1112222   ogin: Puucp ssword:beledi
```

Type フィールドでは、さらに、システムとの接続に使用するプロトコルを定義できます。上記の例では、デバイスタイプ ACUEC に g プロトコルを組み合わせる方法を示しています (プロトコルの詳細は、562ページの「UUCP Devices ファイル内のプロトコル定義」を参照してください)。

UUCP Speed フィールド

このフィールド (Class フィールドとも呼ばれます) は、通信リンクの確立に使用するデバイスの転送速度を指定します。このフィールドには、ダイヤラのクラスを区別するために、1 個の英字と速度を含めることができます (たとえば、C1200、D1200) (詳細は、558ページの「UUCP Class フィールド」を参照してください)。

デバイスにはどのような速度でも使用できるものがあり、その場合はキーワード Any を使用できます。このフィールドは、Devices ファイルの対応するエントリの Class フィールドに一致していなければなりません。

例 27–3　Speed フィールドと /etc/uucp/Devices ファイル

```
File Name System-Name Time  Type    Speed  Phone   Chap-Script
Systems   eagle       Any   ACU, g  D1200  NY3251  ogin: nuucp ssword: Oakgrass
```

このフィールドに情報を入れる必要がない場合は、フィールドの数を合わせるためにダッシュ (-) を指定してください。

UUCP Phone フィールド

このフィールドには、自動ダイヤラ (ポートセレクタ) に与えるリモートコンピュータの電話番号 (トークン) を指定できます。電話番号は、オプションの英字による省略名と数字部分で構成されます。省略名を使用する場合は、Dialcodes ファイル内に列挙されているものの 1 つでなければなりません。

例 **27–4** Phone フィールドの対応関係

```
File Name System-Name  Time  Type  Speed  Phone     Chap-Script

Systems   nubian       Any   ACU   2400   NY5551212 ogin: Puucp ssword:Passuan
```

この文字列の中に等号 (=) が含まれている場合、二次発信音を待ってから残りの数字をダイヤルするという ACU への指示となります。文字列の中にダッシュ (-) があれば、4 秒間待ってから次の数字をダイヤルするという指示になります。

コンピュータがポートセレクタに接続されている場合は、そのセレクタに接続している他のコンピュータにアクセスできます。この種のリモートマシン用の Systems ファイルエントリの Phone フィールドには、電話番号を入れません。代わりに、このフィールドにはスイッチに渡すトークンを指定します。このようにすれば、このホストがどのリモートマシンとの通信を望んでいるかを、ポートセレクタが判断できます (この場合は、システム名だけを指定するのが普通です)。対応する Devices ファイルエントリでは、エントリの末尾に \D を指定して、このフィールドが Dialcode ファイルを使用して解釈されないようにしなければなりません。

UUCP Chat-Script フィールド

このフィールド (Login フィールドとも呼ばれます) には、チャットスクリプトと呼ばれる文字列が入ります。チャットスクリプトには、ローカルマシンとリモートマシンが対話の最初の時点で互いに受け渡ししなければならない文字が含まれています。チャットスクリプトの形式は次のとおりです。

```
expect  send   [expect  send] ....
```

expect は、対話を開始するために、ローカルホストがリモートホストから送られてくることを想定している文字列です。*send* は、ローカルホストが、リモートホスト

からの *expect* 文字列を受信した後で送信する文字列です。チャットスクリプトには、複数の expect-send シーケンスを含めることもできます。

基本的なチャットスクリプトには次の情報が含まれます。

- ローカルホストがリモートマシンから送られてくることを想定しているログインプロンプト
- ログインするためにローカルホストがリモートマシンに送るログイン名
- ローカルホストがリモートマシンから送られてくることを想定しているパスワードプロンプト
- ローカルホストがリモートマシンに送るパスワード

expect フィールドは、次の形式のサブフィールドを持つことができます。

```
expect[-send-expect]...
```

-send は、その前の *expect* が正常に読み取れなかった場合に送られるものであり、*send* の後の *-expect* は、その次に送られてくると想定されている文字列です。

たとえば、`login--login` という文字列を指定した場合、ローカルホストの UUCP は `login` が送られてくることを想定します。リモートマシンから `login` を受信すると、UUCP は次のフィールドに進みます。`login` を受信しなかった場合は、キャリッジリターンを送信し、再度 `login` が送られてくるのを待ちます。ローカルコンピュータが、初期状態でどのような文字も想定していない場合は、*expect* フィールドで文字列 `""` (NULL 文字列) を指定します。*send* 文字列が `\c` で終わっている場合を除き、*send* フィールドの送信の後には必ずキャリッジリターンが伴うという点に注意してください。

次に示すのは、*expect-send* 文字列を使用する `Systems` ファイルエントリの例です。

```
System-Name  Time  Type  Speed  Phone     Chap-Script
sonora Any ACUEC 9600 2223333 "" \r \r ogin:-BREAK-ogin: Puucpx ssword: xyzzy
```

この例は、ローカルホストの UUCP に、2 個のキャリッジリターンを送ってから `ogin:` (`Login:` という場合もあるため) を待つように指示しています。`ogin` を受信しなかった場合は、`BREAK` を送ります。`ogin:` を受信した場合は、ログイン名 `Puucpx` を送ります。`ssword:` (`Password:` を表す) を受け取ったら、パスワード `xyzzy` を送ります。

表 27–2 に、便利なエスケープ文字をいくつか紹介します。

表 27–2 Systems ファイルのチャットスクリプトで使用されるエスケープ文字

エスケープ文字	説明
\b	バックスペース文字を送信または想定する
\c	文字列の末尾で使用すると、普通なら送信されるキャリッジリターンが抑止される。その他の場合は無視される
\d	後続の文字を送る前に 1 ～ 3 秒の遅延が生じる
\E	エコーチェックを開始する (これ以降は、1 文字送信するたびに、その文字が受信されるのを待つ。以後の作業は、これを受信してから行われる)
\e	エコーチェックをオフにする
\H	ハングアップを 1 回無視する。このオプションはコールバックモデム用に使用する
\K	BREAK 文字を送信する
\M	CLOCAL フラグをオンにする
\m	CLOCAL フラグをオフにする
\n	改行文字を送信または想定する
\N	NULL 文字 (ASCII NUL) を送信する
\p	約 1/4 秒間または 1/2 秒間、一時停止する
\r	キャリッジリターンを送信または想定する
\s	スペース文字を送信または想定する
\t	タブ文字を送信または想定する
EOT	EOT とそれに続く 2 個の改行文字を送信する

表 27–2　Systems ファイルのチャットスクリプトで使用されるエスケープ文字　続く

エスケープ文字	説明
BREAK	ブレーク文字を送信する
ddd	8 進数 (*ddd*) で表される文字を送信または想定する

チャットスクリプトを使用したダイヤルバックの有効化

組織によっては、リモートコンピュータからの呼び出しを処理するダイヤルインサーバーを設定する場合があります。たとえば、コールバックモデムを持つダイヤルインサーバーを配備し、社員が自宅のコンピュータから呼び出せるようにすることができます。ダイヤルインサーバーは、リモートマシンを識別すると、そのリモートマシンとのリンクを切断し、逆にそのリモートマシンを呼び出して、通信リンクが再確立されます。

Systems ファイルのチャットスクリプトで、コールバックが必要な箇所で \H オプションを使用することにより、コールバックの操作を簡素化することができます。ダイヤルインサーバーのハングアップが予想される箇所で、expect 文字列の一部として \H を使用します。

たとえば、ダイヤルインサーバーを呼び出すチャットスクリプトに、次のような文字列が含まれているとします。

```
INITIATED\Hogin:
```

ローカルホストの UUCP ダイヤル機能は、ダイヤルインサーバーから INITIATED という文字列を受け取ることを想定しています。INITIATED 文字列を受け取ると、ダイヤル機能は、ダイヤルインサーバーがハングアップするまで、その後受信するすべての文字をフラッシュします。またダイヤル機能は、expect 文字列のその次の部分、つまり ogin: という文字列がダイヤルインサーバーから送られてくるのを待ちます。ogin: を受け取ると、ダイヤル機能はチャットスクリプトを先へ進めます。

上記のサンプルでは \H の前後に文字列が指定されていますが、これらはなくてもかまいません。

UUCP ハードウェアフロー制御

擬似送信文字列 STTY=value を使用して、モデム特性を設定することもできます。たとえば、STTY=crtscts を使用すると、ハードウェアフロー制御が可能になります。 STTY はすべての stty モードを受け入れます。詳細は、stty(1) と termio(7I) のマニュアルページを参照してください。

次の例は、Systems ファイルのエントリ内でハードウェアフロー制御を指定しています。

```
System-Name  Time  Type  Speed  Phone     Chap-Script
unix Any ACU 2400 12015551212 "" \r login:-\r-login:-\r-login:
nuucp password: xxx "" \ STTY=crtscts
```

擬似送信文字列は、Dialers ファイルのエントリの中でも使用できます。

UUCP パリティの設定

場合によっては、呼び出そうとしているシステムがポートのパリティを検査し、パリティに誤りがあると回線を切断することがあります。このようなときは、パリティのリセットが必要になることがあります。expect-send の文字列ペアとして "" P_ZERO を使用すると、上位ビット (パリティビット) が 0 に設定されます。たとえば次のように指定します。

```
System-Name  Time  Type  Speed  Phone     Chap-Script
unix Any ACU 2400 12015551212 "" P_ZERO "" \r login:-\r-login:-\r-login:
nuucp password: xxx
```

同様に、P_EVEN はパリティを偶数 (デフォルト) に設定し、P_ODD は奇数パリティを設定し、P_ONE はパリティビットを 1 に設定します。

パリティ設定は、チャットスクリプトのどこにでも挿入できます。この設定は、チャットスクリプト内の "" P_ZERO より後にあるすべての情報に適用されます。これは、Dialers ファイルのエントリの中でも使用できます。

UUCP /etc/uucp/Devices ファイル

/etc/uucp/Devices ファイルには、リモートコンピュータへのリンクを確立するために使用できるすべてのデバイスに関する情報が入っています。この種のデバイスには、ACU (が含まれます)、直接リンク、ネットワーク接続などがあります。

次に示す /etc/uucp/Devices のエントリは、ポート A に接続され 38,400 bps で動作する US Robotics V.32bis モデムを表しています。

```
Type   Line  Line2 Class Dialer-Token-Pairs
 ACUEC  cua/a -     38400 usrv32bis-ec
```

各フィールドについて、次のセクションで説明します。

UUCP Type フィールド

このフィールドは、デバイスが確立するリンクの種類を記述します。このフィールドには次のセクションに示すキーワードのいずれかを入れることができます。

キーワード Direct

キーワード Direct は、主として cu 接続用のエントリ内で使用されます。このキーワードは、このリンクが他のコンピュータまたはポートセレクタへの直接リンクであることを示します。cu の -l オプションで参照したい各回線について、それぞれ独立したエントリを作成する必要があります。

キーワード ACU

キーワード ACU は、(cu、UUCP、または PPP を介した) リモートコンピュータへのリンクを、モデムを介して確立することを示します。このモデムは、直接ローカルコンピュータに接続しているものでも、ポートセレクタを介して間接的に接続しているものでもかまいません。

ポートセレクタ

これは、ポートセレクタの名前で置き換えるものとして、Type フィールド内で使用される変数です。ポートセレクタは、ネットワークに接続されたデバイスで、呼び出し側モデムの名前を要求し、アクセスを許可します。/etc/uucp/Dialers ファイルに入っている呼び出しスクリプトは、micom ポートセレクタと develcon ポートセレクタについてのものだけです。ユーザーは、Dialers ファイルに独自のポートセレクタエントリを追加できます (詳細は、563ページの「UCCP /etc/uucp/Dialers ファイル」を参照してください)。

Sys-Name

Type フィールド内のこの変数は、特定のマシンの名前で置き換えられます。これは、リンクがこのマシンへの直接リンクであることを示します。この命名スキーマは、この Devices エントリ内の行と、コンピュータ Sys-Name についての /etc/uucp/Systems ファイルエントリを対応付けるために使用されます。

Type フィールドと /etc/uucp/Systems ファイル

例 27–5 は、/etc/uucp/Devices のフィールドと、/etc/uucp/Systems のフィールドの対応を示しています。各列の見出しは Devices ファイルに対応するものです。

フィールドの書体を変えて示したように、Devices ファイルの Type フィールドで使用されているキーワードは、Systems ファイルエントリの 3 番目のフィールドと突き合わされます。Devices ファイルの Type フィールドには ACUEC というエントリが入っており、これは自動呼び出し装置、つまりこの例では V.32bis モデムを示しています。この値は、Systems ファイルの 3 番目のフィールドと突き合わされます。このフィールドにも ACUEC というエントリが入っています (詳細は、547ページの「UUCP /etc/uucp/Systems ファイル」を参照してください)。

例 27–5 Type フィールドと /etc/uucp/Systems ファイルの対応関係

```
File Name Type   Line  Line2 Class Dialer-Token-Pairs

Devices   ACUEC  cua/a -     38400 usrv32bis-ec
```

(続く)

```
System    nubian Any   ACUEC 38400 9998888 '''' \d\d\r\n\c-ogin-\r\n\c-ogin.......
```

UUCP Line フィールド

このフィールドには、Devices エントリに対応付けられる回線 (ポート) のデバイス名が入ります。たとえば、特定のエントリに対応付けられているモデムが /dev/cua/a (シリアルポート A) に接続されている場合、このフィールドに入力する名前は cua/a です。Line フィールドでオプションのモデム制御フラグ M を使用すると、キャリアを待たないでデバイスをオープンすることを指定できます。たとえば次のようになります。

```
cua/a,M
```

UUCP Line2 フィールド

このフィールドは、フィールドの数を合わせるために存在しているだけです。ここには常にダッシュ (-) を指定します。Line2 フィールドを使用するのは 801 型のダイヤラですが、この種類は Solaris 環境ではサポートされていません。801 型以外のダイヤラは通常はこの設定を使用しませんが、このフィールドにダッシュだけは入れておく必要があります。

UUCP Class フィールド

Type フィールドでキーワード ACU または Direct を使用した場合は、Class フィールドにはデバイスの速度が入ります。ただし、このフィールドには、ダイヤラのクラス (Centrex または Dimension PBX) を区別するために、1 個の英字と速度値を含めることができます (たとえば、C1200、D1200)。

大規模な事業所では複数種の電話ネットワークを使用することが多いため、このような指定が必要になります。たとえば、1 つのネットワークは事業所内の内線通信専用に使用し、もう 1 つのネットワークは外線通信に使用するといった方式が考えられます。このような場合は、内線回線と外線回線とを区別する必要があります。

Devices ファイルの Class フィールドで使用するキーワードは、Systems ファイルの Speed フィールドと突き合わされます。

例 **27–6** UUCP Class フィールド

```
File Name  Type   Line  Line2  Class  Dialer-Token-Pairs

Devices    ACU    cua/a -      D2400  hayes
```

どのような速度でも使用できるデバイスでは、Class フィールドにキーワード Any を使用します。Any を使用した場合は、回線は、Systems ファイルの Speed フィールドで要求された任意の速度に適合します。このフィールドが Any で、Systems ファイルの Speed フィールドも Any である場合は、速度はデフォルトの 2400bps となります。

UUCP Dialer-Token-Pairs フィールド

Dialer-Token-Pairs (DTP) フィールドには、ダイヤラの名前とそれに渡すトークンが入ります。DTP フィールドの構文は次のとおりです。

dialer token [dialer token]

dialer の部分は、モデムかポートモニターの名前あるいは直接リンクデバイスの場合は direct または uudirect です。ダイヤラとトークンのペアはいくつでも指定できます。指定しなかった場合は、Systems ファイル内の関連エントリから取得されます。*token* 部は、dialer 部の直後に指定できます。

対応するダイヤラによっては、最後のダイヤラとトークンのペアはない場合もあります。ほとんどの場合は、最後のペアには *dialer* 部だけが含まれます。*token* 部は、対応する Systems ファイルエントリの Phone フィールドから取得されます。

dialer 部の有効エントリは、Dialers ファイル内で定義されているものか、いくつかの特殊ダイヤラタイプのうちの 1 つとなります。これらの特殊ダイヤラタイプはコンパイル時にソフトウェア中に組み込まれているので、Dialers ファイル内に該当エントリがなくても使用できます。表 27–3 に、特殊ダイヤラタイプを示します。

表 27–3　ダイヤラとトークンのペア

TCP	TCP/IP ネットワーク
TLI	トランスポートレベルインタフェースネットワーク (STREAMS を使用しないもの)
TLIS	トランスポートレベルインタフェースネットワーク (STREAMS を使用するもの)

詳細は、562ページの「UUCP Devices ファイル内のプロトコル定義」を参照してください。

Dialer-Token-Pairs フィールドの構造

DTP フィールドの構造は、エントリに対応するデバイスに応じて 4 通りに設定できます。

- 直接接続モデム

  コンピュータのポートにモデムが直接接続されている場合は、対応する Devices ファイルエントリの DTP フィールドに入るペアは 1 つだけです。このペアは、通常はモデムの名前です。この名前は、Devices ファイルの特定のエントリと、Dialers ファイル内のエントリとを対応付けるために使用されます。したがって、Dialer フィールドは、Dialers ファイルエントリの最初のフィールドに一致している必要があります。

例 27–7　直接接続モデム用 Dialer フィールド

```
Dialers   hayes =,-,  ""          \\dA\pTE1V1X1Q0S2=255S12=255\r\c
                                  \EATDT\T\r\c CONNECT
```

Devices ファイルエントリの DTP フィールドには、dialer 部 (hayes) だけが示されている点に注意してください。これは、ダイヤラに渡す *token* (この例では電話番号) が、Systems ファイルエントリの Phone フィールドから取得されることを意味します (例 27–9 で説明する \T が暗黙で指定されます)。

- 直接リンク – 特定のコンピュータへの直接リンクの場合は、対応するエントリの DTP フィールドには、キーワード direct が入ります。これは、Direct、*Sys-Name* の両方の直接リンクエントリにもあてはまります (556 ページの「UUCP Type フィールド」を参照)。

- 同じポートセレクタ上のコンピュータ – 通信したいコンピュータが、ローカルコンピュータと同じポートセレクタスイッチ上にある場合は、ローカルコンピュータはまずそのスイッチにアクセスする必要があります。そのスイッチが、相手のコンピュータとの接続を確立します。この種のエントリでは、ペアは1 つだけです。*dialer* 部が Dialers ファイルのエントリと突き合わされます。

例 27–8 同一ポートセレクタ上のコンピュータ用 UUCP Dialer フィールド

```
Dialers    develcon ,""   ""           \pr\ps\c est:\007 \E\D\e \007
```

token 部が空である点に注意してください。これは、この部分が Systems ファイルから取得されることを示しています。このコンピュータ用の Systems ファイルエントリには、Phone フィールドにトークンが含まれています。このフィールドは、通常、コンピュータの電話番号用として確保されています (547ページの「UUCP /etc/uucp/Systems ファイル」を参照してください)。この種類の DTP にはエスケープ文字 (\D) が含まれています。これは、Phone フィールドの内容が、Dialcode ファイル内の有効エントリとして解釈されないことを保証します。

- ポートセレクタに接続しているモデム – ポートセレクタに高速モデムが接続されている場合は、ローカルコンピュータはまずポートセレクタスイッチにアクセスする必要があります。そして、そのスイッチがモデムとの接続を確立します。この種類のエントリには、ダイヤラとトークンのペアが2 つ必要です。各ペアの *dialer* 部 (エントリの 5 番目と 7 番目のフィールド) が、Dialers ファイル内のエントリと突き合わされます。

例 27–9 ポートセレクタに接続されたモデム用 UUCP Dialer フィールド

```
Dialers    develcon ""    ""     \pr\ps\c  est:\007    \E\D\e      \007

Dialers    ventel   =&-%   t""   \r\p\r\c  $           <K\T%\r>\c  ONLINE!
```

最初のペアでは、develcon がダイヤラで、vent が Develcon スイッチに渡されるトークンです。トークンは、コンピュータに接続するデバイス (たとえば Ventel モデム) をダイヤラに指示しています。各スイッチごとに設定が異なることがあるので、このトークンは各ポートセレクタに固有のものにします。Ventel モデムが接続されると、第 2 のペアがアクセスされます。このペアでは、Ventel がダイヤラで、トークンは Systems ファイルから取得されます。

DTP フィールドでは 2 つのエスケープ文字が使用できます。

- \T – Phone (*token*) フィールドを、/etc/uucp/Dialcodes ファイルを使用して解釈することを指定します。通常、モデム (Hayes、US Robotics など) に対応する各呼び出しスクリプトについて、/etc/uucp/Dialers ファイルにこのエスケープ文字を組み込みます。したがって、呼び出しスクリプトがアクセスされるまでは、解釈は行われません。
- \D – Phone (*token*) フィールドを、/etc/uucp/Dialcodes ファイルを使用して解釈しないことを指定します。Devices エントリの末尾にエスケープ文字が何も指定されていないときは、デフォルトで \D があるものと想定します。\D は、/etc/uucp/Dialers ファイルの中でも、ネットワークスイッチ (develcon と micom) に関連したエントリで使用されます。

UUCP Devices ファイル内のプロトコル定義

/etc/uucp/Devices では、各デバイスに使用するプロトコルを定義できます。通常は、デフォルトを使用するか、または呼び出そうとしている個々のシステムごとにプロトコルを定義できるので、この指定は不要です (547ページの「UUCP /etc/uucp/Systems ファイル」を参照してください)。プロトコルを指定する場合は、次の形式を使用する必要があります。

```
Type,Protocol [parameters]
```

たとえば、TCP/IP プロトコルを指定するには、TCP,te を入力します。

表 27–4 に、Devices ファイルで使用できるプロトコルを示します。

表 27–4 /etc/uucp/Devices で使用されるプロトコル

プロトコル	説明
t	このプロトコルは、TCP/IP や、その他の信頼性のある接続を介した伝送に、最もよく使用される。このプロトコルはエラーのない伝送を前提としている
g	UUCP のネイティブプロトコル。低速で信頼性があり、ノイズの多い電話回線を介した伝送に適している

表 27–4　/etc/uucp/Devices で使用されるプロトコル　続く

プロトコル	説明
e	このプロトコルは、(TCP/IP のようなバイトストリーム指向ではなく) メッセージ指向でエラーのないチャネルを介した伝送を前提としている
f	このプロトコルは X.25 接続を介した伝送に使用される。このプロトコルは、データストリームのフロー制御に関係している。特に X.25/PAD リンクなどのように、完全に (またはほとんど) エラーがないことが保証されるリンクでの使用を意図している。検査合計はファイル全体についてのみ実施される。伝送が失敗した場合は、受信側は再伝送を要求する

次に、デバイスエントリ用のプロトコル指定の例を示します。

```
TCP,te - - Any TCP -
```

この例は、デバイス TCP について t プロトコルの使用を試みるように指示しています。相手側がそれを拒否した場合は、e プロトコルが使用されます。

e と t のどちらも、モデムを介した通信には適していません。モデムがエラーのない伝送を保証するものであったとしても、モデムと CPU との間でデータが失われる可能性があります。

UUCP /etc/uucp/Dialers ファイル

/etc/uucp/Dialers ファイルには、よく使用される多くのモデムに関するダイヤリング指示が入っています。標準外のモデムの使用や、UUCP 環境のカスタマイズを予定している場合以外は、通常このファイルのエントリの変更や追加は必要ありません。しかし、このファイルの内容と、Systems ファイルや Devices ファイルとの関係は理解しておく必要があります。

このファイルの中のテキストは、回線をデータ転送に使用できるようにするために、最初に行わなければならない対話を指定します。チャットスクリプトと呼ばれるこの対話は、通常は送受信される一連の ASCII 文字列で、電話番号をダイヤルするためによく使用されます。

556ページの「UUCP /etc/uucp/Devices ファイル」の例に示したように、Devices ファイルの 5 番目のフィールドは、Dialers ファイルへのインデックスか、または特殊ダイヤラタイプ (TCP、TLI、または TLIS) です。 uucico デー

モンは、Devices ファイルの 5 番目のフィールドを、Dialers ファイルの各エントリの最初のフィールドと突き合わせます。さらに、Devices の 7 番目の位置から始まる奇数番号の各フィールドは、Dialers ファイルへのインデックスとして使用されます。これらが一致すると、その Dialers のエントリがダイヤラ対話を行うために解釈されます。

Dialers ファイルの各エントリの形式は次のとおりです。

dialer	*substitutions*	*expect-send*

例 27–10 に、US Robotics V.32bis モデム用のエントリの例を示します。

例 27–10 /etc/uucp/Dialers ファイルのエントリ

```
Dialer        Substitution  Expaec-Send
usrv32bis-e   =,-,   ""     dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r
                            \EATDT\T\r\c CONNECT\s14400/ARQ STTY=crtscts
```

Dialer フィールドは、Devices ファイルの中の 5 番目以降の奇数番号のフィールドと突き合わされます。Substitutions フィールドは変換文字列です。各文字ペアの最初の文字が 2 番目の文字に変換されます。通常これは = と - を、「発信音待ち」と「一時停止」用としてダイヤラが必要とする文字に変換するために使用されます。

それ以降の expect-send の各フィールドは文字列です。

例 27–11 に、Dialers ファイルのエントリの例をいくつか示します。これは、Solaris インストールプログラムの一環として UUCP をインストールしたときに提供されるファイルです。

例 27–11 /etc/uucp/Dialers の抜粋

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK

ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!

vadic =K-K "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE

develcon "" "" \pr\ps\c est:\007

\E\D\e \n\007 micom "" "" \s\c NAME? \D\r\c GO

hayes =,-, "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r \EATDT\T\r\c CONNECT
```

(続く)

```
#   Telebit TrailBlazer
tb1200 =W-, "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=2\r\c OK\r
\EATDT\T\r\c CONNECT\s1200
tb2400 =W-, "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=3\r\c OK\r
\EATDT\T\r\c CONNECT\s2400
tbfast =W-, "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=255\r\c OK\r
\EATDT\T\r\c CONNECT\sFAST

# USrobotics, Codes, and DSI modems

dsi-ec  =,-,    "" \dA\pTE1V1X5Q0S2=255S12=255*E1*F3*M1*S1\r\c OK\r \EATDT\T\r\c
CONNECT\sEC STTY=crtscts,crtsxoff

dsi-nec =,-,    "" \dA\pTE1V1X5Q0S2=255S12=255*E0*F3*M1*S1\r\c OK\r \EATDT\T\r\c CONNECT
STTY=crtscts,crtsxoff

usrv32bis-ec =,-,  "" \dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r \EATDT\T\r\c
CONNECT\s14400/ARQ STTY=crtscts,crtsxoff

usrv32-nec =,-, "" \dA\pT&FE1V1X1Q0S2=255S12=255&A0&H1&M0&B0&W\r\c OK\r \EATDT\T\r\c
CONNECT STTY=crtscts,crtsxoff

codex-fast =,-, "" \dA\pT&C1&D2*MF0*AA1&R1&S1*DE15*FL3S2=255S7=40S10=40*TT5&W\r\c OK\r
\EATDT\T\r\c CONNECT\s38400 STTY=crtscts,crtsxoff

tb9600-ec =W-,  "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=6\r\c OK\r
\EATDT\T\r\cCONNECT\s9600 STTY=crtscts,crtsxoff

tb9600-nec =W-, "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=6S180=0\r\c OK\r \EATDT\T\r\c
CONNECT\s9600 STTY=crtscts,crtsxoff
```

表 27–5 に、Dialers ファイルの send 文字列でよく使用されるエスケープ文字を示します。

表 27–5 /etc/uucp/Dialers で使用するエスケープ文字

文字	説明
\b	バックスペース文字を送信または想定する
\c	改行、キャリッジリターンを抑止する
\d	遅延 (約 2 秒)
\D	Dialcodes 変換なしの電話番号またはトークン

表 27–5　/etc/uucp/Dialers で使用するエスケープ文字　続く

文字	説明
\e	エコーチェックを使用しない
\E	エコーチェックを使用する (低速デバイス用)
\K	ブレーク文字を挿入する
\n	改行文字を送信する
\nnn	8 進数値を送信する。使用できるその他のエスケープ文字は、547 ページの「UUCP /etc/uucp/Systems ファイル」を参照
\N	NULL 文字 (ASCII NUL) を送信または想定する
\p	一時停止 (約 12 ～ 14 秒)
\r	リターン
\s	スペース文字を送信または想定する
\T	Dialcodes 変換を伴う電話番号またはトークン

次に示すのは Dialers ファイルの penril エントリです。

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
```

最初に、電話番号引数の置換メカニズムが確立されます。その結果、= はすべて W (発信音待ち) で置き換えられ、- はすべて P (一時停止) で置き換えられるようになります。

上記の行の残りの部分に指定されているハンドシェークの働きは、次のとおりです。

- "" – 何も待たない (つまり次へ進む)
- \d – 2 秒間の遅延の後キャリッジリターンを送信する
- > – > を待つ
- Q\c – キャリッジリターンを付けずに Q を送信する
- : – : を待つ

- \d- – 2 秒間の遅延の後 - とキャリッジリターンを送信する
- > – > を待つ
- s\p9\c – s を送信し、一時停止し、9 を送信するが、キャリッジリターンは送信しない
-)-W\p\r\ds\p9\c-) –) を待つ。) が受信されない場合は、- 文字の間の文字列を処理する。つまり、W を送信し、一時停止し、キャリッジリターンを送信し、遅延し、s を送信し、一時停止し、9 を送信し、キャリッジリターンを送信しないで) を待つ
- y\c – キャリッジリターンを付けずに y を送信する
- : – : を待つ
- \E\TP - エコーチェックを有効にする (この時点以降は、1 文字送信すると、その文字が受信されるまでほかの作業を行わない)。次に電話番号を送信する。\T は、引数として渡された電話番号をとり、Dialcodes 変換を適用し、このエントリのフィールド 2 で指定されたモデム機能変換を適用することを意味する。次に、P とキャリッジリターンを送信する
- > – > を待つ
- 9\c – 改行を付けずに 9 を送信する
- OK – 文字列 OK を待つ

UUCP ハードウェアフロー制御

擬似送信文字列 STTY=value を用いることによっても、モデムの特性を設定できます。たとえば、STTY=crtscts は出力ハードウェアフロー制御、STTY=crtsxoff は入力ハードウェアフロー制御を使用可能にし、STTY=crtscts、crtsxoff は入出力両方のハードウェアフロー制御を使用可能にします。

STTY はすべての stty モードを受け入れます。詳細は、stty(1) と termio(7I) のマニュアルページを参照してください。

次の例は、Dialers ファイルエントリ内でハードウェアフロー制御を使用可能にしています。

```
dsi =,--, "" \dA\pTE1V1X5Q0S2=255S12=255*E1*F3*M1*S1\r\c OK\r \EATDT\T\r\c
CONNECT\sEC STTY=crtscts
```

この擬似送信文字列は、Systems ファイルのエントリ中でも使用できます。

UUCP パリティの設定

場合によっては、呼び出そうとしているシステムがポートのパリティを検査し、パリティに誤りがあると回線を切断することがあります。そのため、パリティのリセットが必要になります。expect-send の対を成す文字列として P_ZERO を使用すると、パリティが 0 に設定されます。

```
foo =,-, "" P_ZERO "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r\EATDT\T\r\c CONNECT
```

同様に、P_EVEN はパリティを偶数 (デフォルト) に、P_ODD はパリティを奇数に設定し、P_ONE はパリティを 1 に設定します。この擬似送信文字列は、Systems ファイルのエントリの中でも使用できます。

その他の基本的な UUCP 構成ファイル

この節で紹介するのは、基本的な UUCP 構成を行うときに、Systems、Devices、および Dialers ファイルに加えて使用できるファイルです。

UUCP /etc/uucp/Dialcodes ファイル

/etc/uucp/Dialcodes ファイルにより、/etc/uucp/Systems ファイルの Phone フィールドで使用するダイヤルコードの省略名を定義できます。Dialcodes ファイルは、同じサイトにある複数のシステムが使用する基本的な電話番号について、付加的な情報を指定するために使用できます。

このファイルのエントリの形式は次のとおりです。

abbreviation dial-sequence

abbreviation は、Systems ファイルの Phone フィールドで使用される省略名で、*dial-sequence* は、個々の Systems ファイルのエントリがアクセスされたときにダイヤラに渡されるダイヤルシーケンスです。表 27–6 に、この 2 つのファイル間の対応関係を示します。

表 27–6 Dialcodes ファイルと Systems ファイルの間の対応関係

	フィールド名					
Dialcodes	*Abbreviation*	Dial-Sequence				
Systems	System-Name	Time	Type	Speed	Phone	Chat-Script

表 27–7 に示すのは、Dialcodes ファイルのエントリの例です。

表 27–7 Dialcode ファイルのエントリ

Abbreviation	**Dial-sequence**
NY	1=212
jt	9+847

最初の行の NY は、Systems ファイルの Phone フィールドで使用される省略名です。Systems ファイルのエントリは、たとえば次のようになります。

```
NY5551212
```

uucico は、Systems ファイルから NY を読み取ると、Dialcodes ファイルから NY を探し、それに該当するダイヤルシーケンス 1=212 を取得します。これは、New York City への電話呼び出しに必要なダイヤルシーケンスです。このシーケンスは、1 という番号と、一時停止して次の発信音を待つことを示す等号 (=) と、地域コード 212 で構成されています。uucico はこの情報をダイヤラに送り、再び Systems ファイルに戻って残りの電話番号 5551212を処理します。

jt 9=847- というエントリは、Systems ファイル内の jt7867 などのような Phone フィールドを取り扱います。uucico は、jt7867 を含むエントリを Systems ファイルから読み取ると、ダイヤラとトークンのペアの中のトークンが \T であれば、9=847-7867 というシーケンスをダイヤラに送ります。

UUCP /etc/uucp/Sysfiles ファイル

/etc/uucp/Sysfiles ファイルでは、uucp と cu が Systems、 Devices、 Dialers ファイルとして使用する別のファイルを割り当てます (cu についての詳細

は、cu(1C) のマニュアルページを参照してください)。Sysfiles は次の目的に使用できます。

- 別の Systems ファイルにより、uucp のサービスとは異なるアドレスに対してログインサービスを要求できます。
- 別の Dialers ファイルにより、cu と uucp で異なるハンドシェークを割り当てることができます。
- 複数の Systems、Dialers、Devices ファイル。特に Systems ファイルはサイズが大きくなるので、いくつかの小さいファイルに分割しておくと便利です。

Sysfiles ファイルの形式は次のとおりです。

```
service=w systems=x:x dialers=y:y devices=z:z
```

w には、uucico、cu、またはその両方をコロンで区切って指定します。*x* には、Systems ファイルとして使用される 1 つまたは複数のファイルをコロンで区切って指定します。これらは指定された順序で読み込まれます。*y* は Dialers ファイルとして使用される 1 つまたは複数のファイルで、*z* は Devices ファイルとして使用される 1 つまたは複数のファイルです。

フルパスで指定しない限り、各ファイル名は /etc/uucp ディレクトリからの相対パスとみなされます。

次に示すのは、標準の /etc/uucp/Systems に加えて使用するローカル Systems ファイル (Local_Systems) を定義する /etc/uucp/Sysfiles の例です。

```
service=uucico:cu systems=Systems :Local_Systems
```

/etc/uucp/Sysfiles の中にこのエントリがある場合、uucico と cu はどちらも、まず標準 /etc/uucp/Systems ファイルを調べます。呼び出そうとしているシステムのエントリがそのファイル内にないか、またはそのファイル内の該当エントリの処理に失敗した場合は、/etc/uucp/Local_Systems が調べられます。

上記のエントリの場合は、cu と uucico は、Dialers ファイルと Devices ファイルを共有します。

uucico サービス用と cu サービス用に別の Systems ファイルを定義した場合は、マシンは 2 つの異なる Systems のリストを持つことになります。uucico リストは uuname コマンドを使用して表示でき、cu リストは uuname –C コマンドを使用して表示できます。このファイルのもう 1 つの例として、代替ファイルの方を先に調べ、デフォルトファイルは必要なときだけ調べる場合を示します。

```
service=uucico systems=Systems.cico:Systems
  dialers=Dialers.cico:Dialers \
devices=Devices.cico:Devices
  service=cu systems=Systems.cu:Systems \
dialers=Dialers.cu:Dialers \
  devices=Devices.cu:Devices
```

UUCP /etc/uucp/Sysname ファイル

UUCP を使用するすべてのマシンは、ノード名と呼ばれる識別名を持っている必要があります。この名前は、リモートマシンの /etc/uucp/Systems ファイルに、チャットスクリプトやその他の識別情報とともに格納されます。通常は、UUCP は、uname -n コマンドから返されるものと同じノード名を使用し、TCP/IP でもこの名前を使用します。

/etc/uucp/Sysname ファイルを作成することによって、TCP/IP ホスト名とは別の UUCP ノード名を指定できます。このファイルには、ローカルシステムの UUCP ノード名が入った 1 行のエントリが含まれています。

UUCP /etc/uucp/Permissions ファイル

/etc/uucp/Permissions ファイルは、ログイン、ファイルアクセス、およびコマンド実行に関するリモートコンピュータのアクセス権を指定します。リモートコンピュータがファイルを要求する権限と、ローカルマシンでキューに入れられたファイルを受け取る権限を制限するオプションがあります。また、リモートマシンがローカルコンピュータ上で実行できるコマンドを指定するオプションもあります。

エントリの UCCP 構造

各エントリは 1 行の論理行で、行末にバックスラッシュ (\) がある場合は次の行と継続していることを示します。エントリは、スペースで区切られたオプションから構成されます。各オプションは、次の形式の名前と値のペアです。

name=value

values はコロンで区切ってリストとすることもできます。オプション指定の中では、スペースは使用できないので注意してください。

コメント行はポンド記号 (#) で始まり、その行の改行文字までの全部分を占めます。空行は無視されます (複数行エントリの中の空行も同じです)。

Permissions ファイルのエントリには 2 つの種類があります。

- LOGNAME – リモートマシンがローカルマシンにログインする (呼び出す) ときに有効なアクセス権を指定する。

注 - リモートマシンがローカルマシンを呼び出すとき、固有のログインと検証可能なパスワードを使用しない限り、そのリモートマシンの識別情報は正確なものとはなりません。

- MACHINE – ローカルマシンがリモートコンピュータにログインする (呼び出す) ときに有効なアクセス権を指定する。

LOGNAME エントリには LOGNAME オプションが含まれ、MACHINE エントリには MACHINE オプションが含まれます。1 つのエントリに両方のオプションを含めることもできます。

UUCP の考慮事項

Permissions ファイルを使用して、リモートコンピュータに付与されているアクセスのレベルを制限するときは、以下のことを考慮に入れる必要があります。

- リモートコンピュータが、UUCP 通信を目的としてログインするために使用するすべてのログイン ID は、1 つの LOGNAME エントリだけに指定されていなければならない。
- 呼び出されたサイトの名前が MACHINE エントリにない場合、そのサイトには次に示すデフォルトのアクセス権または制約が適用される。
  - ローカルの送信要求と受信要求は実行される
  - リモートコンピュータは、ローカルコンピュータの /var/spool/uucppublic ディレクトリにファイルを送信できる
  - リモートコンピュータがローカルコンピュータで実行するために送信するコマンドは、デフォルトのコマンドのどれかでなければならない (通常は rmail)

UUCP REQUEST オプション

リモートコンピュータがローカルコンピュータを呼び出し、ファイルの受信を要求したときに、その要求を承認することも拒否することもできます。REQUEST オプ

ションは、リモートコンピュータがローカルコンピュータからのファイル転送を要求できるかどうかを指定します。REQUEST=yes は、リモートコンピュータがローカルコンピュータからのファイル転送を要求できることを指定します。REQUEST=no は、リモートコンピュータがローカルコンピュータからのファイルの受信を要求できないことを指定します。後者は、REQUEST オプションを指定しなかった場合に使用されるデフォルト値です。REQUEST オプションは、LOGNAME エントリ (リモートコンピュータがローカルコンピュータを呼び出す場合) と、MACHINE エントリ (ローカルコンピュータがリモートコンピュータを呼び出す場合) のどちらにも使用できます。

UUCP SENDFILES オプション

リモートコンピュータがローカルコンピュータを呼び出す作業を完了した後で、ローカルコンピュータのキュー中のリモートコンピュータ用の作業を受け取ろうとすることがあります。SENDFILES オプションは、ローカルコンピュータが、リモートコンピュータ用にキューに入れた作業を送信できるかどうかを指定します。

文字列 SENDFILES=yes は、リモートコンピュータが LOGNAME オプションに指定されている名前の 1 つを使用してログインしていれば、ローカルコンピュータがキューに入れた作業を送信できることを指定します。/etc/uucp/Systems の Time フィールドに Never を入力してある場合は、この文字列の使用は必須です。Never を指定すると、ローカルマシンは受動モードに設定され、相手のリモートコンピュータへの呼び出しを開始することはできなくなります (詳細は、547ページの「UUCP /etc/uucp/Systems ファイル」を参照してください)。

文字列 SENDFILES=call は、ローカルコンピュータがリモートコンピュータを呼び出したときに限り、ローカルコンピュータのキュー中のファイルを送信することを指定します。call の値は SENDFILES オプションのデフォルト値です。MACHINE エントリはリモートコンピュータへの呼び出しを送る場合に適用されるものなので、このオプションが意味を持つのは LOGNAME エントリの中で使用した場合だけです。MACHINE エントリでこのオプションを使用しても無視されます。

UUCP MYNAME オプション

このオプションを使用すると、hostname コマンドから戻される TCP/IP ホスト名以外に、固有の UUCP ノード名をローカルシステムに与えることができます。たとえば、偶然に他のシステムと同じ名前をローカルホストに付けてしまった場合などに、Permissions ファイルの MYNAME オプションを指定できます。あるいは、

たとえば、自分の所属組織が widget という名前で認識されるようにしたいが、すべてのモデムが gadget というホスト名を持つマシンに接続されているという場合は、gadget の Permissions ファイルに次のようなエントリを含めることができます。

```
service=uucico systems=Systems.cico:Systems
  dialers=Dialers.cico:Dialers \
  devices=Devices.cico:Devices
service=cu systems=Systems.cu:Systems \
  dialers=Dialers.cu:Dialers \
  devices=Devices.cu:Devices
```

これで、システム world は、あたかも widget にログインしているかのようにマシン gadget にログインできます。ローカルマシンから world マシンを呼び出したときにも、world が widget という別名で認識するようにしたい場合は、次のようなエントリを作成します。

```
MACHINE=world MYNAME=widget
```

MYNAME オプションにより、ローカルマシンが自分自身を呼ぶこともできるので、テストの目的にも利用できます。しかし、このオプションはマシンの実際の識別情報を隠す目的にも使用できてしまうので、577ページの「UUCP VALIDATE オプション」で述べる VALIDATE オプションを使用するようにしてください。

UUCP READ オプションと WRITE オプション

これらのオプションは、uucico がファイルシステムのどの部分を読み書きできるかを指定します。READ オプションと WRITE オプションは、MACHINE エントリと LOGNAME エントリのどちらにも使用できます。

次の文字列に示すように、READ オプションと WRITE オプションのどちらも、デフォルトは uucppublic ディレクトリです。

```
READ=/var/spool/uucppublic WRITE=/var/spool/uucppublic
```

文字列 READ=/ と WRITE=/ は、Other 権を持つローカルユーザーがアクセスできるすべてのファイルにアクセスできる権限を指定します。

これらのエントリの値は、コロンで区切ったパス名のリストです。READ オプションはリモート側からのファイル要求のためのものであり、WRITE オプションはリモート側からのファイル送出のためのものです。値の 1 つは、入力ファイルまたは出力ファイルのフルパス名の接頭辞でなければなりません。公共ディレクトリの他に

/usr/news にもファイルにも送出する権限を付与するには、WRITE オプションに次の値を指定します。

```
WRITE=/var/spool/uucppublic:/usr/news
```

パス名はデフォルトのリストに追加されるものではないので、READ オプションと WRITE オプションを使用するときはすべてのパス名を指定する必要があります。たとえば、WRITE オプションでパス名として /usr/news しか指定しなかったとすると、公共ディレクトリにファイルを送出する権限は失われます。

リモートシステムがどのディレクトリに読み書きのアクセスができるかは、注意して決定しなければなりません。たとえば、/etc ディレクトリには多数の重要なシステムファイルが入っているので、このディレクトリにファイルを送出する権限はリモートユーザーには付与しない方が賢明です。

UUCP NOREAD オプションと NOWRITE オプション

NOREAD オプションと NOWRITE オプションは、READ オプションと WRITE オプションおよびデフォルトに対する例外を指定します。たとえば次のようなエントリを指定したとします。

```
READ=/ NOREAD=/etc WRITE=/var/spool/uucppublic
```

これは、/etc ディレクトリ (およびこの下の各サブディレクトリ。このパス名は接頭辞であることを忘れないでください) の中のファイルを除くすべてのファイルの読み取りを許可しています。デフォルトの /var/spool/uucppublic ディレクトリへの書き込みだけを許可しています。NOWRITE も NOREAD オプションと同じ形で働きます。NOREAD オプションと NOWRITE オプションは、LOGNAME エントリと MACHINE エントリのどちらにも使用できます。

UUCP CALLBACK オプション

LOGNAME エントリの中で CALLBACK オプションを使用すると、呼び出し側システムがコールバックするまで、トランザクションをいっさい行わないことを指定できます。CALLBACK を設定する理由は 2 つあります。1 つはセキュリティを目的とするもので、マシンをコールバックすることで、それが正しいマシンであることを確認できます。 もう 1 つは課金を目的とするもので、大量のデータの伝送を行うときに、その長時間の呼び出しの料金を課すマシンを選択できます。

文字列 CALLBACK=yes は、ファイル転送を行う前に、ローカルコンピュータがリモートコンピュータをコールバックしなければならないということを指定します。

CALLBACK オプションのデフォルトは CALLBACK=no です。CALLBACK を yes に設定する場合は、呼び出し側に対応する MACHINE エントリの中で、以後の通信に影響を与えるアクセス権を指定する必要があります。これらのアクセス権は、LOGNAME の中で指定してはいけません。また同様に、リモートマシンがローカルホストについて設定した LOGNAME エントリの中で指定してもいけません。

注 - 2 つのサイトが互いに CALLBACK オプションを設定すると、通信が開始されないので注意してください。

UUCP COMMANDS オプション

注意 - COMMANDS オプションは、システムのセキュリティを低下させる恐れがあります。このオプションは十分に注意して使用してください。

COMMANDS オプションは、リモートコンピュータがローカルコンピュータ上で実行できるコマンドを指定するために、MACHINE エントリの中で使用できます。uux プログラムは、リモート実行要求を生成し、それらの要求をリモートコンピュータに転送するためにキューに入れます。ファイルとコマンドはターゲットコンピュータに送られて、リモート実行されます。MACHINE エントリは、ローカルシステムが呼び出しを行う場合に限り適用されるという規則がありますが、このオプションは例外です。

COMMANDS は LOGNAME エントリの中では使えないという点に注意してください。MACHINE エントリの中の COMMANDS は、ローカルシステムがリモートシステムを呼び出すのか、リモートシステムがローカルシステムを呼び出すのかに関係なく、コマンド権限を定義します。

リモートコンピュータがローカルコンピュータ上で実行できるデフォルトのコマンドは、文字列 COMMANDS=rmail となります。MACHINE エントリの中で COMMANDS=rmail 文字列を使用した場合は、デフォルトのコマンドは無効化されます。たとえば次のようなエントリを指定したとします。

```
MACHINE=owl:raven:hawk:dove COMMANDS=rmail:rnews:lp
```

これは、COMMANDS のデフォルトを無効にして、owl、raven、hawk、dove という名前の各コンピュータが、rmail、rnews、および lp をローカルコンピュータで実行できるようにします。

上記で指定した名前に加えて、コマンドのフルパス名も指定できます。たとえば次のように入力します。

```
COMMANDS=rmail:/usr/local/rnews:/usr/local/lp
```

これは、rmail コマンドがデフォルトの検索パスを使用することを指定しています。UUCP のデフォルトの検索パスは、/bin と /usr/bin です。リモートコンピュータが、実行するコマンドとして rnews または /usr/local/rnews を指定した場合は、デフォルトのパスに関係なく /usr/local/rnews が実行されます。同様に、実行される lp コマンドは /usr/local/lp です。

リストに ALL という値を含めると、エントリに指定されたリモートコンピュータから、すべてのコマンドが実行できます。この値を使用した場合は、リモートコンピュータにローカルマシンへのフルアクセスを与えることになります。

注意 - これは、通常のユーザーが持っているよりもはるかに多くのアクセス権を与えることになります。この値を使用するのは、両方のマシンが同じサイトにあり、緊密に接続されていて、ユーザーが信頼できる場合に限定するようにしてください。

次の文字列を指定したとします。

```
COMMANDS=/usr/local/rnews:ALL:/usr/local/lp
```

これは次の 2 点を示しています。

- ALL の値は文字列の中のどこでも使用できる
- 要求された rnews、lp コマンドにフルパス名が指定されていない場合は、デフォルトではなく、それぞれについて指定されているパス名が使用される

COMMANDS オプションで cat や uucp などのように、潜在的な危険性のあるコマンドを指定するときは、VALIDATE オプションを使用するようにしてください。UUCP リモート実行デーモン (uuxqt) により実行する場合、ファイルを読み書きをするコマンドは、どれもローカルセキュリティにとって危険性のあるものとなります。

UUCP VALIDATE オプション

VALIDATE コマンドは、マシンのセキュリティにとって危険性があると考えられるコマンドを指定するときに、COMMANDS オプションといっしょに使用します

(VALIDATE は、コマンドアクセスを開放する方法としては ALL より安全ですが、COMMANDS オプションのセキュリティのレベルを補強するだけのものです)。

VALIDATE は、呼び出し側マシンのホスト名と、そのマシンが使用しているログイン名とを相互にチェックするものであり、呼び出し側の識別情報について、ある程度の検証機能を備えています。次のような文字列を指定したとします。

```
LOGNAME=Uwidget VALIDATE=widget:gadget
```

この例では、widget または gadget 以外のマシンが Uwidget としてログインしようとすると、接続は拒否されます。VALIDATE オプションを使用する場合、権限が与えられたコンピュータは UUCP トランザクション用に固有のログインとパスワードを持っていなければなりません。この認証処理では、このエントリに対応するログインとパスワードを保護することが重要な条件の 1 つです。部外者がこの情報を入手してしまうと、VALIDATE オプションはセキュリティに関する役割をまったく果たさなくなります。

UUCP トランザクションについて、特権を持つログインとパスワードをどのリモートコンピュータに付与するかについては、十分に検討する必要があります。ファイルアクセスとリモート実行の権限をリモートコンピュータに与えるということは、そのリモートコンピュータのすべてのユーザーに対して、ローカルコンピュータに対する通常のログインとパスワードを与えるのと同じことです。したがって、リモートコンピュータに信頼の置けないユーザーがいると判断した場合は、そのコンピュータには特権的なログインとパスワードは付与しないようにしてください。

次のような LOGNAME エントリを指定したとします。

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

この例では、リモートコンピュータが eagle、owl、および hawk のどれかとしてローカルコンピュータにログインする場合は、そのコンピュータはログイン uucpfriend を使用する必要があります。部外者が uucpfriend を入手したとすれば、簡単に偽装することができます。

それでは、MACHINE エントリの中でだけ使用される COMMANDS オプションに対して、このオプションはどのような効果を持つのでしょうか。 このオプションは、MACHINE エントリ (および COMMANDS オプション) を、特権ログインに対応する LOGNAME エントリにリンクします。このリンクが必要なのは、リモートコンピュータがログインしている時点では、実行デーモンはまだ動作していないためです。事実、このデーモンは、どのコンピュータが実行要求を送ったのかを認識しない非同期プロセスです。ここで問題になるのが、実行ファイルがどこから送られてきたのかを、ローカルコンピュータがどのようにして知るかという点です。

各リモートコンピュータは、ローカルマシン上にそれぞれ専用スプールディレクトリを持っています。これらのスプールディレクトリの書き込み権限は、UUCP プログラムだけに与えられています。リモートコンピュータからの実行ファイルは、ローカルコンピュータに転送された後に、このスプールディレクトリに入れられます。uuxqt デーモンが動作するときには、スプールディレクトリ名を使用して、Permissions ファイルから MACHINE エントリを見つけ、COMMANDS リストを取得します。Permissions ファイル内に該当するコンピュータ名が見つからない場合は、デフォルトのリストが使用されます。

次の例は、MACHINE エントリと LOGNAME エントリの関係を示しています。

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=rmail:/usr/local/rnews \
READ=/ WRITE=/
LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

COMMANDS オプションの値は、リモートユーザーが、rmail と /usr/local/rnews を実行できることを示しています。

最初のエントリでは、リストされているコンピュータのどれかを呼び出したい場合に、実際には eagle、owl、hawk のどれかを呼び出すということを理解しておく必要があります。したがって、eagle、owl、および hawk のスプールディレクトリに置かれるファイルはすべて、それらのコンピュータのどれかが投入したことになります。あるリモートコンピュータがログインし、この 3 つのコンピュータのどれかであることを主張した場合、その実行ファイルもこの特権スプールディレクトリに入れられます。したがって、ローカルコンピュータでは、そのコンピュータが特権ログイン uucpz を持っていることを確認する必要があります。

UUCP OTHER 用の MACHINE エントリ

特定の MACHINE エントリに記述されていないリモートマシンについて、異なるオプション値を指定したい場合があります。これが必要になるのは、多数のコンピュータがローカルホストを呼び出し、コマンドセットがそのたびに異なるような場合です。次の例に示すように、このようなエントリでは、コンピュータ名として OTHER という名前を使用します。

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/local/Photo:/usr/local/xp
```

他の MACHINE エントリに記述されていないコンピュータについても、MACHINE エントリに使用できるすべてのオプションを設定できます。

UUCP の MACHINE エントリと LOGNAME エントリの結合

MACHINE エントリと LOGNAME エントリを結合して、同じ共通オプションを持つ単一のエントリにすることができます。たとえば、次の 2 つのエントリがあるとします。

```
MACHINE=eagle:owl:hawk REQUEST=yes \
READ=/ WRITE=/
```

```
LOGNAME=uupz REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

これらは、同じ REQUEST、READ、および WRITE オプションを共有しています。この 2 つを結合すると次のようになります。

```
MACHINE=eagle:owl:hawk REQUEST=yes \
logname=uucpz SENDFILES-yes \
READ=/ WRITE=/
```

MACHINE エントリと LOGNAME エントリを結合することによって、Permissions ファイルは、効率的で管理しやすくなります。

UUCP の転送

一連のマシンを介してファイルを送信するときは、リレー (中継) マシンの COMMANDS オプションの中に uucp コマンドが含まれていなければなりません。たとえば次のコマンドを入力したとします。

```
% uucp sample.txt oak\!willow\!pine\!/usr/spool/uucppublic
```

この転送操作が正常に機能するためには、マシン willow がマシン oak に対して uucp プログラムの実行を許可し、oak がローカルマシンに同じことを許可している必要があります。最終宛先マシンである pine は、転送動作を行わないため、uucp コマンドを許可する必要はありません。通常、マシンはこのように設定されていません。

UUCP /etc/uucp/Poll ファイル

/etc/uucp/Poll ファイルには、リモートコンピュータをポーリングするための情報が入っています。Poll ファイル内の各エントリには、呼び出すリモートコンピュータの名前と、それに続くタブ文字またはスペース、最後にそのコンピュータを呼び出す時刻が入ります。Poll ファイル内のエントリの形式は次のとおりです。

sys-name hour ...

たとえば次のようなエントリを指定したとします。

```
eagle 0 4 8 12 16 20
```

これは、コンピュータ eagle を 4 時間おきにポーリングします。

uudemon.poll スクリプトは Poll ファイルを処理しますが、実際にポーリングを行うわけではありません。これは単にスプールディレクトリ内にポーリング作業ファイル (名前は常に C.*file*) を設定するだけです。uudemon.poll スクリプトはスケジューラを起動し、スケジューラは、スプールディレクトリ内のすべての作業ファイルを調べます。

UUCP /etc/uucp/Config ファイル

/etc/uucp/Config ファイルを使用すると、いくつかのパラメータを手動で上書きできます。Config ファイルの各エントリの形式は次のとおりです。

parameter=value

構成可能な全パラメータ名のリストについては、システムに付属している Config ファイルを参照してください。

次の Config エントリは、デフォルトのプロトコル順序を Gge に設定し、G プロトコルのデフォルト値を、ウィンドウ数 7、パケットサイズ 512 バイトに変更します。

```
Protocol=G(7,512)ge
```

UUCP /etc/uucp/Grades ファイル

/etc/uucp/Grades ファイルには、リモートコンピュータへのジョブをキューに入れるときに指定できるジョブグレードが入っています。また、個々のジョブグレードに関するアクセス権も含まれています。このファイルのエントリは、ユーザーがジョブをキューに入れるときに使用する、管理者が定義したジョブグレードの定義を表しています。

Grades ファイルのエントリの形式は次のとおりです。

User-job-grade System-job-grade Job-size Permit-type ID-list

各エントリには、スペースで区切ったいくつかのフィールドがあります。エントリの最後のフィールドは、同じくスペースで区切ったいくつかのサブフィールドから構成されます。1 つのエントリが複数の物理行にわたる場合は、バックスラッシュを使用して、エントリを次の行に継続させることができます。コメント行はポンド記号 (#) で始まり、その行の全体を占めます。空の行は常に無視されます。

UUCP User-job-grade フィールド

このフィールドには、管理者が 64 文字以内で定義したユーザージョブのグレード名が入ります。

UUCP System-job-grade フィールド

このフィールドには、*User-job-grade* が対応付けされる 1 文字のジョブグレードが入ります。有効な文字は A ～ Z、a ～ z で、最も優先順位が高いのは A、最も優先順位が低いのは z です。

ユーザージョブグレードとシステムジョブグレードの関係

ユーザージョブグレードは複数のシステムジョブグレードに割り当てることができます。ここで重要なのは、Grades ファイルは、ユーザージョブグレードのエントリ

を見つけるために先頭から検索されるという点です。したがって、最大ジョブサイズの制限値に応じて、複数のシステムジョブグレードのエントリが列挙されます。

ユーザージョブグレードの最大数には制限はありませんが、システムジョブグレードの許容最大数は 52 です。その理由は、1 つの *System-job-grade* には複数の *User-job-grade* を対応付けできるが、個々の *User-job-grade* はファイル内でそれぞれ単独の行でなければならないという点にあります。次に例を示します。

```
mail N Any User Any netnews N Any User Any
```

`Grades` ファイル内でこのような構成をした場合、2 つの *User-job-grade* が同じ *System-job-grade* を共有します。ジョブグレードに関するアクセス権は、*System-job-grade* ではなく *User-job-grade* に割り当てられるものなので、2 つの *User-job-grade* は同じ *System-job-grade* を共有しながら、それぞれ異なるアクセス権のセットを持つことができます。

デフォルトグレード

デフォルトのユーザージョブグレードとして、システムジョブグレードを割り当てることができます。そのためには、`Grades` ファイルの *User-job-grade* フィールドのユーザージョブグレードとしてキーワード default を使用し、そのデフォルトに割り当てるシステムジョブグレードを指定します。Restriction フィールドと ID フィールドは `Any` と定義して、どのようなユーザー、どのようなサイズのジョブでも、このグレードでキューに入れることができるようにします。次に例を示します。

```
default a Any User Any
```

デフォルトのユーザージョブグレードを定義しなかった場合は、組み込まれているデフォルトグレードである `z` が使用されます。Restriction フィールドのデフォルトは `Any` なので、デフォルトグレードのエントリが複数存在していても検査されません。

UUCP Job-size フィールド

このフィールドは、キューに入れることのできる最大ジョブサイズを指定します。*Job-size* はバイト数で表され、表 27–8 に示すオプションを使用できます。

表 27–8 Job-size フィールド

nnnn	このジョブグレードの最大ジョブサイズを指定する整数
*n*K	K バイト数を表す 10 進数 (K はキロバイトの略号)
*n*M	M バイト数を表す 10 進数 (M はメガバイトの略号)
Any	最大ジョブサイズが指定されないことを指定するキーワード

次に例をいくつか示します。

- 5000 は 5000 バイトを表す
- 10K は 10K バイトを表す
- 2M は 2M バイトを表す

UUCP Permit-type フィールド

このフィールドには、ID リストをどのように解釈するかを指示するキーワードを指定します。表 27–9 に、キーワードとそれぞれの意味を示します。

表 27–9 Permit-type フィールド

キーワード	**ID** リストの内容
User	このジョブグレードの使用を許可されているユーザーのログイン名
Non-user	このジョブグレードの使用を許可されていないユーザーのログイン名
Group	このジョブグレードの使用を許可されているメンバーのグループ名
Non-group	このジョブグレードの使用を許可されていないメンバーのグループ名

UUCP ID-list フィールド

このフィールドには、このジョブグレードへキューを入れることが許可、禁止されるログイン名またはグループ名のリストが入ります。名前のリストはそれぞれスペースで区切り、改行文字で終了します。このジョブグレードへキューを入れることを誰にでも許可する場合は、キーワード Any を使用します。

その他の UUCP 構成ファイル

この節では、UUCP の機能に影響を与えるファイルのうち、比較的変更頻度の低い 3 つのファイルについて説明します。

UUCP /etc/uucp/Devconfig ファイル

/etc/uucp/Devconfig ファイルを使用するとサービス別、つまり uucp 用と cu 用とに分けて、デバイスを構成できます。Devconfig のエントリは、個々のデバイスで使用される STREAMS モジュールを定義します。エントリの形式は次のとおりです。

```
service=x device=y push=z[:z...]
```

x は、cu か uucico、またはその両方をコロンで区切ったものです。*y* はネットワークの名前で、これは Devices ファイルのエントリに一致していなければなりません。*z* には、STREAMS モジュールの名前を、Stream にプッシュする順序で指定します。cu サービスと uucp サービスについて、それぞれ異なるモジュールとデバイスを定義できます。

次のエントリは STARLAN ネットワーク用のもので、このファイル内で最もよく使用されるものです。

```
service=cu       device=STARLAN     push=ntty:tirdwr
service=uucico   device=STARLAN     push=ntty:tirdwr
```

この例では、まず ntty、次に tirdwr がプッシュされます。

UUCP /etc/uucp/Limits ファイル

/etc/uucp/Limits ファイルは、uucp ネットワーク処理で同時に実行できる uucico、uuxqt、および uusched の最大数を制御します。ほとんどの場合は、デフォルトの値が最適であり、変更の必要はありません。変更したい場合は、任意のテキストエディタを使用してください。

Limits ファイルの形式は次のとおりです。

```
service=x max=y:
```

x は uucico、uuxqt、uusched のどれかで、*y* はそのサービスについての制限値です。フィールドは、小文字を使用して任意の順序で入力できます。

次に示すのは、Limits ファイルの中で一般的に使用される内容です。

```
service=uucico max=5
service=uuxqt max=5
service=uusched max=2
```

この例は、5 つの uucico、5 つの uuxqt、2 つの uusched をマシンで実行できることを示しています。

UUCP remote.unknown ファイル

通信機能の使用に影響を与えるファイルとして、もう 1 つ remote.unknown ファイルがあります。このファイルは、どの Systems ファイルにも含まれていないマシンが通信を開始したときに実行されるバイナリプログラムです。このプログラムはその通信をログに記録し、接続を切断します。

注意 - remote.unknown ファイルのアクセス権を変更して、このプログラムが実行できないようにすると、ローカルシステムはどのシステムからの接続も受け入れることになります。

このプログラムが実行されるのは、どの Systems ファイルにも含まれていないマシンが対話を開始した場合です。このプログラムは、その対話を記録し、接続を失敗させます。このファイルのアクセス権を変更して実行できないようにしてしまうと (chmod 000 remote.unknown)、ローカルシステムはすべての通信要求を受け

入れることになります。妥当な理由がない限り、この変更は行わないようにしてください。

UUCP の管理ファイル

次に、UUCP 管理ファイルについて説明します。これらのファイルは、デバイスのロック、一時データの保管、リモート転送や実行に関する情報の保存などのために、スプールディレクトリ内に作成されます。

- 一時データファイル (TM) – これらのデータファイルは、他のコンピュータからファイルを受け取るときに、UUCP プロセスによりスプールディレクトリ /var/spool/uucp/*x* の下に作成されます。ディレクトリ *x* は、ファイルを送信しているリモートコンピュータと同じ名前です。一時データファイル名の形式は次のとおりです。

  ```
  TM.pid.ddd
  ```

  pid はプロセス ID、*ddd* は 0 から始まる 3 桁のシーケンス番号です。

  ファイルの全体が受信されると、TM.*pid.ddd* ファイルは、伝送を発生させた C.*sysnxxx* ファイル (以下で説明します) の中で指定されているパス名に移されます。処理が異常終了した場合は、TM.*pid.ddd* ファイルは *x* ディレクトリ内に残されます。このファイルは、uucleanup を使用することにより自動的に削除されます。

- ロックファイル (LCK) – ロックファイルは、使用中の各デバイスごとに、/var/spool/locks ディレクトリ内に作成されます。ロックファイルは、対話の重複、複数の試行による同じ呼び出しデバイスの使用が発生するのを防ぎます。表 27–10 に、UUCP ロックファイルの種類を示します。

表 27–10 UUCP ロックファイル

ファイル名	説明
LCK..*sys*	*sys* はファイルを使用しているコンピュータ名を表す
LCK.*dev*	*dev* はファイルを使用しているデバイス名を表す
LCK.*LOG*	*LOG* はロックされている UUCP ログファイルを表す

通信リンクが予定外のときに切断された場合 (通常コンピュータがクラッシュしたとき)、これらのファイルがスプールディレクトリ内に残ることがあります。親プロセスが有効でなくなった後は、ロックファイルは無視 (削除) されます。ロックファイルには、ロックを引き起こしたプロセスのプロセス ID が入っています。

- 作業ファイル (C.) – 作業ファイルは、リモートコンピュータに送る作業 (ファイル転送またはリモートコマンド実行) がキューに入れられたときに、スプールディレクトリ内に作成されます。作業ファイル名の形式は次のとおりです。

  ```
  C.sysnxxxx
  ```

  sys はリモートコンピュータ名、*n* は作業のグレード (優先順位) を表す ASCII 文字、*xxxx* は、UUCP が割り当てる 4 桁のジョブシーケンス番号です。作業ファイルには次の情報が含まれています。

  - 送信または要求するファイルのフルパス名
  - 宛先、ユーザー名、またはファイル名を表すフルパス名
  - ユーザーのログイン名
  - オプションのリスト
  - スプールディレクトリ内の関連データファイルの名前。uucp –C オプションか uuto –p オプションが指定されている場合は、ダミー名 (D.0) が使用される
  - ソースファイルのモードビット
  - 転送完了の通知を受け取るリモートユーザーのログイン名

- データファイル (D.) – コマンド行でスプールディレクトリへのソースファイルのコピーを指定すると、データファイルが作成されます。データファイル名の形式は次のとおりです。

D.*systmxxxxyyy* – *systm* はリモートコンピュータ名の最初の 5 文字で、*xxxx* は uucp が割り当てる 4 桁のジョブシーケンス番号です。4 桁のジョブシーケンス番号の後に続く *yyy* はサブシーケンス番号で、これは、1 つの作業 (C.) ファイルについて複数の D. ファイルが作成された場合に使用されます。

- X. *(実行ファイル)* – 実行ファイルは、リモートコマンドの実行の前にスプールディレクトリ内に作成されます。実行ファイル名の形式は次のとおりです。

```
X.sysnxxxx
```

sys はリモートコンピュータ名で、*n* は作業のグレード (優先順位) を表す文字です。*xxxx* は、UUCP が割り当てる 4 桁のシーケンス番号です。実行ファイルには次の情報が入ります。

- 要求元のログイン名とコンピュータ名
- 実行に必要なファイル名
- コマンド文字列への標準入力として使用する入力
- コマンド実行の標準出力を受け取るコンピュータとファイルの名前
- コマンド文字列
- 終了ステータスの要求のためのオプション行

UUCP のエラーメッセージ

この節には、UUCP に関連したエラーメッセージを示します。

UUCP の ASSERT エラーメッセージ

表 27–11 に ASSERT エラーメッセージを一覧にします。

表 27–11 ASSERT エラーメッセージ

エラーメッセージ	説明と処置
CAN'T OPEN	open() または fopen() が失敗した
CAN'T WRITE	write()、fwrite()、fprint()、または類似のコマンドが失敗した

表 27–11 ASSERT エラーメッセージ 続く

エラーメッセージ	説明と処置
CAN'T READ	read()、fgets()、または類似のコマンドが失敗した
CAN'T CREATE	creat() 呼び出しが失敗した
CAN'T ALLOCATE	動的割り当てが失敗した
CAN'T LOCK	LCK (ロック) ファイルを作成しようとしたが失敗した。場合によっては、これは重大なエラーとなる
CAN'T STAT	stat() 呼び出しが失敗した
CAN'T CHMOD	chmod() 呼び出しが失敗した
CAN'T LINK	link() 呼び出しが失敗した
CAN'T CHDIR	chdir() 呼び出しが失敗した
CAN'T UNLINK	unlink() 呼び出しが失敗した
WRONG ROLE	内部ロジックの問題
CAN'T MOVE TO CORRUPTDIR	不良な C. ファイルまたは X. ファイルを、/var/spool/uucp/.Corrupt ディレクトリに移動しようとしたが失敗した。このディレクトリが存在しないか、モードまたは所有者が正しくない
CAN'T CLOSE	close() または fclose() 呼び出しが失敗した
FILE EXISTS	C. ファイルまたは D. ファイルを作成しようとしたが、そのファイルがすでに存在している。この症状は、シーケンスファイルのアクセスに問題がある場合に生じる。これは通常、ソフトエラーを示す
NO uucp SERVICE NUMBER	TCP/IP 呼び出しを試みたが、/etc/services 内に UUCP に関するエントリがない
BAD UID	ユーザー ID がパスワードデータベース内にない。ネームサービス構成の検査が必要
BAD LOGIN_UID	前記と同じ

表 27–11 ASSERT エラーメッセージ 続く

エラーメッセージ	説明と処置
BAD LINE	Devices ファイル内に不良な行がある。引数が足りない行が 1 つ以上ある
SYSLST OVERFLOW	gename.c の内部テーブルがオーバーフローした。1 つのジョブが 30 を超えるシステムに接続しようとした
TOO MANY SAVED C FILES	前記と同じ
RETURN FROM fixline ioctl	失敗するはずのない ioctl(2) が失敗した。システムドライバに問題がある
BAD SPEED	Devices ファイルまたは Systems ファイルの中に不適正な回線速度がある (Class フィールドまたは Speed フィールド)
BAD OPTION	Permissions ファイルの中に不適正な行またはオプションがある。ただちに修正が必要
PKCGET READ	リモートマシンがハングアップした可能性がある。処置は不要
PKXSTART	リモートマシンが回復不可能な状態で異常終了した。通常これは無視できる
TOO MANY LOCKS	内部的な問題がある。システムのご購入先にお問い合わせください
XMV ERROR	ファイル、またはディレクトリのどこかに問題が発生している。このプロセスが実行される前に、宛先のモードがチェックされるべきであるが実行されていないなど、スプールディレクトリに問題がある可能性がある
CAN'T FORK	fork と exec を実行しようとしたが失敗した。現行ジョブは失われず、後で再試行される (uuxqt)。処置は不要

UUCP の STATUS エラーメッセージ

表 27–12 に一般的な STATUS エラーメッセージを示します。

表 27–12 UUCP の STATUS エラーメッセージ

エラーメッセージ	説明と処置
OK	状態は良好
NO DEVICES AVAILABLE	現在、この呼び出し用に使用可能なデバイスがない。該当のシステムについて Devices ファイル内に有効なデバイスがあるかどうかを確認してください。そのシステムの呼び出しに使用するデバイスが Systems ファイル内にあるかどうかを検査してください
WRONG TIME TO CALL	Systems ファイルに指定されている日時以外の時点で、システムに対する呼び出しが行われた
TALKING	会話中
LOGIN FAILED	特定のマシンのログインが失敗した。ログインまたはパスワードが正しくないか、番号が正しくないか、低速のマシンであるか、Dialer-Token-Pairs スクリプトによる処理が失敗した
CONVERSATION FAILED	起動に成功した後で対話が失敗した。一方の側がダウンしたか、プログラムが異常終了したか、回線 (リンク) が切断されたことが考えられる
DIAL FAILED	リモートマシンがまったく応答しない。ダイヤラが不良であるか、電話番号が正しくない可能性がある
BAD LOGIN/MACHINE COMBINATION	あるマシンが、Permissions ファイルの条件を満たしていないログインとマシン名を使用して、ローカルマシンを呼び出そうとした。偽装の疑いがある
DEVICE LOCKED	使用しようとしている呼び出しデバイスは、現在ロックされ、他のプロセスに使用されている
ASSERT ERROR	ASSERT エラーが発生した。/var/uucp/.Admin/errors ファイルにエラーメッセージが入っているかどうかを検査し、589ページの「UUCP のエラーメッセージ」を参照
SYSTEM NOT IN Systems FILE	システムが Systems ファイルの中に記述されていない
CAN'T ACCESS DEVICE	アクセスしようとしたデバイスが存在しないか、またはモードが正しくない。Systems ファイルと Devices ファイルの中の該当のエントリを検査する
DEVICE FAILED	デバイスがオープンできない

表 27–12　UUCP の STATUS エラーメッセージ　続く

エラーメッセージ	説明と処置
WRONG MACHINE NAME	呼び出されたマシンは、予期したのとは異なる名前である
CALLBACK REQUIRED	呼び出されたマシンは、そのマシンがローカルマシンをコールバックする必要があることを示している
REMOTE HAS A LCK FILE FOR ME	リモートマシンは、ローカルマシンに関連する LCK ファイルを持っている。そのリモートマシンがローカルマシンを呼び出そうとしている可能性がある。そのマシンの UUCP のバージョンが古い場合は、プロセスがローカルマシンに接続しようとして失敗し、LCK ファイルがそのまま残されたことが考えられる。UUCP のバージョンが新しく、そのマシンがローカルマシンと通信していない場合は、LCK を持っているプロセスはハングする
REMOTE DOES NOT KNOW ME	リモートマシンの Systems ファイルの中に、ローカルマシンのノード名がない
REMOTE REJECT AFTER LOGIN	ローカルマシンがログインのために使用したログインが、リモートマシンが予期している内容に一致していない
REMOTE REJECT, UNKNOWN MESSAGE	理由は不明だが、リモートマシンがローカルマシンとの通信を拒否した。リモートマシンが標準バージョンの UUCP を使用していない可能性がある
STARTUP FAILED	ログインは成功したが、初期ハンドシェークに失敗した
CALLER SCRIPT FAILED	通常、これは DIAL FAILED と同じ。しかしこれが頻発する場合は、Dialers ファイル内の呼び出し側スクリプトに原因があることが考えられる (Uutry を使用して検査する)

UUCP の数値エラーメッセージ

表 27–13 に、/usr/include/sysexits.h ファイルにより生成されるエラー状態メッセージの終了コード番号を示します。これらのすべてが現在 uucp で使用されているわけではありません。

表 27–13 番号による UUCP のエラーメッセージ

メッセージ番号	内容	説明
64	Base Value for Error Messages	エラーメッセージはこの番号から始まる
64	Command–Line Usage Error	コマンドの使い方に誤りがある。たとえば、引数の数が正しくない、誤ったフラグ、誤った構文など
65	Data Format Error	入力データになんらかの誤りがある。これはユーザーデータだけに使用されるもので、システムファイルには使用されない
66	Cannot Open Input	入力ファイル (システムファイルでない) が存在しないか、または読み取れない。これには、メーラーに対する「No message」のようなエラーも含まれる
67	Address Unknown	指定されたユーザーが存在しない。これは、メールアドレスやリモートログインに使用される
68	Host Name Unknown	ホストが存在しない。これは、メールアドレスやネットワーク要求に使用される
69	Service Unavailable	サービスが使用不可。これは、サポートプログラムまたはファイルが存在しない場合に起こることがある。このメッセージは、何かが正常に働かずその理由が分からない場合の包括的なメッセージでもある
70	Internal Software Error	内部ソフトウェアエラーが検出された。これは、可能な場合は、オペレーティングシステム関係以外のエラーに限定される
71	System Error	オペレーティングシステムエラーが検出された。これは、「フォーク不可」や「パイプ作成不可」などのような状態を示す。たとえば、`getuid` が `passwd` ファイル内に存在しないユーザーを戻した場合などが含まれる
72	Critical OS File Missing	`/etc/passwd` や `/var/admin/utmpx` などのシステムファイルのどれかが存在しないか、開くことができない。あるいは、構文エラーなどがある
73	Can't Create Output File	ユーザーが指定した出力ファイルが作成できない
74	Input/Output Error	あるファイルについて入出力を行なっているときにエラーが起こった
75	Temporary Failure. User is invited to retry	一時的な障害。実際のエラーではない何かを示す。たとえば `sendmail` では、これは、メーラーが接続を確立できなかったため、後で要求を再試行する必要があることなどを意味する

表 27–13 番号による UUCP のエラーメッセージ 続く

メッセージ番号	内容	説明
76	Remote Error in Protocol	プロトコルの交換中に、リモートシステムが「使用不可」を示す何かを戻した
77	Permission Denied	この操作を行うための適正なアクセス権がユーザーにない。これはファイルシステムの問題を示すものではなく (その場合は NOINPUT や CANTCREAT などが使用される)、より高いレベルのアクセス権が必要であることを意味する。たとえば、kre は、メールを送ることのできる学生を制限するために、このメッセージを使用する
78	Configuration Error	システムの構成にエラーがある
79	Entry Not Found	エントリが見つからない
79	Maximum Listed Value	エラーメッセージの最大番号

第 28 章

リモートファイルシステムへのアクセスについてのトピック

第 29 章	NFS サービスの概要
第 30 章	NFS サービスの設定と障害回避方法 (トラブルシューティング)
第 31 章	NFS サービスの背景情報について

第 29 章

Solaris NFS の環境

この章では、NFS 環境の概要について説明します。具体的には、ネットワークの簡単な概要、NFS サービス、NFS システムの把握に必要な概念を説明します。

NFS サーバーとクライアント

クライアントとサーバーという用語は、コンピュータがファイルシステムを共有するときの役割を示すものです。ファイルシステムがあるコンピュータのディスク上に存在し、そのコンピュータがこのファイルシステムをネットワーク上の他のコンピュータから使用できるようにしている場合、そのコンピュータをサーバーと呼びます。そのファイルシステムにアクセスしているコンピュータをクライアントと呼びます。 NFS を使用することによって、どのコンピュータからも他のコンピュータのファイルシステムにアクセスでき、それと同時に自分のファイルシステムへのアクセスも可能となります。ネットワーク上では 1 台のコンピュータがクライアントかサーバー、またはその両方の役割として動作することができます。

クライアントは、サーバーの共有ファイルシステムをマウントすることによってサーバーのファイルにアクセスします。クライアントがリモートファイルシステムをマウントしたとき、ファイルシステムがコピーされるのではありません。マウン

ト処理では一連のリモートプロシージャコールによって、クライアントからサーバーのディスク上にあるファイルシステムに透過的にアクセスできるようになります。マウントはローカルマウントのように行われるので、ユーザーはファイルシステムがローカルにあるのと同じようにコマンドを入力します。

サーバーのファイルシステムは、NFS オペレーションによって共有すると、クライアントからアクセスできるようになります。NFS ファイルシステムは、autofs を使用すると自動的にマウントできます。

NFS ファイルシステム

NFS サービスで共有できるオブジェクトは、ファイル階層の全体、またはその一部です。ファイルを 1 つだけ共有することもできます。すでに共有しているものと重複するファイル階層構造は共有できません。モデムやプリンタなどの周辺機器も共有できません。

多くの UNIX システム環境で共有されるファイル階層構造は、1 つのファイルシステム、またはその一部です。しかし NFS サポートは複数のオペレーティングシステムにまたがって動作しますが、ファイルシステムという考え方は UNIX 以外の環境では通用しません。したがってこのマニュアルでファイルシステムという語を使用する場合、NFS 環境でマウントし共有した、ファイルまたはファイル階層構造を指すことにします。

NFS 環境

NFS サービスとは、アーキテクチャが異なり、別のオペレーティングシステムで動作しているコンピュータが、ネットワークを通じてファイルシステムを共有できるようにするサービスのことです。NFS サポートは、MS-DOS から VMS オペレーティングシステムまで多くのプラットフォームに実装されています。

NFS 環境は、異なるオペレーティングシステムで実現できます。アーキテクチャの仕様を定義するのではなく、ファイルシステムの抽象モデルを定義しているためです。それぞれのオペレーティングシステムでは、ファイルシステムセマンティクスに NFS 抽象モデルを適用します。これにより、書き込みや読み出しのようなファイルシステムオペレーションが、ローカルファイルにアクセスするように機能することになります。

NFS サービスの利点を以下に挙げます。

- 複数のコンピュータで同一のファイルを使用するため、ネットワーク上の誰もが同じデータにアクセスできる
- 各ユーザーアプリケーションがローカルのディスク空間を占めるのではなく、複数のコンピュータでアプリケーションを共有するため、記憶領域を有効利用できる
- すべてのユーザーが同一セットのファイルを読み出すので、データの整合性と信頼性が向上する
- ファイルシステムをユーザーに透過的な形でマウントできる
- リモートファイルに透過的にアクセスできる
- 様々な環境をサポートする
- システム管理の手間を省ける

NFS サービスを使用すると、ファイルシステムの実際の場所をユーザーとは無関係に決めることができます。ユーザーは場所を気にすることなく、すべての適切なファイルにアクセスできるということです。NFS サービスでは、共通して使用するファイルのコピーをすべてのシステムに置くのではなく、コピーを 1 つのコンピュータのディスクに置き、他のシステムからネットワークを通じてアクセスできるようにします。NFS オペレーションでは、リモートファイルとローカルファイルの区別がありません。

NFS バージョン 2

バージョン 2 は、一般に広く使用された初めての NFS プロトコルです。バージョン 2 は、引き続き広範囲のプラットフォームで使用できます。Solaris 2.5 以前のリリースの SunOS では、NFS プロトコルのバージョン 2 が使用できます。

NFS バージョン 3

NFS バージョン 3 のプロトコルは、Solaris 2.5 に新機能を追加したものです。相互運用性と性能を向上させるために、いくつかの変更が行われました。これらをすべて有効に利用するには、NFS サーバーとクライアントの両方で、バージョン 3 プロトコルを使用する必要があります。

バージョン 3 では、サーバーで非同期の書き込みが可能になります。サーバーがクライアントの書き込み要求をメモリーに保存するので、効率が向上しました。ク

ライアントは、サーバーが変更内容をディスクに反映させるのを待つ必要がないため、応答時間が短縮されます。サーバーは要求をバッチ処理することもできるので、サーバー上の応答時間も短縮されました。

NFS バージョン 3 では、どの操作でもローカルキャッシュに保存されているファイル属性が返されます。キャッシュの更新頻度が増えたため、ローカルキャッシュのデータを更新する操作を独立して行う必要性が少なくなります。したがってサーバーに対する RPC コールの回数が減少し、性能が向上します。

ファイルアクセス権の確認処理も改善されました。バージョン 2 では、ユーザーがアクセス権を持っていないリモートファイルをコピーしようとすると、「書き込みエラー」や「読み出しエラー」というメッセージが出力されました。バージョン 3 では、ファイルを開く前に権利がチェックされるので、「オープンエラー」というメッセージが出力されます。

NFS バージョン 3 では、8K バイトの転送サイズ制限が解除されました。クライアントとサーバーは、バージョン 2 で課せられていた 8K バイト制限を受けず、サポートできる転送サイズならばどのようなものでも処理します。Solaris 2.5 では、転送サイズが 32K バイトにデフォルトで設定されています。

NFS ACL サポート

Solaris 2.5 には、アクセス制御リスト (ACL) サポートが追加されました。ACL では、ファイルアクセス権を通常の UNIX よりも正確に設定します。この追加機能では効率は改善されませんが、ファイルへのアクセスがより厳密に制限されるので、セキュリティが向上します。

NFS の TCP への依存

Solaris 2.5 では、NFS プロトコルのデフォルトのトランスポートプロトコルが TCP に変わりました。このため、低速のネットワークおよび広域ネットワークにおける性能が改善されます。TCP には、トラフィック抑制機能とエラー回復機能があります。TCP を利用した NFS は、バージョン 2 でもバージョン 3 でも動作します。2.5 より前のバージョンでは、NFS のデフォルトプロトコルは UDP (User Datagram Protocol) でした。

ネットワークロックマネージャ

Solaris 2.5 には、ネットワークロックマネージャの改良版も含まれています。このため NFS ファイルに対して UNIX のレコードロックと PC のファイル共有が使用できます。NFS ファイルに対するロック機構の信頼性が向上したため、ロックを使用する `ksh` や `mail` などのコマンドがハングする可能性が少なくなります。

NFS 大型ファイルのサポート

Solaris 2.6 の NFS バージョン 3 プロトコルでは、2G バイトを超えるサイズのファイルも正しく処理できます。NFS バージョン 2 プロトコル、および Solaris 2.5 に実装されているバージョン 3 プロトコルでは 2G バイトを超えるサイズのファイルは処理できませんでした。

NFS クライアントのフェイルオーバー機能

Solaris 2.6 では、読み取り専用ファイルシステムの動的フェイルオーバー機能が追加されました。これによって、マニュアルページ、AnswerBook、共有バイナリなどのあらかじめ複製されている読み取り専用リソースを高度に利用できます。フェイルオーバー機能は、ファイルシステムがマウントされた後ならばいつでも実行可能です。手動マウントでは、今までのリリースのオートマウンタのように複数の複製をリストできるようになりました。オートマウンタは、フェイルオーバーの際にファイルシステムが再マウントされるまで待つ必要がなくなったこと以外は変更されていません。

Kerberos による NFS 環境のサポート

Solaris 2.0 では、Kerberos V4 クライアントがサポートされていました。Solaris 2.6 では、`mount` コマンドと `share` コマンドは、Kerberos V5 認証を使用した NFS マウントをサポートするようになりました。また、`share` コマンドはクライアントごとに異なる複数の認証方法を指定できるようになりました。

WebNFS のサポート

Solaris 2.6 には、NFS プロトコルの拡張機能を使用することによってインターネット上のファイルシステムにファイアウォール経由でアクセスできるようにする機能

もあります。この WebNFS™ プロトコルを使用してインターネットにアクセスする利点の 1 つは、NFS バージョン 3 とバージョン 2 プロトコルの拡張機能としてサービスが構築されるということです。今後、この新しいファイルシステムのアクセスプロトコルを使用したアプリケーションがいくつも作成される予定です。また NFS サーバーでは、負荷が大きい状態のときに HTTP (HyperText Transfer Protocol) から Web サーバーへのアクセスよりも高いスループットを確保できます。そのため、ファイルを取得するための時間が短縮されます。さらに、WebNFS ではそうしたファイルを共有しても匿名 `ftp` サイトを管理するオーバーヘッドが生じません。

RPCSEC_GSS セキュリティ方式

Solaris 7 では、新しいセキュリティ方式である RPCSEC_GSS がサポートされています。この方式では、標準的な GSS-API インタフェースを使用して、認証、一貫性、機密性を実現し、複数のセキュリティ機構をサポートしています。現在、これらの新しいセキュリティ方法を使用する機構は、Solaris では、クライアント側だけに組み込まれています。

Solaris 7 の NFS に対する拡張機能

Solaris 7 では、`mount` コマンドと `automountd` コマンドが拡張され、マウント要求で MOUNT プロトコルの代わりに公開ファイルハンドルも使用できるようになりました。これは、WebNFS サービスで使用されているのと同じアクセス方法です。公開ファイルハンドルを使用すると、ファイアウォールを越えたマウントが可能です。さらに、サーバーとクライアント間のトランザクションが少なくて済むため、マウントにかかる時間が短縮されます。

この機能拡張で、標準のパス名の代わりに NFS URL を使用することもできるようになりました。また、`mount` コマンドとオートマウンタのマップに `–public` オプションを指定すると、必ず公開ファイルハンドルを使用するようになります。

WebNFS サービスのセキュリティネゴシエーション

新しいプロトコルが 1 つ追加されて、WebNFS クライアントで NFS サーバーとセキュリティメカニズムのネゴシエーションが行なえるようになりました。このプロトコルの追加により、WebNFS サービスの使用時に、セキュリティ保護されたトランザクションが使用できます。

NFS サーバーログ

NFS サーバーログにより、NFS サーバーはファイルシステム上で実行されるファイル操作の記録を提供することができます。このログには、アクセスされた対象、時間、アクセスした人を追跡するための情報が含まれています。一連の構成オプションを使用して、これらの情報を含むログの場所を指定することができます。また、これらのオプションを使用して、ログにとる操作を選択することもできます。この機能は、NFS クライアントや WebNFS クライアントで anonymous FTP を利用するサイトで特に便利です。

autofs について

NFS サービスを使用して共有されるファイルシステムは、「自動マウント」と呼ばれる方法によってマウントできます。クライアント側のサービスである autofs は、自動マウントを実現するファイルシステム構造です。autofs のファイルシステムは、automount で作成されます。automount は、システムを起動すると自動的に実行されます。automountd という常駐型の automount デーモンが、必要に応じてリモートディレクトリのマウントとアンマウントを行います。

automountd を実行しているクライアントコンピュータ上のユーザーがリモートのファイルまたはディレクトリにアクセスしようとすると、そのファイルまたはディレクトリが所属するファイルシステムがこのデーモンによってマウントされます。このリモートファイルシステムは、必要な間はマウントされたままです。リモートファイルシステムが一定時間アクセスされないと、自動的にアンマウントされます。

ブート時にはマウントする必要はなく、ユーザーはディレクトリをマウントするためにスーパーユーザーのパスワードを知る必要はありません。ユーザーが mount と umount コマンドを使用する必要もありません。autofs は、ユーザーの介入なしに、必要に応じてファイルシステムをマウントまたはアンマウントします。

automountd によって一部のファイル階層をマウントするということは、mount によって他のファイル階層をマウントしないということではありません。ディスクレスコンピュータは、mount と /etc/vfstab ファイルを使用して / (ルート)、/usr、および /usr/kvm をマウントしなければなりません。

autofs サービスについては、627ページの「autofs 管理作業の概要」と 712ページの「autofs のしくみ」で詳しく説明します。

autofs の特徴

autofs は、ローカルの名前空間に指定したファイルシステムで動作します。この情報は、NIS、NIS+、およびローカルファイルに保存されます。

Solaris 2.6 には、完全にマルチスレッド化された automountd が含まれています。この拡張によって autofs はさらに信頼性が高まりました。また、複数のマウントを同時にサービスできるようになったため、サーバーが使用できないときにサービスが停止することも避けられます。

この新しい automountd には、オンデマンドマウント機能もあります。今までのリリースでは、階層に含まれるすべてのファイルシステムがマウントされていました。これからは、一番上のファイルシステムしかマウントされません。そのマウントポイントに関係する他のファイルシステムは、必要に応じてマウントされます。

autofs サービスで、間接マップを表示できるようになりました。これによりユーザーは、どのディレクトリがマウントできるかを確認するためにファイルシステムを実際に 1 つずつマウントする必要がなくなります。autofs マップに –nobrowse オプションが追加されたので、/net や /home などの大きなファイルが自動的に表示されることはありません。また、automount に対して –n を使用することによって、autofs の表示機能を各クライアントでオフにすることもできます。

第 30 章

リモートファイルシステムの管理

この章では、NFS 管理作業の実行方法について説明します。具体的には、NFS サービスの設定、共有する新規ファイルシステムの追加、ファイルシステムのマウント、Secure NFS システムの使用、WebNFS 機能の使用などです。章の最後では障害追跡の手順を説明し、NFS の多くのエラーメッセージとその意味を示します。

NFS 管理者の責任は、サイトの要求やネットワーク上に存在するコンピュータの役割によって変わります。管理者がローカルネットワークのコンピュータすべてに責任を持つこともありえます。そのような場合は、以下の設定事項について判断する必要があります。

- サーバー専用のコンピュータを置く場合、そのコンピュータの決定
- サーバーとクライアントの両方として動作するコンピュータの決定
- クライアントとしてのみ動作するコンピュータの決定

設定が完了したサーバーの保守には、以下の作業が必要です。

- ファイルシステムの共有開始と共有解除
- 管理ファイルを修正し、コンピュータが共有したり、自動的にマウントしたファイルシステムのリストを更新したりすること
- ネットワークの状態のチェック
- NFS に関連した問題の診断と解決
- autofs のマップの設定

コンピュータは、サーバーとクライアントのどちらにもなれることに注意してください。ローカルファイルシステムをリモートコンピュータと共有したり、リモートファイルシステムをマウントしたりできます。

ファイルシステムの自動共有

NFS 環境でファイルシステムを共有することにより、サーバーのファイルシステムにアクセスできるようになります。共有するファイルシステムは、`share` コマンドや `/etc/dfs/dfstab` ファイルに指定します。

`/etc/dfs/dfstab` ファイルの項目は、NFS サーバーオペレーションを起動したときに自動的に共有されます。同じファイルシステムを定期的に共有する必要がある場合は、自動共有を設定しなければなりません。たとえばサーバーがホームディレクトリをサポートしている場合、ホームディレクトリを常に使用できるようにしておく必要があります。ファイルシステムの共有はほとんどが自動的に行われます。共有を手動で実行するのは、テストか障害追跡の場合だけです。

`dfstab` ファイルには、サーバーがクライアントと共有するすべてのファイルシステムが列挙されており、どのクライアントがファイルシステムをマウントできるかを制御します。`dfstab` を修正してファイルシステムの追加や削除を行う場合、または共有方法を修正する場合には、ファイルを `vi` などのテキストエディタで編集します。コンピュータが次に実行レベル 3 に入ったときに、システムが更新された `dfstab` を読み、ファイルシステムの共有方法が自動的に判断されます。

`dfstab` ファイルの各行は、`share` コマンドで構成されています。その `share` コマンドは、コマンド行プロンプトに入力してファイルシステムを共有するのと同じコマンドです。`share` コマンドは、`/usr/sbin` に保存されています。

表 30–1　ファイルシステム共有作業マップ

作業	説明	参照個所
自動ファイルシステムの共有を確立する	サーバーのリブート時、ファイルシステムが自動的に共有されるようにサーバーを設定する手順	609ページの「ファイルシステム自動共有を設定する方法」
WebNFS を有効にする	ユーザーが WebNFS を使用してファイルにアクセスできるようにサーバーを設定する手順	610ページの「WebNFS アクセスを有効にする方法」
NFS サーバーログを有効にする	NFS ログが選択したファイルシステム上で動作するようにサーバーを設定する手順	611ページの「NFS サーバーログを有効にする方法」

▼ ファイルシステム自動共有を設定する方法

1. スーパーユーザーになります。

2. 共有する対象の各ファイルシステムに関してエントリを追加します。

   `/etc/dfs/dfstab` を編集し、自動的に共有したい各ファイルシステムのファイルにエントリを 1 つ追加します。各エントリは、ファイルの 1 行に納める必要があり、次のような構文を使用します。

   ```
   share [-F nfs] [-o specific-options] [-d description] pathname
   ```

   すべてのオプションを記載したリストについては、`share_nfs(1M)` のマニュアルページを参照してください。

3. **NFS** サービスがサーバーで動作していることを確認します。

   `share` コマンド、または `share` コマンドセットを初めて実行する場合は、NFS デーモンが動作していないことがあります。次のコマンドでデーモンを終了し、再起動してください。

   ```
   # /etc/init.d/nfs.server stop
   # /etc/init.d/nfs.server start
   ```

   これで NFS サービスがサーバーで実行されます。ブート時にサーバーが実行レベル 3 になったときには、自動的に再起動されます。

次の手順

次の手順は、サーバー上で共有化したファイルシステムにクライアントがアクセスできるよう autofs マップを設定する手順です。627ページの「autofs 管理作業の概要」を参照してください。

▼ WebNFS アクセスを有効にする方法

リリース 2.6 から採用した機能で、デフォルトでは、NFS のマウントが利用可能なファイルシステムはすべて、WebNFS アクセスも自動的に利用できます。この手順が必要とされるのは、NFS のマウントがまだ許可されていないサーバー上で、NFS の URL を短くするのに有効な公共ファイルハンドルのリセットが必要とされる場合、または –index オプションが必要な場合です。

1. スーパーユーザーになります。

2. **WebNFS** サービスを使用して、共有する各ファイルシステムのエントリを追加します。

   /etc/dfs/dfstab を編集し、–public オプションを使用して各ファイルシステムについて、エントリを 1 つ追加します。次に示す例の –index タグはオプションです。

   ```
   share -F nfs -o ro,public,index=index.html /export/ftp
   ```

   すべてのオプションを記載したリストについては、share_nfs(1M) のマニュアルページを参照してください。

3. **NFS** サービスがサーバー上で動作していることを確認します。

   share コマンドまたは share コマンドのセットを初めて実行する場合、NFS デーモンが動作していないことがあります。その場合、次のコマンドでデーモンを終了し、デーモンを再起動してください。

   ```
   # /etc/init.d/nfs.server stop
   # /etc/init.d/nfs.server start
   ```

4. ファイルシステムを共有します。

エントリを /etc/dfs/dfstab に追加したあと、システムをリブートするか、shareall コマンドを使用して、ファイルシステムを共有可能にできます。NFS デーモンが手順 2 で再起動されている場合、このコマンドはスクリプトにより実行されているため、実行する必要はありません。

```
# shareall
```

5. 情報が正しいことを確認します。

   share コマンドを実行し、適切なオプションが表示されていることを確認します。

```
# share
-       /export/share/man   ro   ""
-       /usr/src     rw=eng   ""
-       /export/ftp    ro,public,index=index.html  ""
```

▼ NFS サーバーログを有効にする方法

1. スーパーユーザーになります。

2. ファイルシステム構成の設定を変更します **(省略可)**。

   /etc/nfs/nfslog.conf で、global タグに関連するデータを変更してすべてのファイルシステムについてデフォルトの設定を編集するか、このファイルシステムに新しいタグを追加することができます。これらの変更が必要でない場合には、このファイルを変更する必要はありません。/etc/nfs/nfslog.conf の形式については、nfslog.conf(1) のマニュアルページで説明しています。

3. **NFS** サーバーログを使用して、共有する各ファイルシステムについてエントリを追加します。

   /etc/dfs/dfstab を編集し、NFS サーバー記録を有効にしたいファイルシステムについてエントリを 1 つ追加します。log=*tag* オプションと共に使用するタグは、/etc/nfs/nfslog.conf にも記述する必要があります。次の例では、global タグ内のデフォルト設定を使用しています。

```
share -F nfs -o ro,log=global /export/ftp
```

すべてのオプションを記載したリストについては、share_nfs(1M) のマニュアルページを参照してください。

4. **NFS** サービスがサーバー上で動作していることを確認します。

   share コマンドまたは share コマンドセットを初めて実行する場合、NFS デーモンが動作していないことがあります。その場合、次のコマンドでデーモンを終了し、デーモンを再起動してください。

```
# /etc/init.d/nfs.server stop
# /etc/init.d/nfs.server start
```

5. ファイルシステムを共有します。

   エントリを /etc/dfs/dfstab に追加したあと、システムをリブートするか、shareall コマンドを使用して、ファイルシステムを共有可能にできます。NFS デーモンがすでに再起動されている場合、このコマンドはスクリプトにより実行されているため、実行する必要はありません。

```
# shareall
```

6. 情報が正しいことを確認します。

   share コマンドを実行し、適切なオプションが表示されていることを確認します。

```
# share
-       /export/share/man   ro   ""
-       /usr/src     rw=eng   ""
-       /export/ftp    ro,log=global  ""
```

7. **NFS** ログデーモン nfslogd がすでに実行されていなければ、起動します。

   nfs.server スクリプトを使用して NFS デーモンの再起動をすると、nfslog.conf ファイルが存在している場合、デーモンが起動されます。それ以外の場合には、サーバーのリブート時にコマンドが自動的に再起動されるように、一度手動でコマンドを実行してファイルを作成する必要があります。

```
# /usr/lib/nfs/nfslogd
```

ファイルシステムのマウント

ファイルシステムをマウントするには、いくつかの方法があります。システムをブートするときに自動的にマウントされるようにするか、コマンド行から必要に応じてマウントするか、オートマウンタを使用します。オートマウンタには、ブート時のマウントやコマンド行からのマウントに比較していくつもの利点がありますが、状況によってこれら 3 つを組み合わせることが必要です。このような 3 つのファイルシステムのマウント方法に加え、ファイルシステムのマウント時に使用するオプションに応じて、プロセスを有効または無効にする方法がいくつかあります。ファイルシステムのマウントに関するすべての作業のリストについては、表 30–2 を参照してください。

表 30–2 ファイルシステム作業マップ

作業	説明	参照個所
ブート時にファイルシステムをマウントする	システムがリブートされる時に必ずファイルシステムがマウントされるようにする手順	614ページの「ブート時のファイルシステムのマウント方法」
コマンドを使用してファイルシステムをマウントする	システムの動作時にファイルシステムをマウントする手順。この手順はテストに有効	615ページの「コマンド行からファイルシステムをマウントする方法」
オートマウンタによりマウントする	コマンド行を使用せずに、要求に応じてファイルシステムにアクセスする手順	615ページの「オートマウンタによるマウント」
大型ファイルを不許可にする	ファイルシステム上に大型ファイルが作成されないようにする手順	616ページの「NFS サーバー上で大型ファイルを無効にする方法」
クライアント側フェイルオーバーを使用する	サーバーの不良時、動作中のファイルシステムへの自動切り換えを有効にする手順	617ページの「クライアント側フェイルオーバーを使用する方法」
クライアントに対するマウントアクセスを無効にする	任意のクライアントがリモートシステムにアクセスする機能を無効にする手順	618ページの「1 つのクライアントに対するマウントのアクセスを無効にする方法」

表 30–2 ファイルシステム作業マップ 続く

作業	説明	参照個所
ファイアウォール経由のファイルシステムへのアクセスを提供する	WebNFS プロトコルを使用してファイアウォールを越えてファイルシステムへのアクセスを許可する手順	618ページの「ファイアウォールを越えて NFS ファイルシステムをマウントする方法」
NFS URL を使用してファイルシステムをマウントする	NFS URL を使用してファイルシステムへのアクセスを許可する手順。このプロセスによって、MOUNT プロトコルを使用しないでファイルシステムへのアクセスが可能になる	619ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」

▼ ブート時のファイルシステムのマウント方法

autofs マップを使用するのではなく、ブート時にファイルシステムをマウントするには、次の手順に従います。この手順は、すべてのローカルファイルシステムで実行しなければなりません。すべてのクライアントで実行しなければならないので、リモートファイルシステムでの実行は推奨できません。

1. **スーパーユーザーになります。**

2. **ファイルシステムに関するエントリを `/etc/vfstab` に追加します。**

`/etc/vfstab` ファイルのエントリ構文は、次のとおりです。

```
special  fsckdev  mountp  fstype  fsckpass  mount-at-boot  mntopts
```

詳細は、`vfstab(4)` のマニュアルページを参照してください。

注意 - NFS サーバーに NFS `vfstab` ファイルのエントリを作成するとデッドロックが発生する可能性があるため、作成しないでください。NFS サービスは、`/etc/vfstab` のエントリがチェックされてから起動されます。そのため、互いのファイルシステムをマウントしている 2 台のサーバーが同時にダウンすると、リブート中にシステムがハングする可能性があります。

vfstab ファイルの項目の例

wasp サーバーの /var/mail ディレクトリをクライアントに /var/mail としてマウントするとします。そのためには、クライアント側で、ファイルシステムを /var/mail としてマウントし、読み出しと書き込みの両方ができるようにします。この場合は、以下の項目をクライアントの vfstab ファイルに追加します。

```
wasp:/var/mail - /var/mail nfs - yes rw
```

▼ コマンド行からファイルシステムをマウントする方法

コマンド行からのファイルシステムのマウントは多くの場合、新しいマウントポイントのテスト、またはオートマウンタを使用しては利用することができないファイルシステムへの一時的なアクセスを許可するために行われます。

1. スーパーユーザーになります。

2. ファイルシステムをマウントします。

   次のコマンドを入力します。

   ```
   # mount -F nfs -o ro bee:/export/share/local /mnt
   ```

   上の例では、bee サーバーの /export/share/local ファイルシステムが、ローカルシステムの /mnt に読み取り専用でマウントされます。コマンド行からこのようにマウントすることにより、ファイルシステムを一時的に表示することができます。umount を実行するかローカルホストをリブートすると、このマウントは解除されます。

注意 - Solaris 2.6 およびそれ以降に出たパッチに置き換えられた mount コマンドでは、無効なオプションを指定しても警告されません。解釈できないオプションがあると無視されるだけです。予想外の結果が生じるのを避けるために、使用するオプションはすべて確認してください。

オートマウンタによるマウント

627ページの「autofs 管理作業の概要」では、オートマウンタによるマウントの確立とサポートについて詳細に説明します。通常のシステムに変更を加えずに、リモー

トファイルシステムが /net マウントポイントでアクセスできるようになります。前の例のように /export/share/local ファイルシステムをマウントする場合、次の内容を入力するだけです。

```
% cd /net/bee/export/share/local
```

オートマウンタでは、すべてのユーザーがファイルシステムをマウントできるので、root としてアクセスする必要はありません。またファイルシステムのマウントを自動的に解除できるので、作業の終了後、ファイルシステムのマウントを解除する必要はありません。

▼ NFS サーバー上で大型ファイルを無効にする方法

2G バイトを越えるファイルを処理できないクライアントをサポートしているサーバーについては、大型のファイルを作成する機能を無効にしておく必要があります。

注 - 以前のバージョンの Solaris による動作環境では、大型のファイルは使用できません。クライアントが大型のファイルにアクセスする必要がある場合には、NFS サーバーのクライアントが 2.6 以降のリリースで動作していることを確認してください。

---

1. スーパーユーザーになります。
2. ファイルシステム上に大型のファイルが存在していないことを確認してください。

   次の例は、大型のファイルを検索するためのコマンドです。

```
# cd /export/home1
# find . -xdev -size +2000000 -exec ls -l {} \;
```

   システム上に大型のファイルが存在する場合には、削除するか、他のファイルシステムに移動する必要があります。

3. ファイルシステムをアンマウントします。

```
# umount /export/home1
```

4. ファイルシステムがマウントされている場合には、–largefiles を使用してファイルシステムをリセットします。

   fsck は、ファイルシステム上に大型のファイルが存在しない場合に、ファイルシステムの状態をリセットします。

```
# fsck /export/home1
```

5. nolargefiles を使用して、ファイルシステムをマウントします。

```
# mount -F ufs -o nolargefiles /export/home1
```

   コマンド行からこの操作を行うことができますが、オプションをさらに固定的にするために、/etc/vfstab に次のようなエントリを追加してください。

```
/dev/dsk/c0t3d0s1 /dev/rdsk/c0t3d0s1 /export/home1  ufs  2  yes  nolargefiles
```

▼ クライアント側フェイルオーバーを使用する方法

1. スーパーユーザーになります。

2. **NFS** クライアント上で、ro オプションを使用してファイルシステムをマウントします。

   これは、コマンド行からも、オートマウンタを使用しても、または /etc/vfstab ファイルに次のようなエントリを追加することによっても実現できます。

```
bee,wasp:/export/share/local  -  /usr/local  nfs  -  no  -o ro
```

   この構文は古いバージョンのオートマウンタでも受け入れられていましたが、ファイルシステムはマウントされてもフェイルオーバー機能は使用できなかったため、サーバーが選択されるだけでした。

注 - 異なるバージョンの NFS プロトコルを実行しているサーバーを、コマンド行や vfstab のエントリに混在させないでください。サポートしているプロトコルが NFS V2 のサーバーと V3 のサーバーとを混在できるのは、autofs を使用するときだけです。この場合、バージョン 2 かバージョン 3 のサーバーのうち、多い方が使用されます。

▼ 1 つのクライアントに対するマウントのアクセスを無効にする方法

1. スーパーユーザーになります。

2. /etc/dfs/dfstab にエントリを追加します。

   最初の例では、rose という名称のホストを除き、eng ネットグループ内のすべてのクライアントへのマウントアクセスを許可しています。2 つ目の例では、rose を除き、eng.sun.com DNS ドメイン内にあるすべてのクライアントへのマウントアクセスを許可しています。

```
share -F nfs -o ro=-rose:eng /export/share/man
share -F nfs -o ro=-rose:.eng.sun.com /export/share/man
```

   アクセスリストに関する補足的な情報については、682ページの「share コマンドを使用してアクセスリストを設定する」を参照してください。

3. ファイルシステムを共有します。

   /etc/dfs/dfstab への変更は、このファイルシステムがもう 1 度共有されるかサーバーがリブートされるまでは NFS サーバーに反映されません。

```
# shareall
```

▼ ファイアウォールを越えて NFS ファイルシステムをマウントする方法

1. スーパーユーザーになります。

2. 手動でファイルシステムをマウントします。たとえば、次のようなコマンドを入力します 。

```
# mount -F nfs -o public bee:/export/share/local /mnt
```

   この例では、/export/share/local というファイルシステムは、公共ファイルハンドルを使用してローカルクライアントにマウントしています。標準のパス

名の代わりに、NFS URL を使用することができます。ただし bee サーバーで公共ファイルハンドルがサポートされていないと、マウント操作は失敗します。

注 - この手順は、NFS サーバー上のファイルシステムが public オプションを使用して共有されていることと、クライアントとサーバーの間のすべてのファイアウォールでポート 2049 による TCP 接続が可能であることが前提です。Solaris 2.6 からは、共有されているファイルシステムはすべて公共ファイルハンドルによるアクセスが可能です。

▼ NFS URL を使用して NFS ファイルシステムをマウントする方法

1. スーパーユーザーになります。

2. 手動でファイルシステムをマウントします。たとえば、次のようなコマンドを入力します 。

```
# mount -F nfs nfs://bee:3000/export/share/local /mnt
```

この例では、bee というサーバーの /export/share/local というファイルシステムが、NFS ポート番号 3000 を使用してマウントされます。ポート番号は指定しなくてもかまいません。その場合、デフォルトの NFS ポート番号である 2049 が使用されます。NFS URL には public オプションを指定できます。public オプションを指定しない場合、サーバーが公共ファイルハンドルをサポートしていなければ、MOUNT プロトコルが使用されます。public オプションを指定すると、必ず公共ファイルハンドルを使用するように指定され、公共ファイルハンドルがサポートされていないとマウントは失敗します。

NFS サービスの設定

この節では、NFS サービスの初期化や使用に必要な作業をいくつか説明します。

表 30–3　NFS サービス作業マップ

作業	説明	参照個所
NFS サーバーを起動する	NFS サービスが自動的に起動されていない場合に、NFS サービスを起動する手順	620ページの「NFS サービスの起動方法」
NFS サーバーを停止する	NFS サービスを停止する手順。通常は、サービスを停止する必要はない	620ページの「NFS サービスの停止方法」
オートマウンタを起動する	オートマウンタを起動する手順。オートマウンタマップが変更された場合、この手順が必要	621ページの「オートマウンタの起動方法」
オートマウンタを停止する	オートマウンタを停止する手順。オートマウンタマップが変更された場合、この手順が必要	621ページの「オートマウンタの停止方法」

▼ NFS サービスの起動方法

1. スーパーユーザーになります。
2. **NFS** サービスデーモンを有効にします。

   次のコマンドを入力します。

```
# /etc/init.d/nfs.server start
```

   /etc/dfs/dfstab 内にエントリがある場合、このコマンドによりデーモンが起動します。

▼ NFS サービスの停止方法

1. スーパーユーザーになります。
2. **NFS** サービスデーモンを無効にします。

   次のコマンドを入力します。

```
# /etc/init.d/nfs.server stop
```

▼ オートマウンタの起動方法

1. スーパーユーザーになります。
2. **autofs** デーモンを有効にします。
   次のコマンドを入力します。

```
# /etc/init.d/autofs start
```

   デーモンが起動されます。

▼ オートマウンタの停止方法

1. スーパーユーザーになります。
2. **autofs** デーモンを無効にします。
   次のコマンドを入力します。

```
# /etc/init.d/autofs stop
```

Secure NFS システムの管理

Secure NFS システムを使用するには、自分が責任を持つすべてのコンピュータにドメイン名が必要です。「ドメイン」とは管理上のエンティティであり、通常、大きなネットワークに参加する複数のコンピュータから構成されます。NIS+ を実行している場合、そのドメインに対して NIS+ ネームサービスを設定しなければなりません。『*Solaris* ネーミングの設定と構成』を参照してください。

Secure NFS 環境は、認証に Diffie-Hellman (DH) を使用するように設定できます。これらの認証サービスについては、『*Solaris* のシステム管理 *(*第 *2* 巻*)*』の「システムセキュリティの管理の概要」を参照してください。

▼ DH 認証を使用して Secure NFS 環境を設定する方法

1. ドメインにドメイン名を割り当て、そのドメイン名をドメイン内の各コンピュータに知らせます。

   ネームサービスとして NIS+ を使用している場合は、『*Solaris* ネーミングの管理』を参照してください。

2. newkey または nisaddcred コマンドを使用してクライアントのユーザーの公開鍵と秘密鍵を設定して、各ユーザーに chkey コマンドを使用して各自の **Secure RPC** パスワードを設定してもらいます。

   注 - これらのコマンドについての詳細は、newkey(1M)、nisaddcred(1M)、および chkey(1) のマニュアルページを参照してください。

   公開鍵と秘密鍵が生成されると、公開鍵と暗号化された秘密鍵が publickey データベースに格納されます。

3. ネームサービスが応答していることを確認します。**NIS+** を実行している場合は、以下を入力してください。

```
# nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

   NIS を実行している場合は、ypbind デーモンが動作していることを確認してください。

4. keyserv デーモン **(**キーサーバー**)** を確認します。

   次のコマンドを入力してください。

```
# ps -ef | grep keyserv
root    100     1  16    Apr 11 ?      0:00 /usr/sbin/keyserv
root   2215  2211   5  09:57:28 pts/0  0:00 grep keyserv
```

keyserv デーモンが動作していない場合は、次の内容を入力してキーサーバーを起動します。

```
# /usr/sbin/keyserv
```

5. 秘密鍵の復号化と保存を実行します。

   通常、ログインパスワードはネットワークパスワードと同じです。この場合、keylogin は不要です。ログインパスワードとネットワークパスワードが異なる場合、ユーザーはログインしてから keylogin を実行しなければなりません。 また、keylogin -r を root として実行し、復号化した秘密鍵を /etc/.rootkey に保存しなければなりません。

   注 - keylogin -r は、root の秘密鍵が変更されたか、/etc/.rootkey が損失した場合にのみ、実行する必要があります。

   ---

6. ファイルシステムに対するマウントオプションを更新します。

   /etc/dfs/dfstab ファイルを編集し、任意のエントリ (Diffie-Hellman 認証) に sec=dh オプションを追加します。

```
share -F nfs -o sec=dh /export/home
```

7. ファイルシステムに対するオートマウンタマップを更新します。

   auto_master データを編集し、任意のエントリ (Diffie-Hellman 認証) 内にマウントオプションとして sec=dh を含めます。

```
/home auto_home -nosuid,sec=dh
```

注 - リリース 2.5 以前の Solaris では、セキュリティ保護されているものとして共有されているファイルシステムを、セキュリティ保護されたものとしてクライアントがマウントしないと、ユーザーはそのユーザー本来の立場ではなく未認証としてのアクセス権しか得られません。Solaris 2.5 以降の NFS バージョン 2 では、セキュリティモードが一致しないと、`share` コマンド行に `–sec=none` が指定されていない限り、NFS サーバーによってアクセスが拒否されます。NFS のバージョン 3 では、セキュリティ保護されていることを示すモードが NFS サーバーから引き継がれるので、クライアントが `sec=krb4` や `sec=dh` を指定する必要はありません。ユーザーは、そのユーザー自身としてファイルにアクセスできます。

コンピュータを設置し直したり、移設したり、アップグレードしたりするときに、新しい鍵を設定せず、`root` 用の鍵も変更しない場合は、必ず `/etc/.rootkey` を保存してください。`/etc/.rootkey` を削除する場合は、次のコマンドを実行してください。

```
# keylogin -r
```

WebNFS の管理作業

この節では、WebNFS システムを管理する方法を説明します。次に示すのは、関連する作業の一覧です。

表 30–4 WebNFS 管理の作業マップ

作業	説明	参照箇所
WebNFS に関する計画を作成する	WebNFS サービスを有効にする前に考慮する項目	625ページの「WebNFS アクセスの計画」
WebNFS を有効にする	WebNFS プロトコルを使用して NFS ファイルシステムのマウントを有効にする手順	610ページの「WebNFS アクセスを有効にする方法」
ファイアウォール経由で WebNFS を有効にする	WebNFS プロトコルを使用して、ファイアウォール経由でファイルへのアクセスを許可する手順	627ページの「ファイアウォール経由で WebNFS アクセスを有効にする方法」

表 30–4　WebNFS 管理の作業マップ　続く

作業	説明	参照箇所
NFS URL を使用してブラウズする	Web ブラウザ内での NFS URL の使用に関する説明	626ページの「NFS URL を使用したブラウズ方法」
autofs で公共ファイルハンドルを使用する	オートマウンタでファイルシステムをマウントする場合に、公共ファイルハンドルの使用を強制するための手順	642ページの「autofs で公共ファイルハンドルを使用する方法」
autofs で NFS URL を使用する	オートマウンタマップに NFS URL を追加するための手順	642ページの「autofs で NFS URL を使用する方法」
ファイアウォールを越えてファイルシステムにアクセスを提供する	WebNFS プロトコルを使用して、ファイアウォールを越えてファイルシステムにアクセスを許可する手順	618ページの「ファイアウォールを越えて NFS ファイルシステムをマウントする方法」
NFS URL を使用してファイルシステムをマウントする	NFS URL を使用してファイルシステムへのアクセスを許可する手順。このプロセスによって、MOUNT プロトコルを使用しないファイルシステムへのアクセスが可能になる	619ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」

WebNFS アクセスの計画

WebNFS の機能を使用するには、まずアプリケーションを実行して NFS URL (`nfs://server/path` など) を読み込む必要があります。次に、WebNFS アクセスのためにエクスポートするファイルシステムを選択します。アプリケーションが Web ブラウザの場合には、Web サーバーの文書のルートがよく使用されます。WebNFS アクセスのためにエクスポートするファイルシステムを選択するときには、考慮すべきことがいくつかあります。

1. 各サーバーには公共ファイルハンドルが 1 つずつあり、このハンドルはデフォルトではサーバーのルートファイルシステムに結び付けられています。NFS URL に示されたパスは、この公共ファイルハンドルが結び付けられているディレクトリからの相対パスとして評価されます。その結果としてパスが示す先のファイルまたはディレクトリが、エクスポートされたファイルシステムの中にあると、サーバーによってアクセスが実現されます。`share` コマンドの `–public` オプションを使用すると、エクスポートされる特定のディレクトリにこの公開ファイルハンドルを結び付けることができます。このオプションを使用すると、URL はサーバーのルートファイルシステムではなく公共ファイルシステムからの相対

パスになります。デフォルトでは公開ファイルハンドルはルートファイルシステムを示していますが、ルートファイルシステムを共有しないかぎりこのファイルハンドルでは Web アクセスはできません。

2. WebNFS 環境では、すでにマウント権限を持っているユーザーはファイルシステムが –public オプションを使用してエクスポートされているかどうかに関係なく、ブラウザからファイルにアクセスできます。ユーザーは NFS の設定によってファイルに対するアクセス権を持っているため、ブラウザからのアクセスを許すことによって新たにセキュリティが損なわれるおそれはありません。 ファイルシステムをマウントできないユーザーは、–public オプションを使用してファイルシステムを共有するだけで WebNFS アクセスを使えるようになります。
3. FTP アーカイブの最上位ディレクトリや Web サイトの中心となる URL など、すでに公開されているファイルシステムは –public オプションを使用する対象の有力な候補です。
4. share コマンドで –index オプションを使用すると、NFS URL がアクセスされたときにディレクトリがリストされるのではなく HTML ファイルがロードされます。

   ファイルシステムを選択したらファイルを確認し、必要に応じてファイルやディレクトリの表示を制限するようにアクセス権を設定します。アクセス権は、共有される NFS ファイルシステムに合わせて設定します。多くのサイトでは、ディレクトリに対しては 755、ファイルに対しては 644 が適切なアクセスレベルです。

   1 つの Web サイトへのアクセスに NFS URL と HTTP URL の両方を使用する場合には、ほかにも考慮すべき要素があります。701ページの「Web ブラウザの使用と比較した場合の WebNFS の制約」 を参照してください。

▼ NFS URL を使用したブラウズ方法

WebNFS アクセスをサポート可能なブラウザは、次のような形式の NFS URL を使用してアクセスを実現します。

```
nfs://server<:port>/path
```

server	ファイルサーバー名
port	使用するポート番号 (デフォルト値は 2049)
path	公共ファイルハンドラまたはルートファイルシステムに関連するファイルへのパス

注 - ほとんどのブラウザでは、URL サービスタイプ (nfs や http など) は別のサービスタイプの URL が読み込まれるまで次のトランザクションに引き継がれます。NFS URL を使用しているときに HTTP URL への参照が読み込まれると、それ以降のページは次に URL で NFS URL が指定されるまで、NFS プロトコルではなく HTTP プロトコルを使用して読み込まれます。

▼ ファイアウォール経由で WebNFS アクセスを有効にする方法

ローカルのサブネットに属していないクライアントに対して WebNFS アクセスを有効にするには、ポート 2049 での TCP 接続を許可するようにファイアウォールを設定します。httpd に対してアクセスを許可するだけでは、NFS URL が使えるようにはなりません。

autofs 管理作業の概要

このセクションでは、ユーザー自身の環境で遭遇する可能性のある最も一般的な作業について説明します。ユーザーのクライアントの必要に最良な形で適合するように autofs を設定するのに役立つ、各シナリオについて推奨される手順も示します。

注 - このセクションで説明される作業を実行するには、Solstice System Management Tools を使用するか、『*Solaris* ネーミングの管理』を参照してください。

autofs 管理の作業マップ

表 30–5 に、autofs に関連する作業についての説明と参照箇所を示します。

表 30–5 autofs 管理の作業マップ

作業	説明	参照箇所
autofs を起動する	システムをリブートすることなくオートマウントサービスを起動する	621ページの「オートマウンタの起動方法」
autofs を停止する	他のネットワークサービスを使用不能にすることなくオートマウントサービスを停止する	621ページの「オートマウンタの停止方法」
autofs でファイルシステムにアクセスする	オートマウントサービスを使用してファイルシステムにアクセスする	615ページの「オートマウンタによるマウント」
autofs マップを修正する	他のマップをリストするために使用されるマスターマップの修正を行う手順	631ページの「マスターマップの修正方法」
	ほとんどのマップに対して使用される間接マップの修正を行う手順	631ページの「間接マップの修正方法」
	クライアント上のマウントポイントとサーバー間の直接の関係が必要な場合に使用される直接マップの修正を行う手順	631ページの「直接マップの修正方法」
非 NFS ファイルシステムにアクセスするために autofs マップを修正する	CD-ROM アプリケーション用のエントリで autofs マップを設定する手順	633ページの「autofs で CD-ROM アプリケーションにアクセスする」
	PC-DOS ディスケット用のエントリで autofs マップの設定を行う手順	633ページの「autofs で PC-DOS データフロッピーディスクにアクセスする方法」
	autofs を使用して CasheFS ファイルシステムにアクセスする手順	634ページの「CasheFS を使用して NFS ファイルシステムにアクセスする方法」
/home を使用する	共通の /home マップの設定方法の例	635ページの「/home の共通表示の設定」
	複数のファイルシステムを参照する /home マップを設定する手順	636ページの「複数のホームディレクトリファイルシステムで /home を設定する方法」
新しい autofs マウントポイントを使用する	プロジェクト関連の autofs マップを設定する手順	637ページの「/ws 下のプロジェクト関連ファイルを統合する方法」
	異なるクライアントアーキテクチャをサポートする autofs マップを設定する手順	639ページの「共有名前空間にアクセスするために異なるアーキテクチャを設定する方法」

表 30–5　autofs 管理の作業マップ　続く

作業	説明	参照箇所
	異なるオペレーティングシステムをサポートする autofs マップを設定する手順	640ページの「非互換のクライアントオペレーティングシステムのバージョンをサポートする方法」
autofs でファイルシステムを複製する	フェイルオーバーしたファイルシステムへのアクセスを提供する	641ページの「複数のサーバーを通じて共用ファイルを複製する方法」
autofs でセキュリティ制限を使用する	ファイルへのリモート root アクセスを制限する一方でファイルシステムへのアクセスを提供する	641ページの「セキュリティ制限を適用する方法」
autofs で公共ファイルハンドルを使用する	ファイルシステムのマウント時に公共ファイルハンドルの使用を強制する	642ページの「autofs で公共ファイルハンドルを使用する方法」
autofs で NFS URL を使用する	オートマウンタが使用できるように、NFS URL を追加する	642ページの「autofs で NFS URL を使用する方法」
autofs のブラウザ機能を無効にする	autofs マウントポイントが 1 つのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	643ページの「1 つの NFS クライアント上の autofs ブラウズ機能を完全に無効にする方法」
	autofs マウントポイントがすべてのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	644ページの「すべてのクライアントの autofs ブラウズ機能を無効にする方法」
	特定の autofs マウントポイントがある 1 つのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	644ページの「1 つの NFS クライアントの autofs ブラウズ機能を無効にする方法」

管理作業を含むマップ

表 30–6 は、autofs マップの管理時に認識しておく必要のある事項について示したものです。選択したマップのタイプおよびネームサービスにより、autofs マップへの変更を行うために使用する必要があるメカニズムが異なります。

表 30–6 に、マップのタイプとその使用方法を示します。

表 30–6 autofs マップのタイプとその使用方法

マップのタイプ	使用方法
マスター	ディレクトリをマップに関連付ける
直接	autofs を特定のファイルシステム向けにする
間接	autofs をリファレンス指向のファイルシステム向けにする

表 30–7 は、ネームサービスに基づいて autofs 環境に変更を加える方法を示したものです。

表 30–7 マップの保守

ネームサービス	方法
ローカルファイル	テキストエディタ
NIS	`make` files
NIS+	`nistbladm`

表 30–8 に、マップのタイプについて行なった修正に応じた `automount` コマンドを実行する場合を示します。たとえば、直接マップに対する追加または削除を行なった場合、ローカルシステム上で `automount` コマンドを実行し、変更が反映されるようにする必要があります。しかし、既存のエントリを修正した場合は、変更を反映するために、`automount` コマンドを実行する必要はありません。

表 30–8 `automount` コマンドを実行する場合

マップのタイプ	`automount` を再実行するか否か	
	追加または削除	修正
`auto_master`	Y	Y
`direct`	Y	N
`indirect`	N	N

表 30–8　automount コマンドを実行する場合　続く

マップの修正

次の手順では、NIS+ をネームサービスとして使用している必要があります。

▼ マスターマップの修正方法

1. nistbladm コマンドを使用して、マスターマップへの変更を行います。
   『*Solaris* ネーミングの管理』を参照してください。
2. 各クライアントで、スーパーユーザーになります。
3. 各クライアントで、automount コマンドを実行し、変更が反映されるようにします。
4. 他のユーザーに変更を通知します。
   他ユーザーがコンピュータ上でスーパーユーザーとして automount コマンドを実行するように、通知が必要になります。

automount コマンドは、実行時にマスターマップからの情報の収集を行います。

▼ 間接マップの修正方法

- nistbladm コマンドを使用して、間接マップへの変更を行います。
  『*Solaris* ネーミングの管理』を参照してください。

変更はマップを次に使用する時、つまり次回のマウント実行時に反映されます。

▼ 直接マップの修正方法

1. nistbladm コマンドを使用して、直接マップに対する変更点の追加または削除を行います。
   『*Solaris* ネーミングの管理』を参照してください。

2. 手順 1 でマウントポイントエントリの追加または削除を行なった場合には、automount コマンドを実行します。

3. 変更した点を他のユーザーに通知します。

   他ユーザーがコンピュータ上でスーパーユーザーとして automount コマンドを実行するように、通知が必要になります。

注 - 既存の直接マップエントリの内容に修正または変更だけを行なった場合は、automount コマンドを実行する必要はありません。

たとえば、異なるサーバーから /usr/src ディレクトリがマウントされるように auto_direct マップを修正するとします。/usr/src がその時点でマウントされていない場合、/usr/src にアクセスするとすぐにその新しいエントリが反映されます。/usr/src がその時点でマウントされている場合、オートアンマウントが実行されるまで待ちます。そのあと、アクセスが可能になります。

注 - これらの追加の手順が必要だったり、直接マップほどにはマウントテーブル内のスペースを必要としないので、可能であれば必ず間接マップを使用してください。間接マップは構築が容易であり、コンピュータのファイルシステムへの要求が少なくて済みます。

マウントポイントの重複回避

/src 上にマウントされたローカルなディスクパーティションがあり、他のソースディレクトリのマウントにもその autofs サービスを使用したい場合、問題が発生する可能性があります。マウントポイント /src を指定する場合、ユーザーがアクセスするたびに、そのサービスが対象のローカルパーティションを隠すことになります。

その場合、そのパーティションはたとえば /export/src などの他の場所にマウントする必要があります。また、/etc/vfstab 内に次のようなエントリが必要になります。

```
/dev/dsk/d0t3d0s5 /dev/rdsk/c0t3d0s5 /export/src ufs 3 yes -
```

このエントリが auto_src にある場合

```
terra  terra:/export/src
```

terra はコンピュータ名です。

非 NFS ファイルシステムへのアクセス

autofs は NFS ファイル以外のファイルもマウントすることができます。autofs は、フロッピーディスクや CD-ROM など、削除可能な媒体上のファイルをマウントします。通常は、Volume Manager を使用して削除可能な媒体上のファイルをマウントすることになります。次の例では、autofs を利用してこのマウントがどのように行われるかを示します。Volume Manager と autofs は同時に動作することができないため、まず Volume Manager を終了してから次に示すエントリを使用する必要があります。

サーバーからファイルシステムのマウントを行う代わりに、ドライブに媒体を配置してマップから参照します。autofs を使用し非 NFS ファイルシステムにアクセスを行いたい場合は、次の手順を参照してください。

autofs で CD-ROM アプリケーションにアクセスする

注 - Volume Manager を使用していない場合に、この手順を行なってください。

1. スーパーユーザーになります。
2. **autofs** マップを更新します。

   CD-ROM のファイルシステムに対するエントリを追加する場合、次のようになります。

```
hsfs      -fstype=hsfs,ro      :/dev/sr0
```

   マウントしたい CD-ROM 装置の名前が、コロンのあとに続けて表示されます。

▼ autofs で PC-DOS データフロッピーディスクにアクセスする方法

注 - Volume Mananger を使用していない場合に、この手順を行なってください。

1. スーパーユーザーになります。

2. **autofs** マップを更新します。

   ファイルシステムに対するエントリを追加する場合、次のようになります。

```
pcfs      -fstype=pcfs      :/dev/diskette
```

CasheFS を使用して NFS ファイルシステムにアクセスする

キャッシュファイルシステム (CacheFS) は、汎用不揮発性キャッシュメカニズムで、小型で高速ローカルディスクを利用して、特定のファイルシステムの性能を向上させます。

CacheFS を使用してローカルディスク上に NFS ファイルシステムからデータをキャッシュすることにより、NFS 環境の性能が改善できます。

▼ CasheFS を使用して NFS ファイルシステムにアクセスする方法

1. スーパーユーザーになります。

2. `cfsadmin` コマンドを実行して、ローカルディスク上にキャッシュディレクトリを作成します。

```
# cfsadmin -c /var/cache
```

3. 任意のオートマウンタマップに **cachefs** エントリを追加します。

   たとえば、次に示すエントリをマスターマップに追加して、すべてのディレクトリがキャッシュされます。

```
/home auto_home -fstype=cachefs,cachedir=/var/cache,backfstype=nfs
```

   以下のエントリを `auto_home` マップに追加すると、`rich` という名称のユーザーのホームディレクトリのキャッシュだけが行われます。

```
rich -fstype=cachefs,cachedir=/var/cache,backfstype=nfs dragon:/export/home1/rich
```

注 - あとから検索されるマップ内のオプションは、先に検索されたマップ内のオプションを無効にします。そのため、最後に検出されたオプションが使用されます。前の例では、マスターマップにリストされたオプションの中に変更の必要がある場合には、auto_home マップに追加された特定のエントリがそのマスターマップのオプションを含む必要だけがあります。

オートマウンタのカスタマイズ

オートマウンタマップの設定方法はいくつかあります。次の作業説明で、オートマウンタマップをカスタマイズして簡単に使用できるディレクトリ構造を実現する方法について詳細な手順を示します。

▼ /home の共通表示の設定

ネットワークユーザーすべてにとって理想的なのは、自分自身のホームディレクトリ、または他の人のホームディレクトリを /home の下に配置できるようにすることです。この表示方法は通常、クライアントでもサーバーでも、すべてのコンピュータを通じて共通です。

Solaris のインストールはそれぞれ、マスターマップ /etc/auto_master を使用して行われます

```
# Master map for autofs
#
+auto_master
/net     -hosts     -nosuid,nobrowse
/home    auto_home  -nobrowse
/xfn     -xfn
```

auto_home 用のマップも、/etc の下にインストールされます。

```
# Home directory map for autofs
#
+auto_home
```

外部 auto_home マップに対する参照を除き、このマップは空になります。/home 下のディレクトリをすべてのコンピュータに対して共通にする場合、この /etc/auto_home マップは修正しないでください。すべてのホームディレクトリ

のエントリは、NIS または NIS+ のネームサービスファイルで表示されなくてはなりません。

注 - ユーザーは、各ホームディレクトリから `setuid` 実行可能ファイルを実行することが許可されていません。この制限がないと、すべてのユーザーがすべてのコンピュータ上でスーパーユーザーの権限を持つことになります。

▼ 複数のホームディレクトリファイルシステムで /home を設定する方法

1. スーパーユーザーになります。

2. `/export/home` の下にホームディレクトリパーティションをインストールします。

   複数のパーティションがある場合には、`/export/home1`、`/export/home2` のように、別のディレクトリにそれぞれインストールを行います。

3. `auto_home` マップを作成して維持するため、**Solstice System Management Tools** を使用します。

   新しいユーザーアカウントを作成する場合には、そのユーザーのホームディレクトリの位置を `auto_home` マップに入力します。マップのエントリは、次のように単純な形式にできます。

```
rusty        dragon:/export/home1/&
gwenda       dragon:/export/home1/&
charles      sundog:/export/home2/&
rich         dragon:/export/home3/&
```

   マップキーの代用する `&` (アンパサンド) の使い方に注意してください。これは、次の例での、2 つ目の `rusty` の使用を省略した形式です。

```
rusty      dragon:/export/home1/rusty
```

   `auto_home` マップが適切に配置されている場合、ユーザーは `/home/`*user* パスを使用して、ユーザー自身のホームディレクトリを含むすべてのホームディレクトリが参照できます。*user* がログイン名であり、マップ内でのキーです。すべてのホームディレクトリを共通に表示する仕組みは、他ユーザーのコンピュータにログインする場合に便利です。autofs は、ユーザー自身のホームディレクトリを

マウントします。同様に、他のコンピュータ上でリモートウィンドウシステムのクライアントを動作させる場合、クライアントプログラムは、ウィンドウシステム表示を提供するコンピュータと同じ /home ディレクトリを表示します。

この共通表示は、サーバーにも拡張されています。前の例を使用すれば、rusty がサーバー dragon にログインする場合、autofs は、/export/home1/rusty を /home/rusty にループバックマウントすることにより、ローカルディスクへの直接アクセスを提供します。

ユーザーは、各ホームディレクトリの実際の位置を意識する必要はありません。rusty がさらにディスクスペースを必要とし、rusty 自身のホームディレクトリを他のサーバーに再配置する必要がある場合には、auto_home マップ内の rusty のエントリを新しい位置を反映するように変更することだけが必要になります。すべてのユーザーは、/home/rusty パスを継続して使用することができます。

▼ /ws 下のプロジェクト関連ファイルを統合する方法

大きなソフトウェアの開発プロジェクトの管理者を想定してください。そこで、プロジェクト関連のファイルをすべて /ws というディレクトリの下で利用できるようにしたいと仮定します。このようなディレクトリは、そのサイトのすべてのワークステーションで共通である必要があります。

1. /ws ディレクトリに対するエントリを、サイトの **NIS** または **NIS+** の auto_master マップに追加します。

```
/ws     auto_ws     -nosuid
```

   auto_ws マップが、/ws ディレクトリの内容を決定します。

2. -nosuid オプションを用心のために追加しておきます。

   このオプションは、すべての作業空間に存在する可能性のある setuid プログラムをユーザーが動作できないようにします。

3. auto_ws マップにエントリを追加します。

   auto_ws マップは、各エントリがサブプロジェクトを記述するように構成されています。最初の操作により、マップが次のようになります。

```
compiler   alpha:/export/ws/&
windows    alpha:/export/ws/&
files      bravo:/export/ws/&
drivers    alpha:/export/ws/&
man        bravo:/export/ws/&
tools      delta:/export/ws/&
```

各エントリの最後のアンパサンド (&) は、エントリキーを省略したものです。たとえば、最初のエントリは次のエントリと同じ意味です。

```
compiler  alpha:/export/ws/compiler
```

この最初の操作によりマップは単純に見えるようになりますが、不適切な場合もあります。プロジェクトの調整者が、man エントリ内のドキュメントが各サブプロジェクトの下のサブディレクトリとして提供されるべきである見なすとします。さらに、各サブプロジェクトは、ソフトウェアの複数のバージョンを記述するために、複数のサブディレクトリを必要とします。この場合、サーバー上のディスクパーティション全体に対して、これらのサブディレクトリをそれぞれ割り当てる必要があります。

次のように、マップ内のエントリを修正してください。

```
compiler \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /vers2.0    bravo:/export/ws/&/vers2.0 \
    /man        bravo:/export/ws/&/man
windows \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /man        bravo:/export/ws/&/man
files \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /vers2.0    bravo:/export/ws/&/vers2.0 \
    /vers3.0    bravo:/export/ws/&/vers3.0 \
    /man        bravo:/export/ws/&/man
drivers \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /man        bravo:/export/ws/&/man
tools \
    /           delta:/export/ws/&
```

現在のマップはかなり長くなっていますが、まだ 5 つのエントリを含んでいるだけです。各エントリは、複数のマウントがあるために長くなっています。たとえば、/ws/compiler に対する参照は、vers1.0、vers2.0、および man ディレクトリ用に 3 つのマウントを必要とします。各行の最後のバックスラッシュ

は、エントリが次の行まで続いていることを autofs に伝えるものです。実際、エントリは 1 つの長い行となっていますが、行ブレークやインデントのいくつかはエントリを読みやすくする目的で使用されています。tools ディレクトリには、すべてのサブプロジェクトに対するソフトウェア開発ツールが含まれているため、同じサブディレクトリ構造の対象とはなっていません。tools ディレクトリは単一のマウントのままです。

この配置は、システムの管理者に大きな柔軟性を提供します。ソフトウェアプロジェクトでは、非常に大きなディスクスペースを消費します。プロジェクトのすべての過程を通じて、さまざまなディスクパーティションを再配置し、拡張することになる可能性もあります。このような変更が auto_ws マップに反映される限り、/ws 下のディレクトリ階層構造が変更されることもなく、ユーザーに対する通知の必要はありません。

サーバー alpha と bravo が同一の autofs マップを参照するため、それらのコンピュータにログインするすべてのユーザーは期待通りに /ws 名前空間を発見できます。このようなユーザーには、NFS マウントではなく、ループバックマウントを通じてのローカルファイルへの直接アクセスが提供されます。

▼ 共有名前空間にアクセスするために異なるアーキテクチャを設定する方法

表計算ツールやワードプロセッサパッケージのようなローカルな実行可能ファイルやアプリケーションについて、共有名前空間を作成する必要があります。この名前空間のクライアントは、異なる実行可能フォーマットを必要とする複数の異なるワークステーションアーキテクチャを使用します。また、ワークステーションには、異なるリリースのオペレーシングシステムを使用するものもあります。

1. nistabladm コマンドで auto_local マップを作成します。

   『*Solaris* ネーミングの管理』を参照してください。

2. その名前空間に属するファイルとディレクトリが簡単に識別できるように、共有名前空間について、サイト固有の名称を **1** つ選択します。

   たとえば、その名称として /usr/local を選択した場合、/usr/local/bin パスは明らかにこの名前空間の一部です。

3. ユーザーのコミュニティ識別を簡単にするため、**autofs** 間接マップを作成し、/usr/local にマウントします。**NIS (**または **NIS+)** の auto_master マップ内で、次のエントリを設定します。

```
/usr/local      auto_local      -ro
```

なお、–ro マウントオプションは、クライアントがファイルやディレクトリのすべてに対して書き込みができないことを示してます。

4. サーバー上の任意のディレクトリをエクスポートします。

5. **auto_local** マップ内に bin エントリを **1** つ含めます。

   ディレクトリ構造は次のようになります。

```
 bin      aa:/export/local/bin
```

   異なるアーキテクチャのクライアントを処理するため、クライアントのアーキテクチャタイプに応じて、その bin ディレクトリへの参照をサーバー上の異なるディレクトリに割り当てる必要があります。

6. 異なるアーキテクチャのクライアントを処理するため、**autofs** CPU 変数を加えて、エントリの変更を行います。

```
bin      aa:/export/local/bin/$CPU
```

   - SPARC クライアント – 実行可能ファイルを /export/local/bin/sparc に配置する
   - IA クライアント – 実行可能ファイルを /export/local/bin/i386 に配置する

▼ 非互換のクライアントオペレーティングシステムのバージョンをサポートする方法

1. クライアントのオペレーティングシステムのタイプを決定する変数と、アーキテクチャタイプを結合します。

   CPU タイプと OS リリースの両方を特定する名称を作成するために、autofs OSREL 変数は CPU 変数と結合することができます。

2. 次のようなマップエントリを作成します。

```
bin      aa:/export/local/bin/$CPU$OSREL
```

   バージョン 5.6 のオペレーティングシステムを動作させているクライアントについて、次のファイルシステムをエクスポートします。

- SPARC クライアント – /export/local/bin/sparc5.6 をエクスポートする
- IA クライアント – /export/local/bin/i3865.6 に実行可能ファイルを配置する

▼ 複数のサーバーを通じて共用ファイルを複製する方法

読み取り専用の複製されたファイルシステムを共有する最良の方法は、フェイルオーバーの利用です。フェイルオーバーについての説明は、696ページの「クライアント側フェイルオーバー機能」を参照してください。

1. スーパーユーザーになります。
2. **autofs** マップ内のエントリを修正します。

   すべての複製サーバーのリストを、コンマ区切りのリストとして、次のように作成します。

```
bin     aa,bb,cc,dd:/export/local/bin/$CPU
```

   autofs は、最も近いサーバーを選択します。サーバー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、各インタフェースのリストを作成してください。autofs はクライアントに最も近接したインタフェースを選択し、NFS トラフィックの不必要なルーティングを避けるようにしています。

▼ セキュリティ制限を適用する方法

1. スーパーユーザーになります。
2. **NIS** または **NIS+** のネームサービス auto_master ファイル内に次のようなエントリを作成します。

```
/home     auto_home     -nosuid
```

   nosuid オプションは、setuid または setgid ビットを設定したファイルをユーザーが作成できないようにします。

このエントリは、通常のローカルな /etc/auto_master ファイル内の /home に関するエントリを無効にします (前述の例を参照)。これは、ファイル内の /home エントリの前に、+auto_master の外部ネームサービスマップへの参照が生じるためです。auto_home マップ内のエントリがマウントオプションを含む場合、nosuid オプションは無効になります。そのため、オプションを auto_home マップで使用しないか、nosuid オプションを各エントリに含むことになります。

注 - サーバー上の /home またはその下に、ホームディレクトリのディスクパーティションをマウントしないでください。

▼ autofs で公共ファイルハンドルを使用する方法

1. スーパーユーザーになります。
2. 次のように、**autofs** マップ内にエントリを作成します。

```
/usr/local      -ro,public     bee:/export/share/local
```

public オプションは、公共ハンドルの使用を強制します。NFS サーバーが公共ファイルハンドルをサポートしない場合、マウントは失敗します。

▼ autofs で NFS URL を使用する方法

1. スーパーユーザーになります。
2. 次のように、**autofs** エントリを作成します。

```
/usr/local      -ro     nfs://bee/export/share/local
```

サービスは、NFS サーバー上で公共ファイルハンドルの使用を試みますが、そのサーバーが公共ファイルハンドルをサポートしない場合、MOUNT プロトコルが使用されます。

autofs のブラウザ機能を無効にする

Solaris 2.6 リリースから使い始めた場合、インストールされる /etc/auto_master のデフォルトバージョンには、/home と /net 用のエントリに追加された –nobrowse オプションが含まれます。さらに、アップグレード手順により、/home と /net のエントリが修正されていない場合には、–nobrowse オプションが それらエントリに追加されます。ただし、このような変更を手動で加えるか、またはインストール後にサイト固有の autofs マウントポイントに対するブラウズ機能をオフにすることが必要な場合もあります。

ブラウズ機能をオフにする方法はいくつかあります。automountd デーモンに対してコマンド行オプションを使用してオフにすると、そのクライアントに対する autofs ブラウザ機能は完全に無効になります。あるいは、NIS または NIS+ 名前空間内の autofs マップを使用してすべてのクライアント上の各マップエントリに対してブラウザ機能をオフにできます。またネットワーク全体に渡る名前空間が使用されていない場合には、ローカルな autofs マップを使用して、各クライアント上の各マップエントリに対してブラウザ機能をオフにできます。

▼ 1 つの NFS クライアント上の autofs ブラウズ機能を完全に無効にする方法

1. スーパーユーザーになります。

2. –n オプションを起動クリプトに追加します。

   root として、/etc/init.d/autofs スクリプトを編集し、automountd デーモンを起動する行に –n オプションを追加します。

```
/usr/lib/autofs/automountd -n \
 < /dev/null > /dev/console 2>&1 # start daemon
```

3. **autofs** サービスを再起動します。

```
# /etc/init.d/autofs stop
# /etc/init.d/autofs start
```

▼ すべてのクライアントの autofs ブラウズ機能を無効にする方法

すべてのクライアントに対するブラウズ機能を無効にするには、NIS または NIS+ のようなネームサービスを使用する必要があります。それ以外の場合には、各クライアント上でオートマウンタマップを手動で編集する必要があります。この例では、/home ディレクトリのブラウズ機能が無効にされています。無効にする必要がある各間接 autofs ノードに対して、この手順を実行してください。

1. ネームサービス auto_master ファイル内の /home エントリに –nobrowse オプションを追加します。

```
/home      auto_home      -nobrowse
```

2. すべてのクライアント上で、automount コマンドを実行します。

   新規の動作は、クライアントシステム上での automount コマンドを実行したあと、またはリブートのあとで反映されます。

```
# /usr/sbin/automount
```

▼ 1 つの NFS クライアントの autofs ブラウズ機能を無効にする方法

この例では、/net ディレクトリのブラウズ機能を無効にします。同じ手順が、/home やその他の autofs マウントポイントに対して使用することができます。

1. automount エントリが /etc/nsswitch.conf にあることを確認します。

   優先するローカルファイルエントリについては、ネームサービススイッチファイル内のエントリがネームサービスの前に files をリストする必要があります。たとえば、次のようになります。

```
automount:  files nisplus
```

これは、標準的な Solaris にインストールされるデフォルトの構成です。

2. /etc/auto_master 内の +auto_master エントリの位置を確認します。

   名前空間内のエントリに優先するローカルファイルへの追加については、+auto_master エントリが /net の下に移動されている必要があります。

```
# Master map for automounter
#
/net    -hosts      -nosuid
/home   auto_home
/xfn    -xfn
+auto_master
```

   標準的な構成では、+auto_master エントリがファイルの先頭に配置されます。これにより、ローカルな変更が使用されなくなります。

3. /etc/auto_master ファイル内の /net エントリに –nobrowse オプションを追加します。

```
/net      -hosts      -nosuid,nobrowse
```

4. すべてのクライアント上で、automount コマンドを実行します。

   新規の動作は、クライアントシステム上で automount コマンドを実行したあと、またはリブートしたあとで反映されます。

```
# /usr/sbin/automount
```

NFS における障害追跡の方法

NFS のトラブルを追跡するとき、主な障害発生ポイントとしてサーバー、クライアント、またはネットワーク自体の 3 つがあることを覚えておいてください。この節で説明するのは、個々の構成要素を切り離して、正常に動作しない部分を見つけ出

そうというものです。リモートマウントを正常に実行するには、サーバー上で mountd と nfsd が動作している必要があります。

注 - /etc/dfs/dfstab ファイルに NFS 共有エントリがある場合、mountd と nfsd はブート時に自動的に起動します。したがって、最初に共有設定を行うときには mountd と nfsd を手作業で起動しなければなりません。

デフォルトでは、すべてのマウントに –intr オプションが設定されます。プログラムが「server not responding」(サーバーが応答しません) というメッセージを出してハングした場合、これはキーボード割り込み (Ctrl-C) で終了できます。

ネットワークまたはサーバーに問題がある場合、ハードマウントされたリモートファイルにアクセスするプログラムの障害と、ソフトマウントされたリモートファイルにアクセスするプログラムの障害とは異なります。ハードマウントされたリモートファイルシステムの場合、クライアントのカーネルは、サーバーが再び応答するまで要求を再試行します。ソフトマウントされたリモートファイルシステムの場合、クライアントのシステムコールは、しばらく試行した後でエラーを返します。このエラーによって予想外のアプリケーションエラーやデータ破壊が起きる恐れがあるため、ソフトマウントは行わないでください。

ファイルシステムがハードマウントされていると、サーバーが応答に失敗した場合には、これにアクセスしようとするプログラムはハングします。この場合、NFS は次のメッセージをコンソールに表示します。

```
NFS server hostname not responding still trying
```

サーバーが少し後に応答すると、次のメッセージがコンソールに表示されます。

```
NFS server hostname ok
```

サーバーが応答しないような、ソフトマウントされたファイルシステムにアクセスしているプログラムは、次のメッセージを表示します。

```
NFS operation failed for server hostname: error # (error_message)
```

注 - 読み取りと書き込みをするデータを持つファイルシステム、または実行可能ファイルを持つファイルシステムは、ソフトマウントしないでください。エラーが発生する可能性があります。アプリケーションがそのようなソフトエラーを無視すれば、書き込み可能なデータが破壊される恐れがあります。またマウントされた実行可能ファイルが正常にロードされず、動作も正常に行われない可能性があります。

NFS における障害追跡の手順

NFS サービスがエラーになった場所を判断するには、いくつかの手順を踏まなければなりません。次の項目をチェックしてください。

- クライアントとサーバーがソフトウェア的に接続されているかどうか
- クライアントが NFS サービスを受けられるかどうか
- NFS サービスがサーバー上で動作しているかどうか

上記の項目をチェックする過程で、ネームサービスやネットワークのハードウェアなど、ネットワークの他の部分が機能していないことが判明する場合があります。NIS+ ネームサービスのデバッグ手順については、『*Solaris* ネーミングの管理』を参照してください。問題がクライアント側のものではないことが判明することもあります (たとえば作業領域の各サブネットから、最低 1 つのトラブルコールがある場合など)。このような場合は、問題がサーバーかサーバー周辺のネットワークハードウェアで発生しているとみなし、クライアントではなく、サーバーでデバッグを開始するほうがよいでしょう。

▼ NFS クライアントの接続性を確認する方法

1. **クライアントと NFS サーバーが、ソフトウェア的に接続されていることを確認します。次のコマンドをクライアントで入力します。**

```
% /usr/sbin/ping bee
bee is alive
```

コマンドを入力した結果、サーバーが動作していることがわかったら、NFS サーバーをリモートで確認します (648ページの「NFS サーバーをリモートで確認する方法」 を参照)。

2. **クライアントとサーバーがソフトウェア的に接続されていない場合は、ローカルネームサービスが動作していることを確認します。NIS+ クライアントでは、次のように入力します。**

```
% /usr/lib/nis/nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

3. ネームサービスが実行されている場合は、クライアントが正しいホスト情報を受け取るために次のように入力します。

```
% /usr/bin/getent hosts bee
129.144.83.117 bee.eng.acme.com
```

4. ホスト情報に誤りがなく、クライアントからサーバーに接続できない場合は、別のクライアントから ping コマンドを実行します。

   ping コマンドが失敗したら、650ページの「サーバーで NFS サービスを確認する方法」 を参照してください。

5. 別のクライアントとサーバーがソフトウェア的に接続されている場合は、ping コマンドを使用して元のクライアントとローカルネット上の他のシステムとの接続性を確認します。

   エラーになる場合は、そのクライアントのネットワークソフトウェアの構成を確認します (/etc/netmasks、/etc/nsswitch.conf など)。

6. ソフトウェアに問題がない場合は、ネットワークハードウェアを確認します。

   クライアントをネットワークの別の場所へ移動して確認します。

▼ NFS サーバーをリモートで確認する方法

1. **NFS** サーバーで **NFS** サービスが実行されていることを、次のコマンドを入力することによって確認します。

```
% rpcinfo -s bee|egrep 'nfs|mountd'
 100003  3,2    tcp,udp                          nfs     superuser
 100005  3,2,1  ticots,ticotsord,tcp,ticlts,udp  mountd  superuser
```

デーモンが起動していなければ、652ページの「NFS サービスを再起動する方法」 を参照してください。

2. サーバーで nfsd プロセスが応答することを確認します。クライアントで次のコマンドを入力します。

```
% /usr/bin/rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

サーバーが動作している場合、プログラムとバージョン番号が表示されます。–t オプションを使用すると、TCP 接続を検査できます。–t オプションでエラーが発生する場合は、650ページの「サーバーで NFS サービスを確認する方法」 に進んでください。

3. サーバーで mountd が応答することを確認します。次のコマンドを入力します。

```
% /usr/bin/rpcinfo -u bee mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
```

–t オプションを使用すると、TCP 接続を検査できます。エラーになる場合は、650ページの「サーバーで NFS サービスを確認する方法」 に進んでください。

4. ローカル **autofs** サービスを使用していた場合は、そのサービスを確認します。

```
% cd /net/wasp
```

/net か /home マウントポイントのうち、適切に動作する方を確認します。動作しない場合は、次のコマンドをルートとしてクライアントから入力し、autofs サービスを再起動します。

```
# /etc/init.d/autofs stop
# /etc/init.d/autofs start
```

5. サーバーのファイルシステムの共有が正常に行えることを確認します。

```
% /usr/sbin/showmount -e bee
/usr/src          eng
/export/share/man      (everyone)
```

サーバーの項目とローカルマウントエントリにエラーがないことをチェックします。名前空間も確認します。この例で最初のクライアントが eng ネットグループの中にない場合、/usr/src ファイルシステムはマウントできません。

すべてのローカルファイルを調べて、マウント情報を含むエントリをすべて検査します。リストには、/etc/vfstab とすべての /etc/auto_* ファイルが含まれています。

▼ サーバーで NFS サービスを確認する方法

1. スーパーユーザーになります。

2. サーバーとクライアントがソフトウェア的に接続されていることを確認します。

```
# ping lilac
lilac is alive
```

3. サーバーとクライアントがソフトウェア的に接続されていない場合は、ローカルネームサービスが動作していることを確認します。**NIS+** クライアントで次のコマンドを入力します。

```
% /usr/lib/nis/nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

4. ネームサービスが動作している場合は、サーバーにあるネットワークソフトウェアの構成を確認します **(**/etc/netmasks、 /etc/nsswitch.conf など**)**。

5. 次のコマンドを入力し、nfsd デーモンが動作していることを確認します。

```
# rpcinfo -u localhost nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
# ps -ef | grep nfsd
root    232      1  0  Apr 07     ?     0:01 /usr/lib/nfs/nfsd -a 16
root   3127   2462  1  09:32:57  pts/3  0:00 grep nfsd
```

rpcinfo の –t オプションを使用し、TCP 接続を確認します。エラーになる場合は、NFS サービスを再起動します (652ページの「NFS サービスを再起動する方法」 を参照)。

6. 次のコマンドを入力し、mountd デーモンが動作していることを確認します。

```
# /usr/bin/rpcinfo -u localhost mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
# ps -ef | grep mountd
root    145      1 0 Apr 07  ?     21:57 /usr/lib/autofs/automountd
root    234      1 0 Apr 07  ?     0:04  /usr/lib/nfs/mountd
```

(続く)

```
root   3084 2462 1 09:30:20 pts/3  0:00  grep mountd
```

rpcinfo に –t オプションを指定し、TCP 接続も確認します。エラーになる場合は、NFS サービスを再起動します (652ページの「NFS サービスを再起動する方法」 を参照)。

7. 次のコマンドを入力し、rpcbind デーモンが動作していることを確認します。

```
# /usr/bin/rpcinfo -u localhost rpcbind
program 100000 version 1 ready and waiting
program 100000 version 2 ready and waiting
program 100000 version 3 ready and waiting
```

rpcbind がハングしている場合は、サーバーをリブートするか、653ページの「rpcbind をウォームスタートする方法」 に示す作業を行なってください。

▼ NFS サービスを再起動する方法

1. スーパーユーザーになります。

2. リブートせずにデーモンを有効にするために、次のコマンドを入力します。

```
# /etc/init.d/nfs.server stop
# /etc/init.d/nfs.server start
```

/etc/dfs/dfstab に項目がある場合、デーモンは停止してから再起動します。

▼ rpcbind をウォームスタートする方法

何らかのプログラムが動作しているために NFS サーバーをリブートできない場合は、次のようにウォームスタートすることで、RPC を使用するすべてのサービスを再起動せずに rpcbind を再起動できます。

1. スーパーユーザーになります。

2. rpcbind の **PID** を決定します。

   ps を実行すると、PID の値が第 2 カラムに表示されます。

```
# ps -ef |grep rpcbind
    root   115     1  0   May 31 ?        0:14 /usr/sbin/rpcbind
    root 13000  6944  0 11:11:15 pts/3    0:00 grep rpcbind
```

3. **SIGTERM** シグナルを rpcbind プロセスに送ります。

   以下の例では、送信するシグナルは term で、プログラムの PID は 115 です (kill(1) のマニュアルページを参照)。これにより、rpcbind は /tmp/portmap.file と /tmp/rpcbind.file に現在登録されているサービスのリストを作成します。

```
# kill -s term 115
```

注 - –s term オプションを使用して rpcbind プロセスを終了させないと、rpcbind のウォームスタートを完了できません。その場合は、サーバーを再起動することによってサービスを再開する必要があります。

---

4. rpcbind を再起動します。

   rpcbind のウォームスタートを実行すると、kill コマンドの実行で作成されたファイルが参照されるので、すべての RPC サービスを再起動せずにプロセスを再起動できます (rpcbind(1M) のマニュアルページを参照)。

```
# /usr/sbin/rpcbind -w
```

▼ NFS ファイルサービスを提供しているホストを識別する方法

–m オプションを指定して nfsstat コマンドを実行し、最新の NFS 情報を取得します。現在のサーバー名は、「currserver=」の後に表示されます。

```
% nfsstat -m
/usr/local from bee,wasp:/export/share/local
 Flags: vers=3,proto=tcp,sec=sys,hard,intr,llock,link,synlink,
  acl,rsize=32768,wsize=32678,retrans=5
 Failover: noresponse=0, failover=0, remap=0, currserver=bee
```

▼ mount コマンドに使用されたオプションを確認する方法

Solaris 2.6 およびそれ以降に出たパッチに置き換えられた mount コマンドでは、無効なオプションを指定しても警告されません。コマンド行に入力したオプション、または /etc/vfstab から指定したオプションが有効であるかどうかを判断するには、以下の手順を実行します。

たとえば、次のコマンドが実行されたとします。

```
# mount -F nfs -o ro,vers=2 bee:/export/share/local /mnt
```

1. 次のコマンドを実行し、オプションを確認します。

```
% nfsstat -m
/mnt from bee:/export/share/local
Flags:  vers=2,proto=tcp,sec=sys,hard,intr,dynamic,acl,rsize=8192,wsize=8192,
        retrans=5
```

bee からマウントされたファイルシステムは、プロトコルのバージョンが 2 に設定されています。nfsstat コマンドを使用しても、一部のオプションの情報は表示されませんが、オプションを確認するにはこれが最も正確な方法です。

2. /etc/mnttab でエントリを確認します。

mount コマンドでは、無効なオプションはマウントテーブルに追加されません。したがって、実行したコマンド行のオプションと /etc/mnttab にある当

該オプションを比較すれば、nfsstat コマンドによってレポートされないオプションがわかります。

```
# grep bee /etc/mnttab
bee:/export/share/local /mnt nfs ro,vers=2,dev=2b0005e 859934818
```

autofs の障害追跡

autofs の使用時、問題に遭遇することがあります。この節では、問題解決プロセスについてわかりやすく説明します。この節は、2 つの項目に分類されます。

この節では、autofs が生成するエラーメッセージのリストを示します。このリストは、2 つのパートに分かれています。

- automout の詳細形式 (–v) オプションにより生成されるエラーメッセージ
- 通常表示されるエラーメッセージ

各エラーメッセージのあとには、そのメッセージの説明と考えられる原因が続きます。

障害追跡時には、詳細形式 (–v) オプションで autofs プログラムを開始します。そうしないと、理由がわからないまま問題に遭遇することになります。

次の節は、autofs のエラー時に表示されがちなエラーメッセージと、生じうる問題についての説明です。

automount -v により生成されるエラーメッセージ

bad key *key* in direct map *mapname*

直接マップのスキャン中、autofs が接頭辞 / のないエントリーキーを発見しました。直接マップ内のキーは、フルパス名でなくてはなりません。

bad key *key* in indirect map *mapname*

間接マップのスキャン中、autofs が / を含むエントリキーを発見しました。間接マップのキーは、パス名ではなく、単なる名称でなくてはなりません。

`can't mount` *server*`:`*pathname: reason*

サーバー上のマウントデーモンが、*server:pathname* のファイルハンドルの提供を拒みました。サーバー上のエクスポートテーブルを確認してください。

`couldn't create mount point` *mountpoint*`:` *reason*

autofs は、マウントに必要なマウントポイントを作成することができませんでした。この症状は、すべてのサーバーのエクスポートされたファイルシステムを階層的にマウントしようとする場合に頻繁に生じます。必要なマウントポイントは、マウントできないファイルシステム内にだけ存在し (エクスポートは不可)、エクスポートされる親ファイルシステムは読み取り専用でエクスポートされるため、必要なマウントポイントが作成できないことになります。

`leading space in map entry` *entry* `text in` *mapname*

autofs はオートマウントマップ内に先頭にスペースを含むエントリを発見しました。通常これは、たとえば次の例のように、適切に継続されていないマップエントリの記述がある場合です。

```
fake
/blat     frobz:/usr/frotz
```

この例では、autofs が 2 つ目の行に遭遇した場合に警告が生成されます。これは、最初の行がバックスラッシュ (\) で終端されていないためです。

mapname`: Not found`

必要とされるマップが配置されていません。このメッセージは、`-v` オプションが使用されている場合にだけ生成されます。マップ名のスペルとパス名を確認してください。

`remount` *server*`:`*pathname* `on` *mountpoint*`: server not responding`

autofs が、アンマウントしたファイルシステムの再マウントに失敗しました。

`WARNING:` *mountpoint* `already mounted on`

autofs が、既存のマウントポイント上にマウントしようとしました。これは、autofs 内で内部エラー (異常) が生じたことを意味しています。

その他のエラーメッセージ

`dir` *mountpoint* `must start with '/'`

オートマウンタのマウントポイントは、フルパス名で与えられなくてはなりません。マウントポイントのスペルとパス名を確認してください。

`hierarchical mountpoints:` *pathname1* `and` *pathname2*

autofs は、マウントポイントが階層的な関係を持つことを許可しません。autofs マウントポイントは、他のオートマウントされたファイルシステムに含まれていてはなりません。

`host` *server* `not responding`

autofs が、*server* で示されるサーバーにコンタクトしようとしましたが、応答がありません。

hostname`: exports:` *rpc_err*

hostname で示される場所からエクスポートリストを得たことを示すエラーです。これは、サーバーまたはネットワークの問題を示します。

`map` *mapname*`, key` *key*`: bad`

マップエントリが不適切な形式であり、autofs が処理できません。そのエントリを再確認してください。そのエントリにエスケープする必要のある文字が含まれている可能性があります。

mapname`:` *nis_err*

NIS マップ内のエントリの参照におけるエラーです。NIS に問題のある可能性があります。

`mount of` *server*`:`*pathname* `on` *mountpoint:reason*

autofs がマウントに失敗しました。サーバーまたはネットワークに問題のある可能性があります。

mountpoint: Not a directory

autofs は、ディレクトリではない *mountpoint* に示される場所に自分自身をマウントすることができません。マウントポイントのスペルとパス名を確認してください。

nfscast: cannot send packet: *reason*

autofs が、複製されたファイルシステムの位置を示すリスト内にあるサーバーへの照会パケットを送信できません。

nfscast: cannot receive reply: *reason*

autofs が、複製されたファイルシステムの位置を示すリスト内にあるいずれのサーバーからも応答を受けられません。

nfscast: select: *reason*

このようなエラーメッセージはすべて、複製されたファイルシステムのサーバーの ping に関する問題を示します。ネットワークの問題を示している場合もあります。

pathconf: no info for *server*:*pathname*

autofs が、パス名に関する pathconf 情報の取得に失敗しました。(fpathconf(2) のマニュアルページを参照。)

pathconf: *server*: server not responding

autofs が、pathconf(2) に情報を提供する *server* に示されるサーバー上のマウントデーモンにコンタクトできませんでした。

autofs のその他のエラー

/etc/auto* ファイルが実行ビットセットを持っている場合、オートマウンタは次のようなメッセージを生成するマップの実行を試みます。

/etc/auto_home: +auto_home: not found

この場合、auto_home ファイルは不適切な権限をもつことになります。このファイル内の各エントリは、非常によく似たエラーメッセージを生成します。ファイルへのこのような権限は、次のコマンドを入力することにより取り消す必要があります。

```
# chmod 644 /etc/auto_home
```

NFS のエラーメッセージ

ここでは、エラーメッセージ、そのエラーの原因となる条件、およびその問題を解決する方法を少なくとも 1 つ示します。

`Bad argument specified with index option - must be a file`

–index オプションにはファイル名を指定する必要があります。ディレクトリ名は使えません。

`Cannot establish NFS service over /dev/`*tcp*`: transport setup problem`

このメッセージは、名前空間の中のサービス情報が更新されなかったときによく発生します。UDP に関して報告されることもあります。この問題を解決するには、名前空間の中のサービスデータを更新します。NIS+ の場合、エントリは以下のとおりです。

```
nfsd nfsd tcp 2049 NFS server daemon
nfsd nfsd ucp 2049 NFS server daemon
```

NIS と /etc/services の場合、エントリは以下のとおりです。

```
nfsd    2049/tcp    nfs    # NFS server daemon
nfsd    2049/ucp    nfs    # NFS server daemon
```

`Cannot use index option without public option`

share コマンドに public オプションを指定してください。–index オプションを使用するには、公開ファイルハンドルを定義する必要があります。

注 - Solaris 2.6 より前の Solaris では、`share` コマンドを使用して公共ファイルハンドルを設定する必要があります。Solaris 2.6 以降では、公共ファイルハンドルはデフォルトで / に設定されるため、このエラーメッセージは出力されません。

`Could not use public filehandle in request to` *server*

このメッセージは、public オプションが指定されているにもかかわらず NFS サーバーが公共ファイルハンドルをサポートしていない場合に表示されます。この場合、マウントは失敗します。この問題を解決するには、公共ファイルハンドルを使用せずにマウント要求を行うか、NFS サーバーが公共ファイルハンドルをサポートするように再設定します。

`NOTICE: NFS3: failing over from` *host1* `to` *host2*

このメッセージは、フェイルオーバーが発生するとコンソールに表示されます。報告のためだけのメッセージです。

filename`: File too large`

NFS バージョン 2 クライアントが、2G バイトを超えるサイズのファイルにアクセスしようとしています。

`mount: ... server not responding:RPC_PMAP_FAILURE - RPC_TIMED_OUT`

実行レベルの誤りか、`rpcbind` の停止かハングのため、マウント先のファイルシステムを共有しているサーバーがダウンしているか、またはソフトウェア的に接続されていないことを示すメッセージです。

`mount: ... server not responding: RPC_PROG_NOT_REGISTERED`

マウントが `rpcbind` によって登録されているにもかかわらず、NFS マウントデーモン (`mountd`) が登録されていないことを示すメッセージです。

`mount: ... No such file or directory`

リモートディレクトリかローカルディレクトリが存在しないことを示すメッセージです。ディレクトリ名のスペルをチェックするか、リモートディレクトリとローカルディレクトリの両方で `ls` コマンドを実行してください。

`mount: ...: Permission denied`

コンピュータ名が、クライアントのリストに載っていないか、マウントするファイルシステムにアクセスできるネットグループに含まれていないことを示すメッセージです。showmount -e を実行し、アクセスリストを確認してください。

`nfs mount: ignoring invalid option "-option"`

-option フラグが無効です。正しい構文を mount_nfs(1M) のマニュアルページで確認してください。

注 - このエラーメッセージは、Solaris 2.6 以降またはそれ以前のバージョンにパッチを適用した状態で mount コマンドを実行したときには表示されません。

`nfs mount: NFS can't support "nolargefiles"`

NFS クライアントが、–nolargefiles オプションを使用して NFS サーバーからファイルシステムをマウントしようとしました。このオプションは、NFS ファイルシステムに対してはサポートされていません。

`nfs mount: NFS V2 can't support "largefiles"`

NFS バージョン 2 プロトコルでは、大型ファイルを処理できません。大型ファイルを扱う必要がある場合は、バージョン 3 を使用してください。

`NFS server` *hostname* `not responding still trying`

ファイル関連の作業中にプログラムがハングすると、NFS サーバーは停止します。このメッセージは、NFS サーバー (*hostname*) がダウンしているか、サーバーかネットワークに問題があることを示すものです。フェイルオーバー機能を使用している場合、*hostname* はサーバー名のリストになります。647ページの「NFS クライアントの接続性を確認する方法」 を参照してください。

`NFS fsstat failed for server` *hostname*`: RPC: Authentication error`

様々な状況で発生するエラーです。最もデバッグが困難なのは、ユーザーの属しているグループが多すぎる場合です。現在、ユーザーは最大 16 個のグループに属すことができますが、NFS マウントでファイルにアクセスしている場合は、それ以下になります。NFS サーバーと NFS クライアントで Solaris 2.5 以降が動作している場合に、ユーザーが 16 以上のグループに属さなければならない場合は、ACL を使用することで必要なアクセス権を獲得できます。

`port` *number* `in nfs URL not the same as port` *number* `in port option`

NFS URL のポート番号は、マウントの –port オプションのポート番号と一致していなければなりません。一致していないと、マウントは失敗します。同じポート番号にしてコマンドを再実行するか、ポート番号の指定を省略してください。原則として、NFS URL と –port オプションの両方にポート番号を指定しても意味がありません。

`replicas must have the same version`

NFS フェイルオーバー機能が正しく機能するためには、複製の NFS サーバーが同じバージョンの NFS プロトコルをサポートしていなければなりません。バージョン 2 とバージョン 3 のサーバーが混在することは許されません。

`replicated mounts must be read-only`

NFS フェイルオーバー機能は、読み書き可能としてマウントされたファイルシステムでは動作しません。ファイルシステムを読み書き可能としてマウントすると、ファイルが変更される可能性が高くなるためです。NFS のフェイルオーバー機能は、ファイルシステムがまったく同じであることが前提です。

`replicated mounts must not be soft`

複製されるマウントの場合、フェイルオーバーが発生するまでタイムアウトを待つ必要があります。soft オプションを指定すると、タイムアウトが開始してすぐにマウントが失敗するため、複製されるマウントには –soft オプションは指定できません。

`share_nfs: Cannot share more than one filesystem with 'public' option`

/etc/dfs/dfstab ファイルを調べて、–public オプションによって共有するファイルシステムを複数選択していないか確認してください。公開ファイルハンドルは、サーバーあたり 1 つしか設定できません。したがって、–public オプションで共有できるファイルシステムは 1 つだけです。

`WARNING: No network locking on` *hostname*`:`*path*`: contact admin to install server change`

NFS クライアントが、NFS サーバー上のネットワークロックマネージャと接続を確立できませんでした。この警告は、マウントできなかったことを知らせるためではなく、ロックが機能しないことを警告するために出力されます。

第 31 章

リモートファイルシステムリファレンスへのアクセス

この章では、NFS コマンドの概要について説明します。また、NFS 環境のすべての構成要素とそれらが互いにどのように連携するかについても説明します。

NFS ファイル

いくつかのファイルでは、いずれのコンピュータ上でも NFS アクティビティをサポートする必要があります。それらの多くは ASCII ファイルで、いくつかはデータファイルです。表 31–1 に、このようなファイルとその機能をまとめます。

表 31–1 NFS ファイル

ファイル名	機能
/etc/default/fs	ローカルファイルシステムにおけるデフォルトファイルシステムのタイプを一覧表示する
/etc/dfs/dfstab	共有するローカルリソースを一覧表示する
/etc/dfs/fstypes	リモートファイルシステムにおけるデフォルトファイルシステムのタイプを一覧表示する
/etc/default/nfslogd	NFS サーバーログデーモン、nfslogd の構成情報を示す
/etc/dfs/sharetab	共有されるローカルとリモートのリソースを一覧表示する (sharetab(4) のマニュアルページを参照)。編集しないこと
/etc/mnttab	自動的にマウントしたディレクトリを含む、現在マウントしているファイルシステムを一覧表示する (mnttab(4) のマニュアルページを参照)。編集しないこと
/etc/netconfig	トランスポートプロトコルのリスト。編集しないこと
/etc/nfs/nfslog.conf	NFS サーバー記録のための一般的な構成情報を示す
/etc/nfs/nfslogtab	nfslogd によるログ後処理のための情報を示す。このファイルは編集しないこと
/etc/nfssec.conf	NFS のセキュリティサービスのリスト。編集しないこと
/etc/rmtab	NFS クライアントがリモートにマウントしたファイルシステムを一覧表示する (rmtab(4) のマニュアルページを参照)。 編集しないこと
/etc/vfstab	ローカルにマウントするファイルシステムを定義する (vfstab(4) のマニュアルページを参照)

/etc/dfs/fstypes の最初の項目は、リモートファイルシステムにおけるデフォルトファイルシステムのタイプとして利用されることがしばしばあります。この項目は、NFS ファイルシステムのタイプをデフォルトとして定義します。

/etx/default/fs には、項目が 1 つしかありません。ローカルディスクにおけるデフォルトファイルシステムのタイプです。クライアントやサーバーでサポートす

るファイルシステムのタイプは、/kernel/fs のファイルを確認して決定することができます。

/etc/default/nfslogd

このファイルは、NFS サーバーログ機能を使用するときに使用されるいくつかのパラメータを定義します。次のパラメータを定義することができます。

CYCLE_FREQUENCY

ログファイルを元の状態に戻す前に経過させる必要がある時間数を決めるパラメータです。デフォルト値は 24 時間です。このパラメータはログファイルが大きくなり過ぎないように使用します。

IDLE_TIME

nfslogd が、バッファーファイル内にさらに情報があるかどうかを確認するまでに休眠しなければならない秒数を設定するパラメータです。このパラメータは、構成ファイルの確認頻度も決定します。このパラメータと MAN_PROCESSING_SIZE によりバッファーファイルの処理頻度が決まります。デフォルト値は 300 秒です。この数値を増加させると、確認の回数が減って性能が向上します。

MAPPING_UPDATE_INTERVAL

ファイルハンドルパスマッピングテーブル内でレコードを更新する間隔を秒数で指定します。デフォルト値は 86400 秒 つまり 1 日です。このパラメータを使用すると、ファイルハンドルパスマッピングテーブルを常時更新せずに最新の状態に保つことができます。

MAX_LOG_PRESERVE

保存するログファイル数を決めます。デフォルト値は 10 です。

MAN_PROCESSING_SIZE

バッファーファイルが処理してログファイルに書き込むための最小限のバイト数を設定します。このパラメータと IDLE_TIME によりバッファーファイルの処理頻度が決まります。このパラメータのデフォルト値は 524288 バイトです。この数値を大きくするとバッファーファイルの処理回数が減って性能を向上できます。

PRUNE_TIMEOUT

ファイルハンドルパスマッピングレコードを中断して除去できるようになるまでに経過しなければならない時間数を選択するパラメータです。デフォルト値は 168 時間、つまり 7 日間です。

UMASK

`nfslogd` が作成するログファイルに対するアクセス権を指定します。デフォルト値は 0137 です。

/etc/nfs/nfslog.conf

このファイルは `nfslogd` で使用するログのパス、ファイル名、およびタイプを定義します。各定義はタグと関連づけられています。NFS サーバーのログを開始するためには、各ファイルシステムについてタグを付ける必要があります。広域タグはデフォルト値を定義します。各タグには必要に応じて次のパラメータを使用することができます。

defaultdir=*path*

ログファイルのデフォルトのディレクトリパスを指定するパラメータです。

log=*path/filename*

ログファイルのパスとファイル名を指定するパラメータです。

fhtable=*path/filename*

ファイルハンドルパスデータベースのパスとファイル名を選択するパラメータです。

buffer=*path/filename*

バッファーファイルのパスとファイル名を決定するパラメータです。

logformat=*basic*|*extended*

ユーザーから読み取り可能なログファイルを作るときに使用するフォーマットを選択します。基本フォーマットは、`ftpd` デーモンと同様なログファイルが作成されます。拡張フォーマットは、より詳細に表示されます。

パスとファイル名の両方を指定することができるパラメータについては、パスが指定されていない場合は、defaultdir が定義するパスが使用されます。絶対パスを使用すると defaultdir を無効にすることができます。

ファイルを識別しやすくするために、ファイルを別々のディレクトリに入れておきます。次に、必要な変更の例を示します。

```
% cat /etc/nfs/nfslog.conf
#ident  "@(#)nfslog.conf        1.5     99/02/21 SMI"
#
  .
  .
# NFS server log configuration file.
#

global  defaultdir=/var/nfs \
        log=nfslog fhtable=fhtable buffer=nfslog_workbuffer

publicftp log=logs/nfslog fhtable=fh/fhtables buffer=buffers/workbuffer
```

この例では、 log=publicftp と共有するファイルはすべて、次の値を使用します。デフォルトのディレクトリは /var/nfs になり、ログファイルは /var/nfs/logs/nfslog* に保存され、ファイルハンドルパスデータベーステーブルは /var/nfs/fh/fhtables に保存され、バッファーファイルは /var/nfs/buffers/workbuffer に保存されます。

NFS デーモン

NFS アクティビティをサポートするには、システムが実行レベル 3 かマルチユーザーモードで動作したときに、いくつかのデーモンを開始します。mountd と nfsd の 2 つのデーモンは、NFS サーバーであるシステム上で動作します。サーバーデーモンの自動起動は、NFS ファイルシステムのタイプでラベル付けされた項目が /etc/dfs/sharetab に存在するかどうかで変わります。

lockd と statd の 2 つのデーモンは NFS クライアントで動作し、NFS ファイルロッキングをサポートします。NFS サーバーでも動作させなければなりません。

automountd

このデーモンは autofs サービスからのマウントおよびデマウント要求を処理します。このコマンドの構文は次の通りです。

```
automountd [ –Tnv ] [ -D name=value ]
```

–T は標準出力への各 RPC 呼び出しの表示を選択し、–n はすべての autofs ノードでのブラウズを無効にし、–v はコンソールへのすべての状態メッセージのレコードを選択し、–D *name=value* は、*name* により示された自動マウント変数を *value* に置き換えます。自動マウントマップのデフォルト値は /etc/auto_master です。障害追跡には –T オプションを使用してください。

lockd

このデーモンは NFS ファイルのレコードロックをサポートします。ロック要求をクライアントから NFS サーバーに送り、NFS サーバーでローカルのロックを開始します。通常は、パラメータを指定せずに起動します。使用できるオプションは 3 つあります (lockd(1M) のマニュアルページを参照してください)。

–g *graceperiod* オプションは、サーバーがリブートした場合に、その何秒後にロックを再要求するかを示します。NFS サーバーはこの秒数の間、それまでのロックの再要求処理しか実行しません。他のサービスに対する要求は、この時間が経過するまで待たされます。このオプションは NFS サーバーの応答性に関係するため、NFS サーバーでしか変更できません。デフォルト値は 45 秒です。この値を小さくすると、サーバーをリブートしてからオペレーションに復帰するまでの時間は短縮されますが、クライアントがすべてのロックを復旧できなくなる可能性が増します。

–t *timeout* オプションは、ロック要求をリモートサーバーに再送信するまで何秒待つかを示します。このオプションは NFS クライアントのサービスに関係します。デフォルト値は 15 秒です。この値を小さくすると、雑音の多いネットワーク上の NFS クライアントに対する応答時間を改善できますが、ロック要求が増えることによってサーバーの負荷が増す可能性があります。

nthreads オプションは、サーバーが 1 つの接続について同時に処理できるスレッドの数の上限を示します。この値は、NFS サーバーに対して予想される負荷に基づいて決定してください。デフォルト値は 20 です。TCP を使用する NFS クライアントはそれぞれ NFS サーバーと 1 つの接続を設定するため、各 TCP クライアントはサーバー上で同時に 20 までのスレッドを使用することが許されます。UDP (ユーザーデーモンプロトコル) を使用する NFS クライアントは、すべてが NFS サーバーと 1 つの接続を共有します。その場合、UDP 接続が使用できるスレッドの数を増やさな

ければならないことがあるかもしれません。簡単な目安は 1 つの UDP クライアントにつき 2 つのスレッドですが、クライアントに対する作業負荷によってはこれで不十分なこともあります。使用するスレッドを増やすことによるマイナスは、スレッドの使用によって NFS サーバーで使用されるメモリーが増えることです。しかしスレッドが使用されないならば、*nthreads* を大きくしても影響はありません。

mountd

これは、リモートシステムからのファイルシステムマウント要求を処理して、アクセス制御を行う RPC (リモートプロシージャコール) サーバーです。`/etc/dfs/sharetab` を調べることによって、リモートマウントに使用可能なファイルシステムと、リモートマウントを実行可能なシステムを判断します。`–v` と `–r` の 2 つのオプションが使えます (`mountd(1M)` のマニュアルページを参照してください)。

`–v` オプションは、コマンドを詳細形式モードで実行します。クライアントが取得すべきアクセス権を NFS サーバーが決定するたびに、コンソールにメッセージが表示されます。この情報は、クライアントがファイルシステムにアクセスできない理由を調べるときに役立ちます。

`–r` オプションは、その後のクライアントからのマウント要求をすべて拒絶します。すでにファイルシステムがマウントされているクライアントには影響しません。

nfsd

これは、他のクライアントからのファイルシステム要求を処理するデーモンです。このコマンドに対してはいくつかのオプションが指定できます。オプションをすべて確認するには `nfsd(1M)` のマニュアルページを参照してください。

`–l` オプションは、接続指向トランスポートでの NFS/TCP に対する接続キューの長さを設定します。デフォルト値は 25 エントリです。

`–c` #*_conn* オプションは、接続指向トランスポート 1 つあたりの接続数の上限を選択します。デフォルト値はありません。

nservers オプションは、1 台のサーバーが同時に処理可能な要求の数の上限です。デフォルト値は 1 ですが、起動スクリプトでは 16 が選択されます。

このデーモンの以前のバージョンとは異なり、このバージョンの `nfsd` では複数のコピーを作成して要求を同時に処理することはありません。処理テーブルを `ps` でチェックすると、動作しているデーモンのコピーが 1 つしかないことがわかります。

nfslogd

このデーモンは操作関係のログ機能を提供します。サーバーに対する NFS 操作は、/etc/default/nfslogd で定義される構成オプションに基づいて記録されます。NFS サーバーのログ機能がオンになると、選択されたファイルシステム上でのすべての RPC 操作の記録がカーネルによりバッファーファイルに書き込まれます。次に nfslogd がこれらの要求を後処理します。ログインおよび IP アドレスへの UID をホスト名に割り当てやすくするために、ネームサービススイッチが使用されます。識別されたネームサービスで一致するものが見つからない場合は、その番号が記録されます。

パス名へのファイルハンドルの割り当ても nfslogd により行われます。このデーモンは、ファイルハンドルパスマッピングテーブル内でこれらの割り当てを追跡します。/etc/nfs/nfslogd において識別される各タグについて 1 つのマッピングテーブルが存在します。後処理のあとに、レコードが ASCII ログファイルに書き込まれます。

statd

lockd とともに動作し、ロック管理機能にクラッシュ機能と回復機能を提供します。NFS サーバーでロックを保持しているクライアントの追跡を行い、サーバーがクラッシュし、リブートしている間に、サーバー側 statd がクライアント側 statd と連絡をとります。次にクライアント側 statd は、サーバー上のすべてのロックを再要求します。クライアントがクラッシュすると、クライアント側 statd はサーバー側 statd にそのことを伝えるので、サーバー上のロックはクリアされます。このデーモンにオプションはありません。詳細は、statd(1M) のマニュアルページを参照してください。

Solaris 7 では、statd がクライアントを追跡する方法が改善されました。Solaris 7 より前のリリースの statd では、クライアントごとにそのクライアントの修飾されていないホスト名を使用して、/var/statmon/sm にファイルが作成されました。そのため、同じホスト名の 2 つのクライアントが異なるドメインに存在する場合や、クライアントが NFS サーバーと異なるドメインに存在する場合に問題が発生していました。修飾されていないホスト名にはドメインや IP アドレスの情報がないため、このようなクライアントを区別する方法がありませんでした。これに対処するため、Solaris 7 の statd では、修飾されていないホスト名に対してクライアントの IP アドレスを使用して /var/statmon/sm にシンボリックリンクを作成します。このリンクは、次のような形式です。

```
# ls -l /var/statmon/sm
lrwxrwxrwx   1 root          11 Apr 29 16:32 ipv4.192.9.200.1 -> myhost
--w-------   1 root          11 Apr 29 16:32 myhost
```

この例では、クライアントのホスト名は myhost で、IP アドレスは 192.9.200.1 です。他のホストが myhost という名前を持ち、ファイルシステムをマウントしていると、myhost というホスト名に対するシンボリックリンクは 2 つ作成されます。

NFS コマンド

次のコマンドは、root 権限で実行しなければ、十分な効果がでません。しかし情報の要求は、すべてのユーザーが行えます。

- 671ページの「automount」
- 672ページの「clear_locks」
- 673ページの「mount」
- 678ページの「mountall」
- 687ページの「setmnt」
- 679ページの「share」
- 685ページの「shareall」
- 686ページの「showmount」
- 677ページの「umount」
- 678ページの「umountall」
- 685ページの「unshare」
- 686ページの「unshareall」

automount

このコマンドは autofs マウントポイントをインストールし、オートマスターファイル内の情報を各マウントポイントに関連づけます。このコマンドの構文は次の通りです。

```
automount [ –t duration ] [ –v ]
```

–t *duration* はファイルシステムがマウントされた状態でいる時間 (秒) で、–v は詳細形式モードを選択します。詳細形式モードでこのコマンドを実行すると障害追跡が容易になります。

継続時間の値は、特に設定しないと 5 分に設定されます。ほとんどの場合この時間は適切な値ですが、自動マウントされたファイルシステムを多く持つシステムではこの時間を長くする必要がある場合があります。特に、サーバーを多くのユーザーが使用中の場合は、自動マウントされたファイルシステムを 5 分ごとにチェックするのは能率的でない場合があります。autofs ファイルシステムは 1800 秒 (30 分) ごとにチェックする方が適しています。このファイルシステムを 5 分ごとにアンマウントしないことで、df によりチェックされる /etc/mnttab が大きくなる可能性があります。–F オプション (df(1M) のマニュアルページを参照)、または egrep を使用して df からの出力にフィルタをかけて、この問題を解決することができます。

検討すべき他の要因に、この継続時間を調節するとオートマウンタマップへの変更が反映される速さを変えられることがあります。変更はファイルシステムがアンマウントされるまでは見ることができません。オートマウンタマップの変更方法については、631ページの「マップの修正」を参照してください。

clear_locks

このコマンドを使用すると、ある NFS クライアントのファイル、レコード、または共有のロックをすべて削除できます。このコマンドを実行するには、スーパーユーザーでなければなりません。NFS サーバーから解除できるのは特定のクライアントのロックであり、NFS クライアントから解除できるのは特定のサーバー上のそのクライアントに対するロックです。次の例では、現在のシステム上の tulip という NFS クライアントに対するロックが解除されます。

```
# clear_locks tulip
```

–s オプションを指定すると、どの NFS ホストからロックを解除するかを指定できます。これは、そのロックをかけた NFS クライアントから実行しなければなりません。次の場合、クライアントによるロックが bee という名前の NFS サーバーから解除されます。

```
# clear_locks -s bee
```

注意 - このコマンドは、クライアントがクラッシュしてロックを解除できないとき以外には使用しないでください。データが破壊されるのを避けるため、使用中のクライアントに関するロックは解除しないでください。

mount

このコマンドを使用すると、指定したファイルシステムをローカルかリモートで、指定したマウントポイントに添付できます。詳細は、mount(1M) のマニュアルページを参照してください。引数を指定しないと、現在ユーザーのコンピュータにマウントされているファイルシステムのリストが表示されます。

Solaris の標準インストールには、さまざまな種類のファイルシステムが含まれています。ファイルシステムの種類ごとにマニュアルページがあり、その種類に対して mount を実行するときに使用可能なオプションのリストが示されています。たとえば、NFS ファイルシステムは mount_nfs(1M) のマニュアルページ、UFS ファイルシステムは mount_ufs(1M) のマニュアルページなどです。

Solaris 7 では、server:/pathname という標準の構文の代わりに NFS URL を使用して NFS サーバー上のマウントするパス名を指定することが可能になりました。詳細は、619ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」を参照してください。

注意 - Solaris 2.6 以後の mount コマンドでは、無効なオプションがあっても警告されません。解釈できないオプションがあると無視されるだけです。予想外の結果が生じるのを避けるために、使用するオプションはすべて確認してください。

NFS ファイルシステムにおける mount オプション

NFS ファイルシステムをマウントするときに –o フラグの後に指定できるオプションの一部を以下に示します。

bg|fg

この 2 つは、マウントが失敗したときの再試行の方法を選択するオプションです。–bg オプションの場合はバックグラウンドで、–fg オプションの場合はフォアグラウンドでマウントが試みられます。デフォルトは –fg です。常に使用可能にしておく必要のあるファイルシステムに対しては –fg が適しています。この場合、

マウントが完了するまで他の処理は実行できません。–bg は、マウント要求が完了しなくてもクライアントは他の処理を実行できるため、必ずしも必要でないファイルシステムに適しています。

forcedirectio

このオプションは大型ファイル上で連続した読み取りをする際に性能を向上させます。データは直接ユーザーファイルにコピーされ、クライアント上のカーネル内ではキャッシュへの書き込みは行われません。この機能はデフォルトではオフです。

largefiles

このオプションを使用すると、Solaris 2.6 が実行されているサーバーに置かれた 2G バイトを超えるサイズのファイルにアクセスできるようになります。大型ファイルにアクセスできるかどうかは、サーバーでしか制御できません。したがって、このオプションは NFS バージョン 3 のマウントでは無視されます。デフォルトでは、2.6 以後の UFS ファイルシステムはすべて –largefiles オプション付きでマウントされます。NFS バージョン 2 プロトコルを使用したマウントでこのオプションを指定すると、エラーが発生してマウントできません。

nolargefiles

UFS マウントでこのオプションを指定すると、ファイルシステム上に大型ファイルが存在せず、この後も作成されないことが保証されます (mount_ufs(1M) のマニュアルページを参照してください)。大型ファイルが存在するかどうかは NFS サーバーでしか制御できないため、NFS マウントを使用する –nolargefiles にはオプションはありません。このオプションを指定してファイルシステムを NFS マウントしようとすると、エラーが発生して拒否されます。

public

このオプションを指定すると、NFS サーバーにアクセスするときに必ず公共ファイルハンドルを使用するようになります。NFS サーバーが公共ファイルハンドルをサポートしていれば、MOUNT プロトコルが使用されないため、マウント操作は短時間で行われます。また、MOUNT プロトコルを使用しないため、ファイアウォールを越えたマウントが可能です。

rw | ro

–rw オプションと –ro オプションは、ファイルシステムが読み書き可能と読み取り専用のどちらでマウントされるかを示します。デフォルトは読み書き可能で、これ

はリモートホームディレクトリやメールスプールディレクトリなどの、ユーザーによる変更が必要なファイルシステムに適しています。読み取り専用オプションは、ユーザーが変更してはいけないディレクトリに適しています。具体的には、マニュアルページの共有コピーなどです。

sec=*mode*

このオプションは、マウント時に使用される認証機構を指定します。*mode* の値は、表 31–2 に示したもののいずれかでなければなりません。モードは、`/etc/nfssec.conf` ファイルにも定義されます。

表 31–2 NFS セキュリティモード

モード	選択される認証サービス
`krb5`	Kerberos バージョン 5
`none`	認証なし
`dh`	Diffie-Hellman (DH) 認証
`sys`	UNIX の標準認証

soft | hard

`soft` オプションを指定してマウントされた NFS ファイルシステムは、サーバーが応答しなくなるとエラーを返します。`hard` オプションが指定されていると、サーバーが応答するまで再試行が続けられます。デフォルトは `hard` です。ほとんどのファイルシステムには `hard` を使用します。ソフトマウントされたファイルシステムからの値を検査しないアプリケーションが多いので、アプリケーションでエラーが発生してファイルが破壊される恐れがあるためです。検査するアプリケーションの場合でも、ルーティングの問題などによってアプリケーションが正しい判断をできずに、ファイルが破壊されることがあります。原則として、`soft` は使用しないでください。`hard` オプションを指定した場合にファイルシステムが使えなくなると、そのファイルシステムを使用するアプリケーションはファイルシステムが復旧するまでハングする可能性があります。

mount コマンドの使用

次のコマンドのどちらも、bee サーバーから NFS ファイルシステムを読み取り専用としてマウントします。

```
# mount -F nfs -r bee:/export/share/man /usr/man
```

```
# mount -F nfs -o ro bee:/export/share/man /usr/man
```

このコマンドでは –O オプションによって、/usr/man がすでにマウントされていても bee サーバーのマニュアルページがローカルシステムにマウントされます。

```
# mount -F nfs -O bee:/export/share/man /usr/man
```

このコマンドでは、クライアント側フェイルオーバー機能が使用されています。

```
# mount -F nfs -r bee,wasp:/export/share/man /usr/man
```

注 - コマンド行から使用する場合、リスト内のサーバーがサポートしている NFS プロトコルは同じバージョンでなければなりません。コマンド行から mount を実行するときは、バージョン 2 とバージョン 3 のサーバーを混在させないでください。autofs では混在が可能なので、バージョン 2 サーバーとバージョン 3 サーバーの最適な組み合わせを使用できます。

---

mount コマンドで NFS URL を使用する例を示します。

```
# mount -F nfs nfs://bee//export/share/man /usr/man
```

mount コマンドに引数を指定しないと、クライアントにマウントされたファイルシステムが表示されます。

```
% mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid on Tues Jan 24 13:20:47 1995
/usr on /dev/dsk/c0t3d0s6 read/write/setuid on Tues Jan 24 13:20:47 1995
/proc on /proc read/write/setuid on Tues Jan 24 13:20:47 1995
/dev/fd on fd read/write/setuid on Tues Jan 24 13:20:47 1995
/tmp on swap read/write on Tues Jan 24 13:20:51 1995
/opt on /dev/dsk/c0t3d0s5 setuid/read/write on Tues Jan 24 13:20:51 1995
/home/kathys on bee:/export/home/bee7/kathys
  intr/noquota/nosuid/remote on Tues Jan 24 13:22:13 1995
```

umount

このコマンドにより、現在マウントされているリモートファイルシステムが削除されます。umount コマンドは、テストのために –v オプションをサポートしています。また、–a オプションを使用することによって 1 度に複数のファイルシステムをアンマウントできます。–a オプションに *mount_points* を指定すると、そのファイルシステムがアンマウントされます。マウントポイントを指定しないと、/etc/mnttab のリストにあるファイルシステムのうち required でないものすべてのアンマウントが試みられます。required のファイルシステムとは、/、/usr、/var、/proc、/dev/fd、/tmp などです。

ファイルシステムがすでにマウントされていて、/etc/mnttab に項目が指定されている場合、ファイルシステムのタイプのフラグを指定する必要はありません。

ファイルシステムが使用中だと、このコマンドは実行できません。たとえば、あるユーザーが cd コマンドによってファイルシステムにアクセスしていると、作業ディレクトリが他に変更されるまでそのファイルシステムは使用中となります。umount コマンドは、NFS サーバーに接続できないと一時的にハングすることがあります。

umount コマンドの使用

次の例では、/usr/man にマウントしたファイルシステムのマウントが解除されます。

```
# umount /usr/man
```

次の例では、umount –a -v の実行結果が表示されます。

```
# umount -a -V
umount /home/kathys
umount /opt
umount /home
umount /net
```

このコマンドでは、ファイルシステムのアンマウント自体は実行されないことに注意してください。

mountall

このコマンドを使用すると、ファイルシステムテーブルに指定したすべてのファイルシステム、または特定グループのファイルシステムをマウントできます。アクセスするファイルシステムタイプを選択するための –F *FSType* オプション、ファイルシステムテーブル内のリモートファイルシステムをすべて選択する –r オプション、ローカルファイルシステムをすべて選択する –l オプションがあります。 NFS ファイルシステムタイプと指定されているファイルシステムはすべてリモートファイルシステムなので、これらのオプションは余分な指定になることがあります。詳細は、mountall(1M) のマニュアルページを参照してください。

mountall コマンドの使用

次の 2 つの例を実行すると、同じ結果になります。

```
# mountall -F nfs
```

```
# mountall -F nfs -r
```

umountall

このコマンドを使用すると、ファイルシステムのグループをアンマウントできます。–k オプションは、*mount_point* に結び付けられているプロセスを終了させるには fuser –k *mount_point* コマンドを使用する必要があることを表します。–s オプションは、アンマウントを並行処理しないことを示します。–l は、ローカルファイルシステムだけを使用することを、–r はリモートファイルシステムだけを使用することを示します。–h *host* オプションは、指定されたホストのファイルシステムをすべてアンマウントすることを指定します。–h オプションは、–l または –r とは同時に指定できません。

umountall コマンドの使用

次のコマンドでは、リモートホストからマウントしたすべてのファイルシステムが切り離されます。

```
# umountall -r
```

次のコマンドでは、bee サーバーからマウントしたすべてのファイルシステムが切り離されます。

```
# umountall -h bee
```

share

このコマンドを使用すると、NFS サーバーのローカルファイルシステムをマウントできるようになります。また、システム上のファイルシステムのうち、現在共有しているもののリストを表示します。NFS サーバーが動作していないと、share コマンドは使用できません。NFS サーバーソフトウェアは、/etc/dfs/dfstab に項目がある場合、ブートの途中で自動的に起動されます。NFS サーバーソフトウェアが動作していないくても、このコマンドはエラーを表示しません。NFS サーバーソフトウェアが動作していることを確認してからこのコマンドを使用するようにしてください。

ディレクトリツリーはすべて共有できるオブジェクトですが、各ファイルシステムの階層構造は、そのファイルシステムが位置するディスクスライスやパーティションで制限されます。たとえばルート (/) ファイルシステムを共有しても、/usr が同じディスクパーティションかスライスに存在しなければ、/usr を共有することはできません。通常、ルートはスライス 0 に、/usr はスライス 6 にインストールされます。また /usr を共有しても、/usr のサブディレクトリにマウントされているローカルディスクパーティションは共有できません。

すでに共有している大きいファイルシステムの一部であるファイルシステムを共有することはできません。たとえば /usr と /usr/local が同じディスクスライスにある場合、/usr も /usr/local も共有することができますが、両方を別々の共有オプションで共有する場合、/usr/local は別のディスクスライスに移動しなければなりません。

注 - 2 つのファイルシステムが同じディスクスライスにある場合、読み取り専用で共有しているファイルシステムに、読み取りと書き込みが可能な状態で共有しているファイルシステムのファイルハンドルでアクセスすることができます。読み取りと書き込みの両方を行うファイルシステムは、読み取り専用で共有する必要があるファイルシステムとは別のパーティションかディスクスライスに保存するほうが安全です。

非ファイルシステム用 share オプション

–o フラグに指定できるオプションの一部を次に示します。

rw | ro

pathname に指定したファイルシステムを、読み取りと書き込みの両方が可能な状態で共有するか、読み取り専用で共有するかを指定します。

rw=*accesslist*

ファイルシステムは、リスト上のクライアントに対してだけ読み書き可能で共有されます。それ以外の要求は拒否されます。*accesslist* に定義されるクライアントのリストは、Solaris 2.6 から拡張されました。詳細は、682ページの「share コマンドを使用してアクセスリストを設定する」 を参照してください。このオプションは –ro オプションよりも優先されます。

NFS 用 share オプション

NFS ファイルシステムで指定できるオプションは、次のとおりです。

aclok

このオプションを指定すると、NFS バージョン 2 プロトコルをサポートしている NFS サーバーが NFS バージョン 2 クライアントのアクセス制御を行うように設定できます。このオプションを指定しないと、すべてのクライアントは最低限のアクセスしかできません。指定すると、最大限のアクセスができるようになります。たとえば –aclok オプションを指定して共有したファイルシステムでは、1 人のユーザーが読み取り権を持っていれば全員が読み取りを許可されます。このオプションを指定しないと、アクセス権を持つべきクライアントからのアクセスが拒否される可能性があります。アクセス権の与えすぎと制限しすぎのどちらを選ぶかは、現在のセキュリティシステムによって決定します。アクセス制御リスト (ACL) について詳細は、『*Solaris* のシステム管理 *(*第 2 巻*)*』の「ファイルのセキュリティの適用手順」を参照してください。

注 - ACL を活用するためには、クライアントでもサーバーでも NFS バージョン 3 と NFS_ACL プロトコルをサポートしているソフトウェアを実行します。NFS バージョン 3 プロトコルしかサポートしていないソフトウェアの場合、クライアントは正しいアクセス権を取得できますが、ACL を操作することはできません。NFS_ACL プロトコルをサポートしていれば、正しいアクセス権を取得した上で ACL の操作も可能です。この両方をサポートしているのは、Solaris 2.5 およびその互換バージョンです。

---

anon=*uid*

uid は、認証されていないユーザーのユーザー ID を選択するために使用します。*uid* を -1 に設定すると、認証されていないユーザーからのアクセスは拒否されます。anon=0 とするとルートアクセス権を与えることができますが、これは認証されていないユーザーにルートアクセス権を与えることになるため、代わりに root オプションを使用してください。

index=*filename*

–index=*filename* オプションを使用すると、ユーザーが NFS URL にアクセスするとディレクトリのリストが表示されるのではなく、HTML (HyperText Markup Language) ファイルが強制的に読み込まれます。これは、HTTP URL がアクセスしているディレクトリに index.html ファイルが見つかるとブラウザのような動作をするというものです。このオプションを設定することは、httpd に対して DirectoryIndex オプションを指定するのと同じ意味があります。たとえば、dfstab ファイルのエントリが次のとおりであるとします。

```
share -F nfs -o ro,public,index=index.html /export/web
```

このとき、次の URL によって表示される情報はすべて同じです。

```
nfs://<server>/<dir>
nfs://<server>/<dir>/index.html
nfs://<server>//export/web/<dir>
nfs://<server>//export/web/<dir>/index.html
http://<server>/<dir>
http://<server>/<dir>/index.html
```

log=*tag*

このオプションは、ファイルシステム用の NFS サーバーレコード構成情報の入った /etc/nfs/nfslog.conf 内のタグを指定します。NFS サーバーログ機能を使用可能にするにはこのオプションを選択する必要があります。

nosuid

このオプションを使用すると、setuid モードまたは setgid モードを有効にしようとしても無視されます。NFS クライアントは、setuid か setgid のビットがオンの状態ではファイルを作成できません。

public

–public オプションは、WebNFS ブラウズのために追加されました。このオプションで共有できるのは、1 台のサーバーにつき 1 つのファイルシステムだけです。

root=*accesslist*

サーバーが、リスト上のホストに対してルートアクセス権を与えます。デフォルトでは、サーバーはどのリモートホストにもルートアクセス権は与えません。選択されているセキュリティモードが –sec=sys 以外だと、*accesslist* に指定できるホストはクライアントだけです。*accesslist* に定義されたクライアントのリストは、Solaris 2.6 で拡張されました。詳細については、682ページの「share コマンドを使用してアクセスリストを設定する」 を参照してください。

注意 - 他のホストにルートアクセス権を与えるには、広い範囲でセキュリティが保証されていることが前提です。–root= option は十分慎重に使用してください。

sec=*mode*[:*mode*]

mode は、ファイルシステムへのアクセス権を取得するために必要なセキュリティモードです。デフォルトのセキュリティモードは、UNIX の認証です。モードは複数指定できますが、コマンド行に指定するときは 1 行につき 1 つのセキュリティモードだけにしてください。–mode の各オプションは、次に –mode が出現するまでその後の –rw、–ro、–rw=、–ro=、–root=、–window= オプションに適用されます。–sec=none とすると、すべてのユーザーがユーザー nobody にマップされます。

window=*value*

value は、NFS サーバーで資格が有効な時間の上限です。デフォルトは 30000 秒 (8.3 時間) です。

share コマンドを使用してアクセスリストを設定する

リリース 2.6 より前の Solaris で、share コマンドの –ro=、–rw=、–root= オプションに指定する *accesslist* の内容は、ホスト名がネットグループ名に限定されていました。Solaris 2.6 以降では、このアクセス制御リストにドメイン名、サブネット番号、およびアクセス権を与えないエントリも指定できます。この拡張により、名

前空間を変更したり多数のクライアントを定義したリストを使用することなく、サーバーでのファイルアクセス制御を今までより簡単に管理できます。

次のコマンドでは、rose と lilac では読み取りと書き込みの両方のアクセスが認められますが、その他では、読み取りのみが許可されます。

```
# share -F nfs -o ro,rw=rose:lilac /usr/src
```

次の例では、eng ネットグループのすべてのホストで読み取りのみができるようになります。rose クライアントでは、読み取りと書き込みの両方ができます。

```
# share -F nfs -o ro=eng,rw=rose /usr/src
```

注 - rw と ro には必ず引数が必要です。読み書き可能オプションを指定しないと、デフォルトによってすべてのクライアントが読み書き可能になります。

---

1 つのファイルシステムを複数クライアントで共有するには、すべてのオプションを同じ行に指定しなければなりません。同じオブジェクトに share コマンドを複数回実行しても、最後のコマンドしか有効になりません。以下のコマンドでは、3 つのクライアントシステムで読み取りと書き込みができますが、rose と tulip では、ファイルシステムに root でアクセスできます。

```
# share -F nfs -o rw=rose:lilac:tulip,root=rose:tulip /usr/src
```

複数の認証機構を使用してファイルシステムを共有するときには、セキュリティモードの後に必ず –ro、–ro=、–rw、–rw=、–root、–window の各オプションを指定してください。この例では、eng というネットグループ内のすべてのホストに対して UNIX 認証が選択されています。これらのホストは、ファイルシステムを読み取り専用モードでしかマウントできません。ホスト tulip と lilac は、Diffie-Hellman (DH) 認証を使えば読み書き可能でファイル・システムをマウントできます。tulip と lilac は、そのホスト名が eng ネットグループのリストに含まれていれば、DH 認証を使用していなくても読み取り専用でマウントすることは可能です。

```
# share -F nfs -o sec=dh,rw=tulip:lilac,sec=sys,ro=eng /usr/src
```

UNIX 認証はデフォルトのセキュリティモードですが、–sec を指定するとデフォルトは無効になります。他の認証機構とともに UNIX 認証も使用する場合には、必ず –sec=sys オプションを指定してください。

実際のドメイン名の名前にドットを付けると、アクセスリストの中で DNS ドメイン名が使えます。ドットは、その後の文字列が完全に修飾されたホスト名ではなくドメイン名であることを表します。次のエントリは、マウントから eng.sun.com ドメイン内のすべてのホストへのアクセスを許可するためのものです。

```
# share -F nfs -o ro=.:.eng.sun.com /export/share/man
```

この例で、“.” はそれぞれ NIS または NIS+ 名前空間を通じて一致するすべてのホストに対応します。ネームサービスから返される結果にはドメイン名は含まれません。“.eng.sun.com” というエントリは、名前空間の解決に DNS を使用するすべてのホストに一致します。DNS が返すホスト名は必ず完全に修飾されるので、DNS と他の名前空間を組み合わせると長いエントリが必要です。

実際のネットワーク番号かネットワーク名の前に “@” を指定すると、アクセスリストの中でサブネット番号が使えます。これは、ネットワーク名をネットグループ名や完全に修飾されたホスト名と区別するためです。サブネットは、/etc/networks の中か NIS または NIS+ 名前空間の中で識別できなければなりません。次のエントリは、サブネット 129.144 が eng ネットワークと識別されるならばすべて同じ意味を持ちます。

```
# share -F nfs -o ro=@eng /export/share/man
# share -F nfs -o ro=@129.144 /export/share/man
# share -F nfs -o ro=@129.144.0.0 /export/share/man
```

2 番目と 3 番目のエントリは、ネットワークアドレス全体を指定する必要がないことを表しています。

ネットワークアドレスの先頭部分がバイトによる区切りでなく、CIDR (Classless Inter-Domain Routing) のようになっている場合には、マスクの長さをコマンド行で具体的に指定できます。この長さは、ネットワーク名かネットワーク番号の後ろにスラッシュで区切ってアドレスの接頭辞に有効ビット数として指定します。たとえば、次のようにします。

```
# share -f nfs -o ro=@eng/17 /export/share/man
# share -F nfs -o ro=@129.144.132/17 /export/share/man
```

この例で、“/17” はアドレスの先頭から 17 ビットがマスクとして使用されることを表します。CIDR について詳細は、RFC 1519 を参照してください。

また、エントリの前に “-” を指定することでアクセスの拒否を示すこともできます。エントリは左から右に読み込まれるため、アクセス拒否のエントリはそのエントリを適用するエントリの前に置く必要があることに注意してください。

```
# share -F nfs -o ro=-rose:.eng.sun.com /export/share/man
```

この例では、eng.sun.com ドメイン内のホストのうち、rose を除いたすべてに対してアクセス権が許可されます。

unshare

このコマンドを使用すると、以前に使用可能な状態になっていたファイルシステムを、クライアントがマウントできないようにします。unshare コマンドを使用すると、share コマンドで共有したファイルシステムや、/etc/dfs/dfstab で自動的に共有しているファイルシステムが共有できないようになります。unshare コマンドを使用し、dfstab ファイルで共有しているファイルシステムの共有を解除する場合は、実行レベル 3 を終了して再度実行レベル 3 に戻ると、そのファイルシステムがまた共有されることに注意してください。実行レベル 3 を終了しても変更内容を継続させるには、そのファイルシステムを dfstab ファイルから削除しなければなりません。

NFS ファイルシステムの共有を解除している場合、クライアントから既存マウントへのアクセスは禁止されます。クライアントにはファイルシステムがまだマウントされている可能性がありますが、ファイルにはアクセスできません。

unshare コマンドの使用

次のコマンドでは、指定したファイルシステムの共有が解除されます。

```
# unshare /usr/src
```

shareall

このコマンドを使用すると、複数のファイルシステムを共有することができます。オプションなしで使用すると、/etc/dfs/dfstab 内のすべてのエントリが共有されます。share コマンドを並べたファイルの名前を指定することができます。ファイル名を指定しないと、/etc/dfs/dfstab の内容が検査されます。“-” を使用してファイル名を置き換えれば、標準入力から share コマンドを入力できます。

shareall コマンドの使用

次のコマンドでは、ローカルファイルに羅列されているすべてのファイルシステムが共有されます。

```
# shareall /etc/dfs/special_dfstab
```

unshareall

このコマンドを使用すると、現在共有されているリソースがすべて使用できなくなります。–F *FSType* オプションによって、/etc/dfs/fstypes に定義されているファイルシステムタイプのリストを選択します。このフラグによって、特定のタイプのファイルシステムだけを共有解除できます。デフォルトのファイルシステムタイプは、/etc/dfs/fstypes に定義されています。特定のファイルシステムを選択するには、unshare コマンドを使います。

unshareall コマンドの使用

次の例では、NFS タイプのすべてのファイルシステムの共有が解除されます。

```
# unshareall -F nfs
```

showmount

このコマンドを使用すると、NFS サーバーから共有したファイルシステムをリモートにマウントしたすべてのクライアントや、クライアントがマウントしたファイルシステム、または共有しているファイルシステムとクライアントのアクセス情報が表示されます。構文は以下のとおりです。

```
showmount [ –ade ] [ hostname ]
```

–a を指定すると、すべてのリモートマウント (クライアント名とディレクトリ) のリストが表示されます。–d を指定すると、クライアントがリモートにマウントしたディレクトリのリストが表示されます。–e では、共有 (またはエクスポート) しているファイルのリストが表示されます。*hostname* には、表示する情報が保存されている NFS サーバーを指定します。*hostname* を指定しないと、ローカルホストを入力するように要求されます。

showmount コマンドの使用

次のコマンドでは、すべてのクライアント、およびマウントしたディレクトリが表示されます。

```
# showmount -a bee
lilac:/export/share/man
lilac:/usr/src
rose:/usr/src
tulip:/export/share/man
```

次のコマンドでは、マウントしたディレクトリが表示されます。

```
# showmount -d bee
/export/share/man
/usr/src
```

次のコマンドでは、共有しているファイルシステムが表示されます。

```
# showmount -e bee
/usr/src        (everyone)
/export/share/man     eng
```

setmnt

このコマンドを使用すると、/etc/mnttab テーブルが作成されます。このテーブルは、mount コマンドと umount コマンドで参照されます。通常、このコマンドを使用することはありません。システムがブートされるときに自動的に使用されます。

その他のコマンド

NFS の障害追跡には以下のコマンドを使用します。

nfsstat

このコマンドを使用すると、NFS と RPC 接続について統計情報を収集できます。構文は次のとおりです。

```
nfsstat [  –cmnrsz ]
```

–c を指定すると、クライアント側の情報が表示され、–m を指定すると、NFS マウントされた各ファイルシステムの統計が表示されます。–n では、クライアントとサーバーの両方の NFS 情報が表示され、–r では、RPC の統計が表示されます。–s を指定すると、サーバー側の情報が表示され、–z を指定すると、統計がゼロに設定されます。コマンド行にオプションを指定しないと、–cnrs が使用されます。

新しいソフトウェアやハードウェアを処理環境に追加した場合、サーバー側の統計を収集することが、デバッグにたいへん役立ちます。このコマンドを週に最低 1 度は実行し、履歴を作成するようにしてください。統計を保存しておくと、以前の効率のよい記録になります。

nfsstat コマンドの使用

```
# nfsstat -s

Server rpc:
Connection oriented:
calls      badcalls   nullrecv   badlen      xdrcall    dupchecks  dupreqs
11420263   0          0          0           0          1428274    19
Connectionless:
calls      badcalls   nullrecv   badlen      xdrcall    dupchecks  dupreqs
14569706   0          0          0           0          953332     1601

Server nfs:
calls      badcalls
24234967   226
Version 2: (13073528 calls)
null       getattr    setattr    root        lookup     readlink   read
138612 1%  1192059 9% 45676 0%   0 0%        9300029 71% 9872 0%   1319897 10%
wrcache    write      create     remove      rename     link       symlink
0 0%       805444 6%  43417 0%   44951 0%    3831 0%    4758 0%    1490 0%
mkdir      rmdir      readdir    statfs
2235 0%    1518 0%    51897 0%   107842 0%
Version 3: (11114810 calls)
null       getattr    setattr    lookup      access     readlink   read
141059 1%  3911728 35% 181185 1% 3395029 30% 1097018 9% 4777 0%   960503 8%
write      create     mkdir      symlink     mknod      remove     rmdir
763996 6%  159257 1%  3997 0%    10532 0%    26 0%      164698 1%  2251 0%
rename     link       readdir    readdirplus fsstat     fsinfo     pathconf
53303 0%   9500 0%    62022 0%   79512 0%    3442 0%    34275 0%   3023 0%
commit
73677 0%

Server nfs_acl:
Version 2: (1579 calls)
null       getacl     setacl     getattr     access
0 0%       3 0%       0 0%       1000 63%    576 36%
```

(続く)

```
Version 3: (45318 calls)
null        getacl     setacl
0 0%        45318 100% 0 0%
```

上記は、NFS サーバーの統計です。最初の 5 行は RPC に関するもので、残りの部分は NFS のアクティビティのレポートです。どちらの統計でも総コール数に対する badcalls の数や 1 週間あたりの calls 数がわかるので、障害が発生した時点を突き止めのに役立ちます。badcallsの値は、クライアントからの不良メッセージの数を表すもので、ネットワーク上のハードウェアにおける問題を突き止められます。

いくつかの接続では、ディスクに対する書き込みアクティビティが発生します。この数値の急激な上昇は障害の可能性を示すものなので、調査が必要です。NFS バージョン 2 の場合、特に注意しなければならない接続は、setattr、write、create、remove、rename、link、symlink、mkdir、および rmdir です。 NFS バージョン 3 の場合には、commit の値に特に注意します。ある NFS サーバーの commit レベルが、それと同等のサーバーと比較して高い場合は、NFS クライアントに十分なメモリーがあるかどうかを確認してください。サーバーの commit オペレーションの数は、クライアントにリソースがない場合に上昇します。

pstack

このコマンドを使用すると、各プロセスにおけるスタックトレースが表示されます。root で実行しなければなりません。プロセスがハングした場所を判断するのに使用します。使用できるオプションは、チェックするプロセスの PID だけです (proc(1) のマニュアルページを参照)。

以下の例では、実行中の nfsd プロセスをチェックしています。

```
# /usr/proc/bin/pstack 243
243:    /usr/lib/nfs/nfsd -a 16
 ef675c04 poll     (24d50, 2, ffffffff)
 000115dc ???????? (24000, 132c4, 276d8, 1329c, 276d8, 0)
 00011390 main     (3, effffff14, 0, 0, ffffffff, 400) + 3c8
 00010fb0 _start   (0, 0, 0, 0, 0, 0) + 5c
```

プロセスが新規の接続要求を待っていることが示されています。これは、正常な反応です。要求が行われた後でもプロセスがポーリングしていることがスタックからわかった場合、そのプロセスはハングしている可能性があります。652ページの「NFS サービスを再起動する方法」 の指示に従って問題を解決してください。ハングしたプログラムによって問題が発生しているかどうかを確実に判断するには、647ページの「NFS における障害追跡の手順」 を参照してください。

rpcinfo

このコマンドは、システムで動作している RPC サービスに関する情報を生成します。RPC サービスの変更にも使用できます。このコマンドには、たくさんのオプションがあります (`rpcinfo(1M)` のマニュアルページを参照)。以下は、このコマンドで使用できるオプションの構文です。

```
rpcinfo [ –m | –s ] [ hostname ]
rpcinfo [ –t | –u ] [ hostname ] [ progname ]
```

`–m` は `rpcbind` 操作の統計テーブル、`–s` は登録済みの RPC プログラムすべての簡易リスト、`–t` は TCP を使用する RPC プログラム、`–u` は UDP を使用する RPC プログラムを表示します。*hostname* は情報を取得する元のサーバー、*progname* は情報を収集する対象の RPC プログラムです。*hostname* を指定しないと、ローカルホスト名が使用されます。*progname* の代わりに RPC プログラム番号が使えますが、ユーザーが覚えやすいのは番号よりも名前です。NFS バージョン 3 が実行されていないシステムでは、`–s` オプションの代わりに `–p` オプションが使えます。

このコマンドで生成されるデータには、以下のものがあります。

- RPC プログラム番号
- 特定プログラムのバージョン番号
- 使用されているトランスポートプロトコル
- RPC サービス名
- RPC サービスの所有者

rpcinfo コマンドの使用

次の例では、サーバーで実行している RPC サービスに関する情報を収集しています。生成されたテキストには sort コマンドのフィルタをかけ、より読みやすくしています。この例では、RPC サービスの数行を省略しています。

```
% rpcinfo -s bee |sort -n
   program version(s) netid(s)                          service     owner
    100000  2,3,4     udp,tcp,ticlts,ticotsord,ticots  portmapper  superuser
    100001  4,3,2     ticlts,udp                       rstatd      superuser
    100002  3,2       ticots,ticotsord,tcp,ticlts,udp  rusersd     superuser
    100003  3,2       tcp,udp                          nfs         superuser
    100005  3,2,1     ticots,ticotsord,tcp,ticlts,udp  mountd      superuser
    100008  1         ticlts,udp                       walld       superuser
    100011  1         ticlts,udp                       rquotad     superuser
    100012  1         ticlts,udp                       sprayd      superuser
    100021  4,3,2,1   ticots,ticotsord,ticlts,tcp,udp  nlockmgr    superuser
    100024  1         ticots,ticotsord,ticlts,tcp,udp  status      superuser
    100026  1         ticots,ticotsord,ticlts,tcp,udp  bootparam   superuser
    100029  2,1       ticots,ticotsord,ticlts          keyserv     superuser
    100068  4,3,2     tcp,udp                          cmsd        superuser
    100078  4         ticots,ticotsord,ticlts          kerbd       superuser
    100083  1         tcp,udp                          -           superuser
    100087  11        udp                              adm_agent   superuser
    100088  1         udp,tcp                          -           superuser
    100089  1         tcp                              -           superuser
    100099  1         ticots,ticotsord,ticlts          pld         superuser
    100101  10        tcp,udp                          event       superuser
    100104  10        udp                              sync        superuser
    100105  10        udp                              diskinfo    superuser
    100107  10        udp                              hostperf    superuser
    100109  10        udp                              activity    superuser
  .
  .
    100227  3,2       tcp,udp                          -           superuser
    100301  1         ticlts                           niscachemgr superuser
    390100  3         udp                              -           superuser
1342177279  1,2       tcp                              -           14072
```

次の例では、サーバーの特定トランスポートを使用している RPC サービスの情報を収集する方法について説明しています。

```
% rpcinfo -t bee mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
% rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

最初の例では、TCP で実行している mountd サービスをチェックしています。2 番目の例では、UDP で実行している NFS サービスをチェックしています。

snoop

このコマンドは、ネットワーク上のパケットの監視によく使用されます。root として実行しなければなりません。クライアントとサーバーの両方で、ネットワークハードウェアが機能しているかどうかを確認する方法としてよく使用されます。使用できるオプションは多数あります (snoop(1M) のマニュアルページを参照)。以下で、このコマンドの概要を説明します。

```
snoop  [ -d device ] [ –o filename ] [ host hostname  ]
```

–d *device* には、ローカルネットワークインタフェースを指定します。–o *filename* には、取り込んだすべてのパケットを保存するファイルを指定します。*hostname* には、表示するパケットが通過したホストを指定します。

–d *device* オプションは、複数のネットワークインタフェースがあるサーバーで特に有効です。ホストの設定以外にも、使用できる式が多数あります。コマンド式を grep で組み合わせることでも、十分に使用できるデータを生成できます。

障害追跡をする場合は、パケットの発信元と送信先のホストが正しいことを確認してください。また、エラーメッセージも調べてください。パケットをファイルに保存すると、データの検査が容易になります。

truss

このコマンドを使用すると、プロセスがハングしたかどうかを確認できます。root で実行しなければなりません。このコマンドに指定できるオプションは多数あります (truss(1) のマニュアルページを参照)。構文の概要は次のとおりです。

```
truss [ -t syscall ] –p pid
```

–t *syscall* には、追跡するシステムコールを指定します。–p *pid* には、追跡するプロセスの PID を指定します。*syscall* には、追跡するシステムコールをコンマで区切って指定することもできます。また、*syscall* の指定を ! で始めると、そのシステムコールは追跡されなくなります。

次の例は、プロセスが新しいクライアントからの接続要求を待っていることを示しています。

```
# /usr/bin/truss -p 243
poll(0x00024D50, 2, -1)         (sleeping...)
```

これは正常な反応です。新規接続の要求が行われた後でも反応が変わらない場合、そのプロセスはハングしている可能性があります。652ページの「NFS サービスを再起動する方法」 の指示に従ってプログラムを修正してください。ハングしたプログラムによって問題が発生しているかどうかを確実に判断するには、647ページの「NFS における障害追跡の手順」 を参照してください。

コマンドを組み合わせて使用する

以下の節では、NFS の複雑な機能をいくつか紹介します。

バージョン 2 とバージョン 3 のネゴシエーション

NFS サーバーがサポートしているクライアントが NFS バージョン 3 を使用していない場合に備えて、開始手順にはプロトコルレベルのネゴシエーションが含まれています。クライアントとサーバーの両方がバージョン 3 をサポートしていると、バージョン 3 が使用されます。どちらか片方でもバージョン 2 しかサポートしていないと、バージョン 2 が使用されます。

ネゴシエーションによって決まった値は、`mount` コマンドに対して `–vers` オプションを使用することで変更できます (`mount_nfs(1M)` のマニュアルページを参照してください)。ほとんどの場合、デフォルトによって最適なバージョンが選択されるため、ユーザーが指定する必要はありません。

UDP と TCP のネゴシエーション

開始時には、トランスポートプロトコルもネゴシエートされます。デフォルトでは、クライアントとサーバーの両方がサポートしているコネクション型トランスポートの中で最初に見つかったものが選択されます。それが見つからない場合には、コネクションレス型トランスポートプロトコルの中で最初に見つかったものが

使用されます。システムでサポートされているトランスポートプロトコルのリストは、/etc/netconfig にあります。TCP はコネクション型トランスポートプロトコルで、Solaris 2.6 でサポートされています。UDP はコネクションレス型トランスポートプロトコルです。

NFS プロトコルのバージョンとトランスポートプロトコルが両方ともネゴシエーションによって決まった場合は、NFS プロトコルのバージョンがトランスポートプロトコルよりも優先されます。UDP を使用する NFS バージョン 3 プロトコルの方が、TCP を使用する NFS バージョン 2 プロトコルよりも優先されます。mount コマンドでは NFS プロトコルのバージョンもトランスポートプロトコルも手動で選択できます (mount_nfs(1M) のマニュアルページを参照)。ほとんどの場合、ネゴシエーションによって選択されるオプションの方が適切です。

ファイル転送サイズのネゴシエーション

ファイル転送サイズは、クライアントとサーバーの間でデータを転送するときに使用されるバッファーのサイズです。原則として、ファイル転送サイズが大きいほど性能が向上します。NFS バージョン 3 には転送サイズに上限はありませんが、Solaris 2.6 以降がデフォルトで提示するバッファーサイズは 32K バイトです。クライアントは、必要であればマウント時にこれより小さい転送サイズを提示することができますが、ほとんどの場合必要ありません。

転送サイズは、NFS バージョン 2 を使用しているシステムとはネゴシエートされません。このとき、ファイル転送サイズの上限は 8K バイトに設定されます。

mount コマンドに対して –rsize オプションと –wsize オプションを使用すると、転送サイズを手動で設定できます。PC クライアントの一部では転送サイズを小さくする必要があります。また、NFS サーバーが大きなファイル転送サイズに設定されている場合には、転送サイズを大きくすることができます。

ファイルシステムのマウントの詳細

クライアントがサーバーからファイルシステムをマウントするとき、そのファイルシステムに対応するファイルハンドルをサーバーから取得する必要があります。そのためには、クライアントとサーバーの間でいくつかのトランザクションが発生します。この例では、クライアントはサーバーから /home/terry をマウントします。snoop によって追跡したトランザクションは、次のとおりです。

```
client -> server PORTMAP C GETPORT prog=100005 (MOUNT) vers=3 proto=UDP
server -> client PORTMAP R GETPORT port=33492
client -> server MOUNT3 C Null
server -> client MOUNT3 R Null
client -> server MOUNT3 C Mount /export/home9/terry
server -> client MOUNT3 R Mount OK FH=9000 Auth=unix
client -> server PORTMAP C GETPORT prog=100003 (NFS) vers=3 proto=TCP
server -> client PORTMAP R GETPORT port=2049
client -> server NFS C NULL3
server -> client NFS R NULL3
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

この追跡結果では、クライアントがまずマウントポート番号を NFS サーバーの portmap サービスに要求します。クライアントが取得したマウントポート番号 (33492) は、サーバーに対する存在確認のために使用されます。このポート番号でサービスが実行中であることが確認できると、クライアントはマウントを要求します。この要求により、サーバーはマウントされるファイルシステムに対するファイルハンドル (9000) を送ります。これに対してクライアントは、NFS ポート番号を要求します。クライアントはサーバーからポート番号を受け取り、NFS サービス (nfsd) を ping してから、ファイルハンドルを使用してファイルシステムに関する NFS 情報を要求します。

次の追跡結果では、クライアントは –public オプションを使用してファイルシステムをマウントしています。

```
client -> server NFS C LOOKUP3 FH=0000 /export/home9/terry
server -> client NFS R LOOKUP3 OK FH=9000
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

デフォルトの公共ファイルハンドル (0000) を使用しているために、すべてのトランザクションにポートマップサービスから情報が与えられ、NFS ポート番号を決定するためのトランザクションはありません。

マウント時の -public オプションと NFS URL の意味

–public オプションを使用すると、マウントが失敗することがあります。NFS URL を組み合わせると、状況がさらに複雑になる可能性があります。これらのオプショ

ンを使用した場合にファイルシステムがどのようにマウントされるかは、次のとおりです。

public オプションと **NFS URL** – 公共ファイルハンドルが使用されます。公共ファイルハンドルがサポートされていないと、マウントは失敗します。

public オプションと通常のパス – 公共ファイルハンドルが使用されます。公共ファイルハンドルがサポートされていないと、マウントは失敗します。

NFS URL のみ – NFS サーバーでサポートされていれば、公共ファイルハンドルを使用します。公共ファイルハンドルを使用するとマウントが失敗する場合は、MOUNT プロトコルを使用してマウントします。

通常のパスのみ – 公共ファイルハンドルは使用しないでください。MOUNT プロトコルが使用されます。

クライアント側フェイルオーバー機能

クライアント側のフェイルオーバー (障害時回避) 機能を使用すると、複製されたファイルシステムをサポートしているサーバーが使用不能になったときに、NFS クライアントは別のサーバーに切り替えることができます。ファイルシステムが使用不能になる原因としては、接続しているサーバーのクラッシュ、サーバーの過負荷、ネットワーク障害が考えられます。通常、このような場合のフェイルオーバー機能はユーザーにはわかりません。設定が行われていれば、フェイルオーバー機能はクライアント上のプロセスを中断することなく実行されます。

フェイルオーバー機能が行われるためには、ファイルシステムが読み取り専用でマウントされている必要があります。また、ファイルシステムが完全に同じでないとフェイルオーバー機能は成功しません。ファイルシステムが同一になる条件については、697ページの「複製されたファイルシステムとは」 を参照してください。フェイルオーバー機能の候補としては、静的なファイルシステム、または変更の少ないファイルシステムが適しています。

CacheFS を使用してマウントされたファイルシステムは、フェイルオーバー機能には使えません。CacheFS ファイルシステムは、それぞれについて追加情報が格納されています。この情報はフェイルオーバーの際に更新できないため、ファイルシステムをマウントするときにはフェイルオーバー機能と CasheFS のどちらか片方の機能しか使えません。

各ファイルシステムについて用意すべき複製の数を決める要素はさまざまです。一般的に、サーバーを何台か用意してそれぞれが複数のサブネットをサポートするという環境の方が、サブネット 1 つについて 1 台のサーバーを用意するよりもすぐ

れています。この場合、リストにあるサーバーを 1 台ずつチェックする必要があるため、リスト上のサーバーが増えるにつれてマウントにかかる時間も増えます。

フェイルオーバー機能に関する用語

フェイルオーバー機能のプロセスを完全に理解するには、以下の 2 つの用語を理解しておく必要があります。

- フェイルオーバー機能 – 複製されたファイルシステムに対応するサーバーのリストから、サーバーを選択すること。通常、ソートされたリストの順番を元に、次のサーバーが応答するならばそのサーバーが使用されます。
- 再マッピング – 新しいサーバーを使用すること。クライアントは、正常な状態のときにリモートファイルシステム上のアクティブなファイルそれぞれのパス名を格納します。再マッピング時には、そのパス名に基づいて新しいサーバー上のファイルを見つけます。

複製されたファイルシステムとは

フェイルオーバー機能に関して、あるファイルシステムのすべてのファイルが元のファイルシステムのファイルとサイズも vnode タイプも同じ場合に、そのファイルシステムを「複製」といいます。アクセス権、作成日付などのファイル属性は関係ありません。ファイルサイズか vnode タイプが異なると再マッピングは失敗し、元のサーバーが再び使用可能になるまでプロセスはハングします。

複製されたファイルシステムを保守するには、`rdist` や `cpio` などのファイル転送機構を使います。複製されたファイルシステムを更新すると不整合が発生するので、できるだけ以下を守ってください。

- 新しいバージョンのファイルをインストールするときは、あらかじめ古い方の名前を変更する
- クライアントによる使用が少ない夜間に更新を実行する
- 更新は小規模にとどめる
- コピーの数を最小限にする

フェイルオーバー機能と NFS ロック

ソフトウェアパッケージの一部は、ファイルに読み取りロックをかける必要があります。そのようなソフトウェアが正常に動作できるようにするため、読み取り専用

ファイルシステムに対しても読み取りロックがかけられるようになっています。ただし、これはクライアント側でしか認識されません。サーバー側で意識されないため、再マッピングされてもロックはそのまま残ります。ファイルはもともと変更が許されないので、サーバー側でファイルをロックする必要はありません。

大型ファイル

Solaris 2.6 およびその互換バージョンでは、2G バイトを超えるファイルを扱えます。デフォルトでは、UFS ファイルシステムはこの新機能を活かすために `-largefiles` オプション付きでマウントされます。以前のリリースでは、2G バイトを超えるファイルは扱えません。具体的な方法については 616ページの「NFS サーバー上で大型ファイルを無効にする方法」 を参照してください。

サーバーのファイルシステムが `-largefiles` オプション付きでマウントされていれば、Solaris 2.6 の NFS クライアントでは何も変更しなくても大型ファイルにアクセスできます。しかし、Solaris 2.6 のコマンドすべてで大型ファイルが扱えるわけではありません。大型ファイルを処理可能なコマンドのリストは、`largefile(5)` を参照してください。大型ファイル用機能拡張を備えた NFS バージョン 3 プロトコルをサポートしていないクライアントは、大型ファイルには一切アクセスできません。Solaris 2.5 クライアントでは、NFS バージョン 3 プロトコルを使用することはできますが、大型ファイルを扱う機能は含まれていません。

NFS サーバーログ機能の働き

NFS サーバーログ機能は NFS の読み取りと書き込み、およびこのファイルシステムを変更する操作の記録を提供します。このデータは情報へのアクセスを追跡するのに利用できます。さらに、この記録は、情報へのアクセスを測定する定量的な方法を提供します。

ログ機能が有効になっているファイルシステムにアクセスすると、カーネルが raw データをバッファーファイルに書き込みます。このデータには、時刻表示、クライアント IP アドレス、要求者の UID、アクセスされているファイルまたはディレクトリオブジェクトのファイルハンドル、および発生している操作のタイプがあります。

`nfslogd` デーモンはこの raw データを、ログファイルに保存される ASCII レコードに変換します。使用可能なネームサービス機能が一致しているものを見付けると、その変換中に IP アドレスはホスト名に変更され、UID はログインに変更されます。ファイルハンドルはパス名にも変換されます。この操作を完了するために、デーモンはファイルハンドルを追跡し続けて別のファイルハンドル内の情報をパス

テーブルに保存します。その結果、ファイルハンドルにアクセスするたびにパスを識別し直す必要はありません。nfslogd がオフのときは、ファイルハンドル内でのパステーブルへのマッピングへの変更を追跡する機能が働かなくなるため、このデーモンは常時実行させておく必要があります。

WebNFS サービスの動作方法

WebNFS サービスとは、あるディレクトリに置かれたファイルを、公開ファイルハンドルを使用してクライアントからアクセスできるようにするものです。ファイルハンドルは、NFS クライアントがファイルを識別できるようにカーネルが生成するアドレスです。公開ファイルハンドルの値はあらかじめ決まっているため、サーバーがクライアントに対してファイルハンドルを生成する必要はありません。定義済みのファイルハンドルを使用するというこの機能によって、MOUNT プロトコルが不要になってネットワークトラフィックが減り、クライアントにとっては性能が向上します。

デフォルトでは、NFS サーバーの公開ファイルハンドルはルートファイルシステムに対して設定されます。このデフォルトのため、サーバーに対してマウント権限を持っているすべてのクライアントに対して WebNFS アクセス権が与えられます。公開ファイルハンドルは、share コマンドによって任意のファイルシステムに切り替えることができます。

あるファイルシステムに対するファイルハンドルをクライアントが持っているとき、アクセスするファイルに対応するファイルハンドルを知るには LOOKUP を実行します。 NFS プロトコルでは、パス名の構成要素を 1 度に 1 つしか評価できません。したがって、ディレクトリ階層のレベルが 1 つ増えるたびに 1 回ずつ LOOKUP を実行します。公開ファイルハンドルからの相対パスに対して LOOKUP を実行する場合には、WebNFS サーバーは複数構成要素参照という方法によって 1 度にパス名全体を評価できます。複数構成要素参照を使用することにより、WebNFS サーバーはパス名の中のディレクトリレベルを 1 つずつファイルハンドルに変換しなくても目的のファイルに対するファイルハンドルを取得できます。

また、NFS クライアントは単一の TCP 接続上で同時に複数のダウンロードを行うこともできます。これにより、複数の接続を設定することによる余分な負荷をサーバーにかけずに、高速なアクセスが実現できます。Web ブラウザアプリケーションも複数ファイルを同時にダウンロードできますが、それぞれのファイルに独自の接続が確立されます。WebNFS ソフトウェアは接続を 1 つしか使用しないため、サーバーに対するオーバーヘッドを軽減できます。

パス名の中の最後の構成要素が他のファイルシステムに対するシンボリックリンクである場合、通常の NFS アクティビティによってあらかじめそのファイルへのアクセス権を持っていれば、クライアントはそのファイルにアクセスできます。

通常、NFS URL は公開ファイルハンドルからの相対位置として評価されます。パスの先頭にスラッシュを 1 つ追加すると、サーバーのルートファイルシステムからの相対位置に変更できます。次の例では、公開ファイルハンドルが `/export/ftp` ファイルシステムに設定されていればこの 2 つの NFS URL は同等です。

```
nfs://server/junk
nfs://server//export/ftp/junk
```

WebNFS セキュリティネゴシエーション機能の働き方

Solaris 8 リリースには、WebNFS クライアントが WebNFS サーバーと、選択されたセキュリティメカニズムについてネゴシエーションできるようにする新しいプロトコルがあります。この新しいプロトコルは、セキュリティネゴシエーションマルチコンポーネントルックアップを使用しています。これは、WebNFS プロトコルの以前のバージョンで使用されていたマルチコンポーネントルックアップの拡張版です。

WebNFS クライアントは、公共ファイルハンドルを使用して通常のマルチコンポーネントルックアップ要求を行うことにより、このプロセスを開始します。このクライアントには、サーバーがどのようにしてこのパスを保護しているかについての知識がないため、デフォルトのセキュリティメカニズムが使用されます。デフォルトのセキュリティメカニズムでは不十分な場合は、サーバーは `AUTH_TOOWEAK` エラーを返します。このメッセージは、そのデフォルトメカニズムが有効ではなく、クライアントはより強力なメカニズムを使用する必要があることを意味しています。

クライアントは、`AUTH_TOOWEAK` エラーを受信すると、サーバーに対してどのセキュリティメカニズムが必要か決定するように要求します。この要求が成功すると、サーバーは、指定されたパスに必要なセキュリティメカニズムの配列を返します。このセキュリティメカニズムの配列のサイズによっては、クライアントは完全な配列を得るためにさらに要求を出さなければならない場合があります。サーバーが WebNFS セキュリティネゴシエーションをサポートしていない場合は、この要求は失敗します。

要求が正常に受け入れられたら、WebNFS クライアントは、配列からの 1 番目のセキュリティメカニズムを選択します。クライアントはこの配列をサポートしていて、ファイルハンドルを取得するために選択されたセキュリティメカニズムを使用して通常のマルチコンポーネントルックアップ要求を発行します。このあとに続く

すべての NFS 要求は、選択されたセキュリティメカニズムとファイルハンドルを使用して出されます。

Web ブラウザの使用と比較した場合の WebNFS の制約

HTTP を使用する Web サイトで実現可能な機能のいくつかは、WebNFS ではサポートされていません。この違いは、NFS サーバーはファイルを送るだけであるため、特別な処理はすべてクライアントで行う必要があることが原因です。ある Web サイトを WebNFS と HTTP 両方のアクセスに対応させるには、以下を考慮してください。

- NFS によるブラウズでは CGI スクリプトは実行されません。したがって、CGI スクリプトを多用している Web サイトを含むファイルシステムは、NFS によるブラウズに適していない可能性があります。
- ブラウザからは、形式の異なるファイルを扱うために別のビューアを起動されることがあります。NFS URL からそうしたファイルにアクセスすると、ファイル名からファイルタイプが判別できるならば外部のビューアが起動されます。ブラウザは、NFS URL が使用されている場合、標準の MIME タイプで決まっているファイル名拡張子をすべて認識します。WebNFS は一部の Web ブラウザとは異なり、ファイルタイプを決定するときにファイルの内部は調べません。ファイル名拡張子だけで判断します。
- NFS によるブラウズでは、サーバー側のイメージマップ (クリッカブルイメージ) は使えません。しかしクライアント側のイメージマップ (クリッカブルイメージ) は、位置とともに URL が定義されているため使えます。文書サーバーからの応答は不要です。

Secure NFS システム

NFS 環境は、アーキテクチャやオペレーティングシステムの異なるコンピュータから構成されるネットワーク上でファイルシステムを共有するためには、強力で使いやすい手段です。しかし、NFS の操作によるファイルシステムの共有を便利にする機能が、一方ではセキュリティ上の問題につながっています。今まで、NFS はほとんどのバージョンで UNIX (AUTH_SYS) 認証を使用してきましたが、現在では AUTH_DH のようなより強力な認証方式も使用可能です。UNIX 認証の場合、NFS サーバーはファイル要求を認証するために、その要求を行なったユーザーではなくコンピュータを認証します。したがって、クライアント側のユーザーがスーパーユーザーでログインすると、ファイルの所有者になりすますことができ

ます。DH 認証では、NFS サーバーはユーザーを認証するため、このような操作が困難になります。

ルートへのアクセス権とネットワークプログラミングについての知識があれば、だれでも任意のデータをネットワークに入れ、ネットワークから任意のデータを取り出すことができます。ネットワークに対する最も危険な攻撃は、有効なパケットを生成したり、または「対話」対話を記録し後で再生することによってユーザーを装うなどの手段により、データをネットワークに持ち込むことです。これらはデータの整合性に影響を与えます。許可を持つユーザーを装うことなく、単にネットワークトラフィックを受信するだけの受動的な盗み聞きならば、データの整合性が損なわれることはないため、それほど危険ではありません。ユーザーはネットワークに送信されるデータを暗号化することによって、機密情報のプライバシを守ることができます。

ネットワークのセキュリティ問題における共通の対処方法は、解決策を各アプリケーションにゆだねることです。さらに優れた手法としては、すべてのアプリケーションを対象として、標準の認証システムを導入することです。

Solaris オペレーティングシステムには、NFS が実装されるメカニズムであるリモート手続き呼び出し (RPC) のレベルで、認証システムが組み込まれています。このシステムは Secure RPC と呼ばれ、ネットワーク環境のセキュリティを大幅に向上させるとともに、NFS のセキュリティを強化します。Secure RPC の機能を利用した NFS システムを Secure NFS システムと呼びます。

Secure RPC

Secure RPC は Secure NFS システムの基本となるメカニズムです。Secure RPC の目標は、少なくともタイムシェアリングシステム (すべてのユーザーが 1 台のコンピュータを共有するシステム) 程度に安全なシステムを構築することです。タイムシェアリングシステムはログインパスワードによりユーザーを認証します。DES (Data Encryption Service) 認証でもこれは同じです。ユーザーは、ローカル端末の場合と同じように、任意のリモートコンピュータにログインできます。ユーザーのログインパスワードは、ネットワークセキュリティへのパスポートです。タイムシェアリングでは、システム管理者は信頼のおける人で、パスワードを変更して誰かを装うようなことはしないという道徳上の義務を負います。Secure RPC では、ネットワーク管理者は「公開鍵」を格納するデータベースのエントリを変更しないという前提で信頼されています。

RPC 認証システムを理解するには、「資格 (credential)」と「ベリファイア」という 2 つの用語を理解する必要があります。ID バッジを例にとれば、資格とは、名前、

住所、誕生日など人間を識別するものです。ベリファイアとはバッジに添付された写真であり、バッジの写真をその所持者と照合することによって、そのバッジが盗まれたものではないことを確認できます。RPC では、クライアントプロセスは RPC 要求のたびに資格とベリファイアの両方をサーバーに送信します。クライアントはサーバーの資格をすでに知っているため、サーバーはベリファイアだけを送り返します。

RPC の認証機能は拡張が可能で、さまざまな認証システムを組み込むことができます。現在のところ、このようなシステムには UNIX と DH の 2 つがあります。

ネットワークサービスで UNIX 認証を使用する場合、資格にはクライアントのコンピュータ名、UID、GID、グループアクセスリストが含まれ、ベリファイアには何も含まれません。ベリファイアが存在しないため、ルートユーザーは `su` などのコマンドを使用して、適切な資格を偽ることができます。UNIX 認証でのもう 1 つの問題は、ネットワーク上のすべてのコンピュータを UNIX コンピュータと想定していることです。UNIX 認証を異機種ネットワーク内の他のオペレーティングシステムに適用した場合、これは正常に動作しません。

UNIX 認証の欠点を補うために、Secure RPC では DH 認証を使います。

DH 認証

DH 認証は、Data Encryption Standard (DES) と Diffie-Hellman 公開鍵暗号手法を使用してネットワーク上のユーザーとコンピュータの両方を認証します。DES は標準暗号化機能であり、Diffie-Hellman 公開鍵暗号手法は、公開鍵と非公開鍵という 2 つの鍵を使用する暗号方式です。公開鍵と非公開鍵は名前空間に格納されます。NIS の場合、それらの鍵を publickey マップに格納し、NIS+ は cred テーブルに格納します。これらのマップにはすべての認証の候補ユーザーの公開鍵と非公開鍵が入っています。マップとテーブルの設定については、『*Solaris* ネーミングの管理』を参照してください。

DH 認証のセキュリティは、送信側が現在時刻を暗号化する機能に基づいていて、受信側はこれを復号して、自分の時刻と照合します。タイムスタンプは DES を使用して暗号化されます。この方式が機能するには次の条件が必要です。

- 2 つのエージェントの現在時刻が一致している。
- 送信側と受信側が同じ暗号化鍵を使用する。

ネットワークが時間同期プログラムを実行する場合、クライアントとサーバー上の時間は自動的に同期されます。時間同期プログラムを使用できない場合、ネットワーク時間ではなく、サーバーの時間を使用してタイムスタンプを計算できま

す。クライアントは、RPC セッションを開始する前にサーバーに時間を要求し、自分のクロックとサーバーのクロックとの時間差を計算します。タイムスタンプを計算するときには、この差を使用してクライアントのクロックを補正します。サーバーがクライアントの要求を拒否するほど、クライアントとサーバーのクロック同期がずれた場合、DH 認証システムはサーバーとの間で再び同期をとります。

クライアントとサーバーは、ランダムな対話鍵 (セッションキーとも呼びます) を生成することによって、同じ暗号化鍵に到達します。次に、公開鍵暗号手法 (公開鍵と秘密鍵を必要とする暗号化方式) を使用して共通鍵を推理します。この共通鍵は、クライアントとサーバーだけが推理できる鍵です。対話鍵は、クライアントのタイムスタンプを暗号化および復号化するために使用されます。共通鍵は、この対話鍵を暗号化および復号化するために使用されます。

KERB 認証

Kerberos は MIT で開発された認証方式です。Kerberos での暗号化は DES に基づいています。Kerberos サポートは、現在では Secure RPC の一部としては供給されていませんが、Solaris 8 リリースにはクライアント側の実装が含まれています。

NFS での Secure RPC の使用

Secure RPC を使用する場合は、次の点に注意してください。

- サーバーがクラッシュしたとき周囲に誰もいない場合 (停電の後など) には、システムに格納されていた秘密鍵はすべて消去されます。そのためどのプロセスからも、セキュリティ保護されたネットワークサービスにアクセスしたり NFS ファイルシステムをマウントしたりできません。リブートの際に重要なプロセスは、通常は root として実行されます。したがって、root の秘密鍵を別に保存してあればこれらのプロセスを実行できますが、周囲に誰もいない状況では秘密鍵を復号化するパスワードを入力するユーザーがいません。`keylogin -r` を使用すると root の秘密鍵がそのまま `/etc/.rootkey` に格納され、`keyserv` がそれを読み取ります。
- システムによっては、シングルユーザーモードでブートし、コンソールには root のログインシェルが表示されてパスワードの入力が要求されないことがあります。このような場合には、物理的なセキュリティが不可欠です。
- ディスクレスコンピュータのブートは、完全に安全とはいえません。ブートサーバーになりすましてリモートコンピュータに対する秘密鍵の入力を記録するような、不正なカーネルを誰かがブートすることが考えられます。Secure NFS システ

ムによって保護されているのはカーネルとキーサーバーが起動した後だけです。それまでの間に、ブートサーバーからの応答を認証する手段はありません。これは重大な問題につながる可能性がありますが、この部分を攻撃するにはカーネルのソースコードを使用した高度な技術が必要です。また、不法行為の痕跡が残ります。すなわち、ネットワークを通じてブートサーバーにポーリングすれば、不正なブートサーバーの位置が分かります。

- ほとんどの setuid プログラムは root が所有者です。root の秘密鍵が /etc/.rootkey に格納されていれば、これらのプログラムは正常に動作します。しかし、ユーザーが所有者である setuid プログラムは動作しない可能性があります。たとえば、ある setuid プログラムの所有者が dave であり、ブート以降 dave が 1 度もログインしていないと、このプログラムはセキュリティ保護されたネットワークサービスにはアクセスできません。
- リモートコンピュータに (login、rlogin、または telnet を使用して) ログインし、keylogin を使用してアクセスすると、自分のアカウントへのアクセスを許したことになります。これは、秘密鍵が相手側のコンピュータのキーサーバーに渡され、キーサーバーがその秘密鍵を格納したためです。これが問題になるのは、相手側のリモートコンピュータを信用できない場合だけです。しかし、疑いがある場合にはパスワードを要求するリモートコンピュータにはログインしないでください。代わりに NFS 環境を使用して、そのリモートコンピュータから共有されているファイルシステムをマウントします。または、keylogout を使用してキーサーバーから秘密鍵を消去します。
- ホームディレクトリが共有されていて –o sec=dh オプションが指定されていると、リモートログインによって問題が生じる可能性があります。/etc/hosts.equiv ファイルか /.rhosts ファイルでパスワードを要求しないように設定すると、ユーザーはログインできますが、ローカルで認証が行われていないために自分のホームディレクトリにアクセスできません。パスワードを要求され、入力したパスワードがネットワークパスワードと一致すれば自分のホームディレクトリにアクセスできます。

autofs マップ

autofs は 3 種類のマップを使用します。

- マスターマップ
- 直接マップ

■ 間接マップ

autofs マスターマップ

auto_master マップでは、ディレクトリからマップへの関連付けを行います。これは、すべてのマップを指定するマスターリストであり、autofs が参照します。auto_master ファイルの内容の例を次に示します。

例 **31–1** /etc/auto_master ファイルの例

```
# Master map for automounter
#
+auto_master
/net            -hosts           -nosuid,nobrowse
/home           auto_home        -nobrowse
/xfn            -xfn
/-              auto_direct     -ro
```

この例では、汎用の auto_master ファイルに auto_direct マップのための追加が行われています。マスターマップ /etc/auto_master の各行は、次の構文に従っています。

```
mount-point  map-name  [  mount-options  ]
```

mount-point

mount-point はディレクトリのフル (絶対) パス名です。このディレクトリが存在しない場合、可能ならば autofs はこのディレクトリを作成します。このディレクトリが存在し、しかも空ではない場合、マウントすることによってその内容が隠されます。この場合、autofs は警告を出します。

マウントポイントとして /- を指定すると、マップが直接マップであり、このマップ全体に関連付けられている特定のマウントポイントがないことを表します。

map-name

map-name 名は、位置に対する指示またはマウント情報を検出するために、autofs が使用するマップです。この名前がスラッシュ (/) で始まる場合、autofs はこの名前をローカルファイルとして解釈します。そうでない場合、autofs はネームサービススイッチ構成ファイルで指定される検索によりマウント情報を検索します。/net と /xfn に対して使用される特殊なマップもあります (708ページの「マウントポイント /net」と 708ページの「マウントポイント /xfn」 を参照してください)。

mount-options

mount-options は、省略可能です。*map-name* のエントリに他のオプションがある場合を除き、*map-name* で指定されたエントリのマウントに適用されるオプションをコンマで区切って並べます。具体的なファイルシステムごとのオプションについては、そのファイルシステムで mount のマニュアルページを参照してください (たとえば NFS 固有のマウントオプションについては、mount_nfs(1M) のマニュアルページを参照のこと)。NFS 固有のマウントポイントの場合、bg (バックグラウンド) オプションと fg (フォアグラウンド) オプションは適用されません。

で始まる行はコメント行です。この場合、その行の最後まですべて無視されます。

長い行を短い行に分割するには、行末にバックスラッシュ (\) を入力します。入力できる文字数の上限は 1024 です。

注 - 2 つのエントリで同じマウントポイントが使用されるときは、1 番目のエントリは automount コマンドが使用します。2 番目のエントリは無視されます。

マウントポイント /home

マウントポイント /home は、/etc/auto_home (間接マップ) に記述されたエントリがマウントされるディレクトリです。

注 - autofs はすべてのコンピュータで動作し、デフォルトでは /net と /home (自動マウントされるホームディレクトリ) をサポートします。このデフォルトは、NIS ならば auto.master マップ、NIS+ ならば auto_master テーブルを使用して、またはローカルの /etc/auto_master ファイルを編集することによって変更できます。

マウントポイント /net

autofs は、特別のマップ -hosts 内の全エントリをディレクトリ /net の下にマウントします。これは hosts データベースだけを使用する組み込みマップです。たとえば、コンピュータ gumbo が hosts データベース内にあり、しかもそのファイルシステムのどれかをエクスポートする場合、次のコマンドによって、カレントディレクトリがコンピュータ gumbo のルートディレクトリに変更されます。

```
%cd /net/gumbo
```

なお、autofs はホスト gumbo のエクスポートされたファイルシステムだけをマウントできます。つまり、ローカルディスク上のファイルシステムではなく、ネットワークユーザーが使用できるサーバー上のファイルシステムです。したがって、gumbo にあるすべてのファイルとディレクトリは、/net/gumbo では利用できない場合があります。

/net を使用したアクセスでは、サーバー名はパスの中に指定されるため、位置に依存します。したがって、エクスポートされるファイルシステムを別のサーバーに移動すると、そのパスは使えなくなります。このような場合は /net を使用せずに、そのファイルシステムに対応するエントリをマップの中に設定します。

注 - autofs はマウント時だけサーバーのエクスポートリストを調べます。サーバーのファイルシステムが一度マウントされると、そのファイルシステムがアンマウントされ、次にマウントされるまで autofs はそのサーバーをチェックしません。したがって、新たにエクスポートされたファイルシステムは、それがサーバーからアンマウントされ、再度マウントされるまでは見えません。

マウントポイント /xfn

このマウントポイントは、FNS 名前空間を使用して共有されるリソースの autofs ディレクトリ構造がマウントされます (FNS については、『*Solaris* ネーミングの設定と構成』を参照してください)。

直接マップ

直接マップは自動マウントポイントです。つまり、直接マップによって、クライアント上のマウントポイントとサーバー上のディレクトリが直接対応付けられます。

直接マップには完全なパス名があり、明示的に関係を示します。次に一般的な /etc/auto_direct マップを示します。

```
/usr/local          -ro \
   /bin                     ivy:/export/local/sun4 \
   /share                   ivy:/export/local/share \
   /src                     ivy:/export/local/src
   /usr/man         -ro     oak:/usr/man \
                            rose:/usr/man \
                            willow:/usr/man
   /usr/games       -ro     peach:/usr/games
   /usr/spool/news  -ro     pine:/usr/spool/news \
                            willow:/var/spool/news
```

直接マップの行は、次の構文に従っています。

key [*mount-options*] *location*

key — *key* は直接マップでのマウントポイントのパス名です。

mount-options — *mount-options* は、このマウントに適用したいオプションです。これらのオプションは、マップのデフォルトと異なる場合だけ必要です。各ファイルシステムの種類ごとのオプションについては、そのファイルシステムの mount のマニュアルページを参照してください (たとえば CacheFS に固有のマウント操作については、mount_cachefs(1M) のマニュアルページを参照してください)。

location — *location* にはファイルシステムの位置を、NFS ファイルシステムならば *server:pathname*、High Sierra ファイルシステム (HSFS) ならば *:devicename* という形式で指定します。

注 - *pathname* には自動マウントしたマウントポイントを含めず、ファイルシステムへの実際の絶対パスである必要があります。たとえば、ホームディレクトリの位置は、*server*:/home/*username* ではなく、*server*:/export/home/*username* として表示する必要があります。

マスターマップと同様、# で始まる行はコメントです。その行のテキストの最後まですべて無視されます。長い行を短い行に分割するには、行の最後にバックスラッシュを入力します。

すべてのマップの中で、直接マップのエントリが、/etc/vfstab (vfstab にはマウントされるすべてのファイルシステムのリストが含まれる) の対応するエントリと最もよく似ています。/etc/vfstab のエントリは次のとおりです。

```
dancer:/usr/local - /usr/local/tmp nfs - yes ro
```

直接マップでは次のようになります。

```
/usr/local/tmp      -ro      dancer:/usr/local
```

注 - オートマウンタマップの間では、オプションの連結はされません。あるオートマウンタマップでオプションが追加されると、それまでに見つかったマップに指定されているオプションはすべて無視され、新しいオプションだけが使用されます。たとえば、auto_master マップに指定されているオプションは、他のマップの中の対応するエントリによって上書きされます。

この種類のマップについては、他にも重要な機能があります。717ページの「autofs がクライアント用の最も近い読み取り専用ファイルを選択する方法 (複数ロケーション)」 を参照してください。

マウントポイント /–

例 31–1 にある /- というマウントポイントは、auto_direct の中のエントリを具体的なマウントポイントに関連付けないように autofs に指示します。間接マップの場合は、auto_master ファイルに定義されたマウントポイントを使います。直接マップの場合は、ここに示されたマップの中で指定されたマウントポイントを使います (直接マップのキーとマウントポイントはフルパス名であることに注意してください)。

NIS または NIS+ の auto_master ファイルには、直接マップのエントリは 1 つしか存在できません。マウントポイントは 1 つの名前空間の中で一意でなければならないためです。auto_master がローカルファイルならば、重複しないかぎり直接マップのエントリがいくつあってもかまいません。

間接マップ

間接マップは、キーの置換値を使用してクライアント上のマウントポイントとサーバー上のディレクトリとを対応させます。間接マップは、ホームディレクトリなどの特定のファイルシステムをアクセスするのに便利です。auto_home マップは間接マップの一例です。

間接マップ内の行は次の一般的な構文になります。

```
key  [  mount-options  ]  location
```

表 31–3 Table Caption

key	*key* は間接マップでの単純名 (スラッシュなし) です。
mount-options	*mount-options* は、このマウントに適用するオプションです。これらのオプションが必要なのは、マップのデフォルトと異なる場合だけです。各ファイルシステムタイプごとのオプションについては、そのファイルシステムの mount のマニュアルページを参照してください (たとえば NFS 固有のマウントオプションについては、mount_nfs(1M) のマニュアルページを参照してください)。
location	*location* はファイルシステムステムの位置を示し、*server*:*pathname* により (1 つまたは複数) 指定されます。 **注 -** *pathname* には自動マウントしたマウントポイントを含めず、ファイルシステムへの実際の絶対パスである必要があります。たとえば、ディレクトリの位置は、*server*:/net/*server*/usr/local ではなく、*server*:/usr/local として表示する必要があります。

マスターマップと同様、# で始まる行はコメントです。その行のテキストの最後まですべて無視されます。長い行を短い行に分割するには、行の最後にバックスラッシュ (\) を入力します。例 31–1 に、次のエントリを含む auto_master マップを示します。

```
/home       auto_home          -nobrowse
```

auto_home は、/home のもとでマウントされるエントリを含む間接マップの名前です。一般的な auto_home マップには次の構文が含まれます。

```
david                   willow:/export/home/david
rob                     cypress:/export/home/rob
gordon                  poplar:/export/home/gordon
rajan                   pine:/export/home/rajan
tammy                   apple:/export/home/tammy
jim                     ivy:/export/home/jim
linda     -rw,nosuid    peach:/export/home/linda
```

例として、前のマップがホスト oak にあると想定します。ユーザー linda がホームディレクトリを /home/linda として指定するパスワードデータベースにエントリがある場合、コンピュータ oak にログインするたびに、autofs はコンピュータ peach に常駐する /export/home/linda ディレクトリをマウントします。彼女のホームディレクトリは、読み書き可能な nosuid にマウントされます。

次のような状況が発生したと想定してください。ユーザー linda のホームディレクトリがパスワードデータベースに、/home/linda として表示されます。Linda も含め誰でも、前の例のマップを参照するマスターマップで設定されたどのコンピュータからでも、このパスにアクセスできます。

こうした状況のもとでは、ユーザー linda はこれらのどのコンピュータでも login や rlogin を実行し、代わりに彼女用のホームディレクトリをマウントさせることができます。

さらに、これで linda は次のコマンドも入力できます。

```
% cd ~david
```

autofs は彼女のために David のホームディレクトリをマウントします (すべてのアクセス権で許可されている場合)。

注 - オートマウンタマップの間には、オプションの連結はありません。オートマウンタマップに追加されたいずれのオプションも、前に検索されたマップに表示されているすべてのオプションを上書きします。たとえば、auto_master マップに含まれているオプションは、その他いずれのマップの対応するエントリによって上書きされます。

ネームサービスのないネットワークでこれを行うには、ネットワーク上のすべてのシステムで、すべての関連ファイル (/etc/passwd など) を変更する必要があります。NIS では、NIS マスターサーバーで変更を行い、関連するデータベースをスレーブのデータベースに伝達します。NIS+ を稼働中のネットワークでは、変更後に関連データベースがスレーブサーバーに自動的に伝達されます。

autofs のしくみ

autofs は、自動的に適切なファイルシステムをマウントするためのクライアント側のサービスです。クライアントが現在マウントされていないファイルシステムをアクセスしようとすると、autofs ファイルシステムはその要求に介入し、automountd を呼び出して要求されたディレクトリをマウントします。automountd はディレクトリを検索してマウントし、応答します。応答を受け取ると、autofs は待たせてあった要求の処理を続行させます。それ以降のそのマウントへの参照は autofs によって切り替えられ、このファイルシステムが一定時間使用されないために自動的にアンマウントされるまで、automountd は不要となります。

自動マウントを行うのに、次のコンポーネントが相互に動作します。

- automount コマンド
- autofs ファイルシステム
- automountd デーモン

automount コマンドは、システム起動時に呼び出され、マスターマップファイル auto_master を読み取って autofs マウントの最初のセットを作成します。これらの autofs のマウントは起動時に自動的にはマウントされません。後でファイルシステムがマウントされるポイントです。このようなポイントをトリガーノードと呼ぶこともあります。

autofs マウントが設定されると、要求があったときにファイルシステムをマウントすることができます。たとえば、autofs が、現在マウントされていないファイルシステムをアクセスする要求を受け取ると、automountd を呼び出して要求されたファイルシステムを実際にマウントさせます。

Solaris 2.5 からは、automountd デーモンは完全に automount コマンドから独立しました。そのため、automountd デーモンを 1 度停止してから起動し直さなくてもマップ情報を追加、削除、および変更できるようになりました。

最初に autofs マウントをマウントすると、automount コマンドを使用して、auto_master 内のマウントのリストをマウントテーブルファイル /etc/mnttab (以前は /etc/mtab) 内のマウントされているファイルシステムのリストと比較し、必要な変更を行うことにより、autofs マウントを更新します。こうすることにより、システム管理者は auto_master 内のマウント情報を変更し、autofs デーモンを停止したり、再起動したりすることなく、それらの変更結果を autofs プロセスに使用させることができます。ファイルシステムがマウントされれば、以後のアクセスに automountd は不要となります。次に automountd が必要になるのは、ファイルシステムが自動的にアンマウントされるときです。

mount とは異なり、automount はマウントすべきファイルシステムを調べるために /etc/vfstab ファイル (これは各コンピュータごとに異なる) を参照しません。automount コマンドは、ドメイン内とコンピュータ上で名前空間とローカルファイルを通して制御されます。

autofs のしくみの概要を簡単に説明します。

自動マウントのデーモンである automountd は、ブート時に /etc/init.d/autofs スクリプトから起動されます (図 31–1 を参照してください)。このスクリプトは automount コマンドも実行します。このコマンドはマス

ターマップを読み取り (715ページの「autofs のナビゲーションプロセス開始法 (マスターマップ)」 を参照)、autofs のマウントポイントをインストールします。

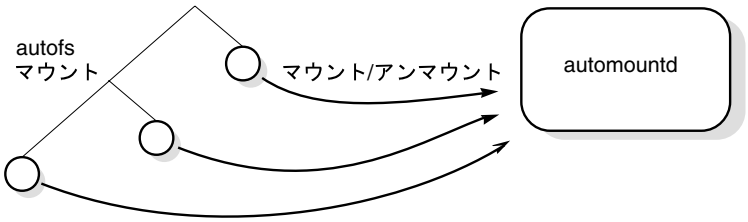


図 31–1　/etc/init.d/autofs スクリプトによる automount の起動

autofs は、自動マウント操作とアンマウント操作をサポートするカーネルファイルシステムの 1 つです。

autofs のマウントポイントにおいてファイルシステムへのアクセス要求が出された場合、autofs は次のように機能します。

1. autofs がその要求に介入します。
2. autofs は要求されたファイルシステムをマウントするよう、automountd にメッセージを送信します。
3. automountd がマップからファイルシステム情報を見つけ、マウントを実行します。
4. autofs は、介入した要求の実行を続行させます。
5. 一定時間そのファイルシステムがアクセスされないと、autofs はそのファイルシステムをアンマウントします。

注 - autofs サービスによって管理されるマウントは、手動でマウントまたはアンマウントは行わないでください。たとえこの操作がうまくいったとしても、autofs サービスはオブジェクトがアンマウントされたことを認識しないので、一貫性が損なわれる恐れがあります。リブートによって、autofs のマウントポイントがすべて消去されます。

---

autofs のネットワークナビゲート (マップ)

autofs は一連のマップを探索することによって、ネットワークをナビゲートします。マップとは、ネットワーク上の全ユーザーのパスワードエントリ、またはネットワーク上の全ホストコンピュータ名などの情報が収められているファイルです。つまり、UNIX 管理ファイルのネットワーク版といえます。マップはローカル

に使用するか、または NIS や NIS+ のようなネットワークネームサービスを通じて使用できます。Solstice システム管理ツールを使用して、ユーザーは自分の環境ニーズに適合するマップを作成します。724ページの「autofs のネットワークナビゲート法の変更 (マップの変更)」 を参照してください。

autofs のナビゲーションプロセス開始法 (マスターマップ)

automount コマンドはシステムの起動時にマスターマップを読み取ります。図 31–2 に示すように、マスターマップ内の各エントリは、直接または間接のマップ名、そのパス、およびそのマウントオプションです。エントリの順序は重要ではありません。automount は、マスターマップ内のエントリとマウントテーブル内のエントリを比較して、現在のリストを生成します。

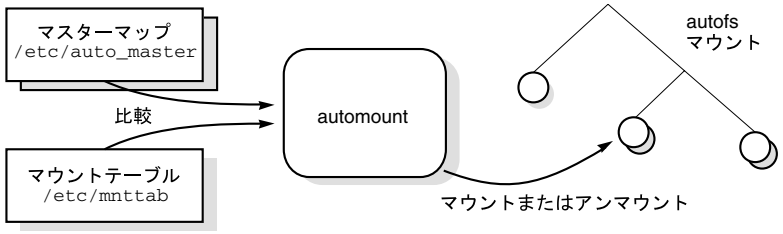


図 31–2　マスターマップによるナビゲーション

autofs マウントプロセス

マウント要求が発生したときに autofs サービスが何を実行するかは、オートマウンタマップの設定によって異なります。マウントプロセスの基本はすべてのマウントで同じですが、指定されているマウントポイントとマップの複雑さによって結果が変わります。Solaris 2.6 ではマウントプロセスも変更され、トリガーノードも作成されるようになりました。

単純な autofs マウント

autofs マウントプロセスの説明のために、以下のファイルがインストールされていると仮定します。

```
$ cat /etc/auto_master
# Master map for automounter
```

```
#
+auto_master
/net        -hosts        -nosuid,nobrowse
/home       auto_home     -nobrowse
/xfn        -xfn
/share      auto_share
$ cat /etc/auto_share
# share directory map for automounter
#
ws          gumbo:/export/share/ws
```

/share ディレクトリがアクセスされると、autofs サービスは /share/ws に対するトリガーノードを作成します。これは、/etc/mnttab の中では以下のようなエントリとなります。

```
-hosts  /share/ws     autofs  nosuid,nobrowse,ignore,nest,dev=###
```

/share/ws ディレクトリがアクセスされると、autofs サービスは以下の手順を実行します。

1. サーバーのマウントサービスが有効かどうかを確認するために、サービスに対して ping を行います。
2. 要求されたファイルシステムを、/share の下にマウントします。これで、/etc/mnttab ファイルには以下のエントリが追加されます。

```
-hosts  /share/ws     autofs  nosuid,nobrowse,ignore,nest,dev=###
gumbo:/export/share/ws /share/ws   nfs   nosuid,dev=####    #####
```

階層型マウント

オートマウンタファイルに複数の層が定義されていると、マウントプロセスは多少複雑になります。上記の例の /etc/auto_shared ファイルを拡張して以下のエントリを追加したとします。

```
# share directory map for automounter
#
ws        /       gumbo:/export/share/ws
          /usr    gumbo:/export/share/ws/usr
```

この場合、/share/ws マウントポイントがアクセスされたときのマウントプロセスは基本的に最初の例と同じです。また、/share/ws ファイルシステムの中に次のレベル (/usr) へのトリガーノードを作成することにより、そのレベルがアクセスされたときにマウントできるようにします。 この例でトリガーノードが作成されるためには、NFS に /export/share/ws/usr が存在している必要があります。

注意 - 階層を指定するときに、–soft オプションは使用しないでください。 この制限に関する説明は、717ページの「autofs アンマウント」を参照してください。

autofs アンマウント

一定時間アクセスがないためにアンマウントされるときは、マウントと反対の順序で実行されます。あるディレクトリより上位のディレクトリが使用中であれば、それより下のディレクトリだけがアンマウントされます。 アンマウントすると、トリガーノードがすべて削除され、ファイルシステムがアンマウントされます。ファイルシステムが使用中であれば、アンマウントは失敗してトリガーノードは再インストールされます。

注意 - 階層的にマウントを指定する場合は、–soft オプションは使用しないでください。–soft オプションを使用すると、トリガーノードを再インストールする要求がタイムアウトすることがあります。トリガーノードを再インストールできないと、マウントの次の階層にアクセスできません。この問題は、オートマウンタがすべての階層の構成要素をアンマウントすることでしか回避できません。具体的には、ファイルシステムが自動的にアンマウントされるのを待つか、システムをリブートすることになります。

autofs がクライアント用の最も近い読み取り専用ファイルを選択する方法 (複数ロケーション)

次のような直接マップの例では、マウントポイント /usr/man と /usr/spool/news には、複数のロケーション (/usr/man には 3 つ、/usr/spool/news には 2 つ) が記述されています。

```
/usr/local          -ro \
   /bin                    ivy:/export/local/sun4\
   /share                  ivy:/export/local/share\
```

(続く)

```
   /src                    ivy:/export/local/src
/usr/man           -ro     oak:/usr/man \
                           rose:/usr/man \
                           willow:/usr/man
/usr/games         -ro     peach:/usr/games
/usr/spool/news    -ro     pine:/usr/spool/news \
                           willow:/var/spool/news
```

このような場合、複製された位置のどれからマウントしてもユーザーは同じサービスを受けられます。ユーザーの書き込みまたは変更が可能ならば、その変更をロケーション全体で管理しなければならなくなるので、この手順は、読み取り専用のファイルシステムをマウントするときにだけ意味があります。あるときに、あるサーバー上のファイルを変更し、またすぐに別のサーバー上で「同じ」ファイルを変更しなければならないとしたら、大変面倒な作業になります。この利点は、最も利用しやすいサーバーが、そのユーザーの手をまったく必要としないで自動的にマウントされるということです。

ファイルシステムを複製として設定してあると (697ページの「複製されたファイルシステムとは」 を参照)、クライアントはフェイルオーバー機能を使用できます。最適なサーバーが自動的に決定されるだけでなく、そのサーバーが使用できなくなるとクライアントは自動的に 2 番目に適したサーバーを使います。フェイルオーバー機能は、Solaris 2.6 の新機能です。

複製として設定するのに適しているファイルシステムの例は、マニュアルページです。大規模なネットワークでは、複数のサーバーがマニュアルページをエクスポートできます。どのサーバーからマニュアルページをマウントしても、そのサーバーが動作しており、しかもそのファイルシステムをエクスポートしている限り、問題ありません。上の例では、複数のマウント位置は、マップエントリ内のマウント位置のリストになっています。

```
/usr/man -ro oak:/usr/man rose:/usr/man willow:/usr/man
```

これで、サーバー `oak`、`rose`、`willow` のどれからでもマニュアルページをマウントできます。どのサーバーが最適であるかは、いくつかの要素によって決まります。具体的には、ある特定の NFS プロトコルレベルをサポートしているサーバーの数、サーバーとの距離、重み付けです。

順位を決定するときには、NFS バージョン 2 と NFS バージョン 3 のプロトコルをサポートしているサーバーの数が数えられます。サポートしているサーバーの数が

多いプロトコルがデフォルトになります。そのため、クライアントにとっては利用できるサーバーの数が最大になります。

プロトコルのバージョンが同じサーバーの組の中で数が最も多いものがわかると、サーバーのリストが距離によってソートされます。ローカルサブネット上のサーバーには、リモートサブネット上のサーバーよりも高い優先順位が付けられます。最も近いサーバーが優先されることにより、待ち時間が短縮されネットワークトラフィックは軽減されます。図 31–3 に、サーバーとの距離を示します。

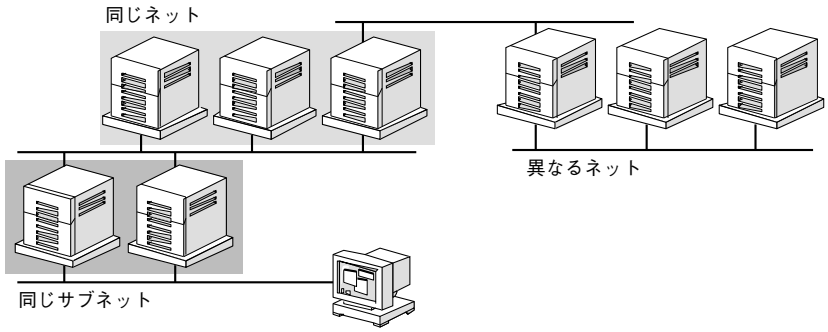


図 31–3　サーバーとの距離

ローカルサブネット上に同じプロトコルをサポートしているサーバーが複数あるときは、それぞれのサーバーに接続する時間が計測され、速いものが使用されます。優先順位には、重み付けも関係します (720ページの「autofs と重み付け」 を参照してください)。

バージョン 3 のサーバーの方が多いと、優先順位の決定は複雑になります。通常、ローカルサブネット上のサーバーはリモートサブネット上のサーバーよりも優先されます。バージョン 2 のサーバーがあり、それが最も近いバージョン 3 サーバーよりも近いと状況が複雑になる可能性があります。ローカルサブネットにバージョン 2 サーバーがあり、最も近いバージョン 3 サーバーがリモートサブネット上にあると、バージョン 2 サーバーが優先されます。このことは、バージョン 3 サーバーの方がバージョン 2 サーバーよりも多い場合にしかチェックされません。バージョン 2 サーバーの方が多いと、バージョン 2 サーバーしか選択されません。

フェイルオーバー機能を指定していると、この優先順位はマウント時に 1 回、マウントするサーバーを選択するときにチェックされ、その後は選択されたサーバーが使用できなくなるたびにチェックされます。複数位置を指定しておくと、個々のサーバーが一時的にファイルシステムをエクスポートできないときに便利です。

多くのサブネットを持つ大規模なネットワークでは、この機能は特に便利です。autofs は最も近いサーバーを選択するため、NFS のネットワークトラフィックを

ローカルネットワークセグメントに制限します。複数のネットワークインタフェースを持つサーバーの場合は、それぞれが別々のサーバーであるとみなして、各ネットワークインタフェースに対応付けられているホスト名を指定します。autofs はそのクライアントに一番近いインタフェースを選択します。

autofs と重み付け

距離のレベルが同じサーバーから 1 つを選択するために、autofs マップに重み付けの値を追加することができます。たとえば次のようにします。

```
/usr/man -ro oak,rose(1),willow(2):/usr/man
```

括弧内の数値が重み付けを表します。重み付けのないサーバーの値はゼロです (選択される可能性が最高です)。重み付けの値が大きいほど、そのサーバーが選択される可能性は低くなります。

注 - 重み付けは、サーバーの選択に関係する要素の中で最も小さい影響力しかありません。ネットワーク上の距離が同じサーバーの間で選択を行う場合に考慮されるだけです。

マップエントリ内の変数

変数名の前にドル記号 ($) を付けることによって、クライアント固有の変数を作成できます。この機能は、同じファイルシステムの位置にアクセスする異なるアーキテクチャタイプの調整に役立ちます。変数名を括弧でくくることで、そのうしろに続く文字や数字と変数とを区切ることができます。表 31–4 に定義済みのマップ変数を示します。

表 31–4 定義済みのマップ変数

変数	意味	情報提供元	例
ARCH	アーキテクチャタイプ	uname -m	sun4
CPU	プロセッサタイプ	uname -p	sparc
HOST	ホスト名	uname -n	dinky
OSNAME	オペレーティングシステム名	uname -s	SunOS

表 31–4 定義済みのマップ変数 続く

変数	意味	情報提供元	例
OSREL	オペレーティングシステムのリリース	uname -r	5.4
OSVERS	オペレーティングシステムのバージョン (リリースのバージョン)	uname -v	FCS1.0

キーとして使用する場合を除いて、変数はエントリ行内のどこにでも使用できます。たとえば、SPARC と IA のアーキテクチャ用のバイナリを、それぞれ /usr/local/bin/sparc と /usr/local/bin/x86 からエクスポートしているファイルサーバーがある場合、クライアントは次のようにマップエントリを通じてマウントできます。

```
/usr/local/bin  -ro server:/usr/local/bin/$CPU
```

これで、すべてのクライアント上の同じエントリがすべてのアーキテクチャに適用されます。

注 - どの sun4 アーキテクチャ向けに書かれたアプリケーションでも、ほとんどはすべての sun4 プラットフォームで実行できます。したがって、–ARCH 変数は sun4m ではなく、sun4 に固定されています。

他のマップを参照するマップ

ファイルマップで使用されたマップエントリ *+mapname* により、automount は指定されたマップを、あたかも現在のマップに組み込まれているかのように読み取ります。*mapname* の前にスラッシュがない場合、autofs はそのマップ名を文字列として扱い、ネームサービススイッチ方式を使用してこれを検出します。パス名が絶対パス名の場合、automount はその名前のローカルマップを捜します。マップ名がダッシュ「-」で始まる場合、automount は xfn や hosts といった適切な組み込みマップを参照します。

このネームサービススイッチファイルには、automount と指定された autofs 用のエントリが収められています。そしてそのエントリには、ネームサービスが検索される順序が収められています。ネームサービススイッチファイルの例を次に示します。

```
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the /etc/netconfig
# file contains "switch.so" as a nametoaddr library for "inet" transports.
# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:         files nis
group:          files nis

# consult /etc "files" only if nis is down.
hosts:          nis [NOTFOUND=return] files
networks:       nis [NOTFOUND=return] files
protocols:      nis [NOTFOUND=return] files
rpc:            nis [NOTFOUND=return] files
ethers:         nis [NOTFOUND=return] files
netmasks:       nis [NOTFOUND=return] files
bootparams:     nis [NOTFOUND=return] files
publickey:      nis [NOTFOUND=return] files
netgroup:       nis
automount:      files nis
aliases:        files nis
# for efficient getservbyname() avoid nis
services:       files nis
```

この例では、まずローカルマップ、次に NIS マップが検索されます。したがって、最も頻繁にアクセスされるホームディレクトリに対してはローカルマップ /etc/auto_home に少数のエントリを登録しておき、その他のエントリについてはネームサービススイッチを使用して NIS マップに戻るという方法が可能です。

```
bill                cs.csc.edu:/export/home/bill
bonny               cs.csc.edu:/export/home/bonny
```

組み込まれたマップを参照したあと、一致するものがなければ、automount は現在のマップの走査を続けます。これは、+ エントリのあとにさらにエントリを追加できることを意味します。たとえば、マップが次のように組み込まれているとします。

```
bill                cs.csc.edu:/export/home/bill
bonny               cs.csc.edu:/export/home/bonny
+auto_home
```

組み込まれたマップは、ローカルファイル (ローカルファイルだけが + エントリを持つことができることに注意) または組み込みマップとすることができます。

```
+auto_home_finance      # NIS+ map
+auto_home_sales        # NIS+ map
+auto_home_engineering  # NIS+ map
+/etc/auto_mystuff      # local map
+auto_home              # NIS+ map
+-hosts                 # built-in hosts map
```

注 - NIS+ または NIS のマップでは「+」エントリを使用できません。

実行可能な **autofs** マップ

autofs マウントポイントを生成するコマンドを実行する autofs マップを作成することもできます。データベースやフラットファイルから autofs 構造を作成しなければならない場合には、実行可能な autofs マップが有効なことがあります。短所は、マップをすべてのホストにインストールしなければならないことです。実行可能なマップは、NIS と NIS+ のどちらのネームサービスにも含めることができません。

実行可能マップは、auto_master ファイルにエントリが必要です。

```
/execute    auto_execute
```

実行可能マップの例を示します。

```
#!/bin/ksh
#
# executable map for autofs
#

case $1 in
          src)  echo '-nosuid,hard bee:/export1' ;;
esac
```

この例が機能するためには、ファイルが /etc/auto_execute としてインストールされ、実行可能ビットがオン (パーミッションが 744) になっている必要があります。これらの条件のときに次のコマンドを実行すると、bee から /export1 ファイルシステムがマウントされます。

```
% ls /execute/src
```

autofs のネットワークナビゲート法の変更 (マップの変更)

マップへのエントリを変更、削除、または追加して、ユーザーの環境ニーズに合わせることができます。ユーザーが必要とするアプリケーションやその他のファイルシステムがその位置を変更すると、マップはこれらの変更を反映しなければなりません。autofs のマップは、いつでも変更できます。automountd が次にファイルシステムをマウントしたときにその変更内容が有効となるかどうかは、変更したマップと変更内容によって決まります。

ネームサービスに対する autofs のデフォルトの動作

ブート時に、autofs は /etc/init.d/autofs にあるスクリプトを使用して起動され、マスターマップ auto_master が検索されます (次に説明する規則が適用されます)。

autofs は、/etc/nsswitch.conf ファイルの自動マウントエントリで指定されたネームサービスを使用します。ローカルファイルや NIS ではなく NIS+ が指定された場合、マップ名はすべてそのまま使用されます。NIS が選択されていて autofs が必要なマップを検出できず、1 つまたは複数の下線を含むマップを検出した場合、以前の NIS ファイル名を使えるようにするため、autofs はその下線をドットに変換します。次に autofs はもう 1 度マップを調べます。この手順を図 31–4 に示します。

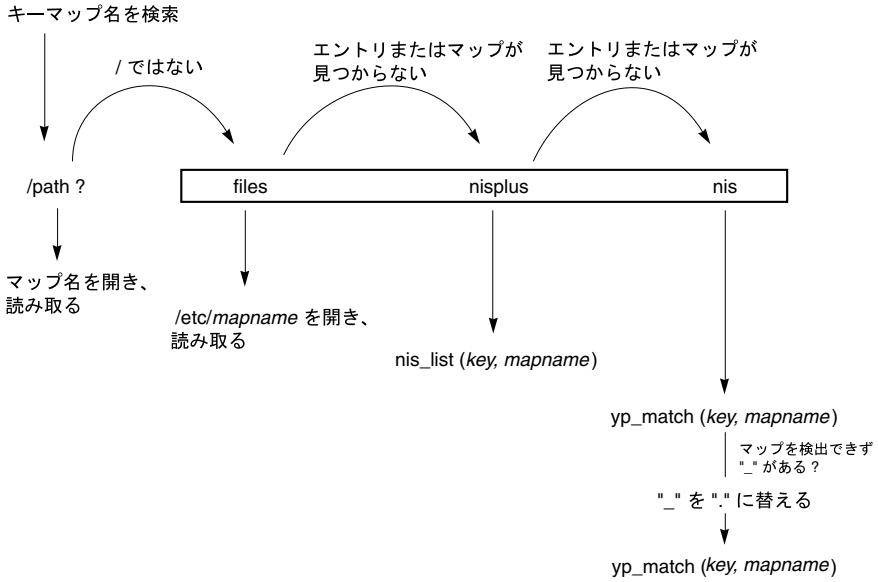


図 31–4　autofs によるネームサービスの使用

このセッションでの画面の動きは次の例のようになります。

```
$ grep /home /etc/auto_master
/home           auto_home

$ ypmatch brent auto_home
Can't match key brent in map auto_home.  Reason: no such map in
server's domain.

$ ypmatch brent auto.home
diskus:/export/home/diskus1/&
```

ネームサービスとして「ファイル」が選択された場合、すべてのマップは /etc ディレクトリ内のローカルファイルとみなされます。autofs は、使用するネームサービスとは無関係に、スラッシュ (/) で始まるマップ名をローカルとして解釈します。

autofs リファレンス

これ以降の節では、autofs の高度な機能を取り上げます。

メタキャラクタ

autofs は一部の文字を、特別な意味を持つものとして認識します。これらの文字には、置換に使用される文字や、他の文字を autofs のマップ構文解析機能から保護する目的のものがあります。

アンパサンド (&)

たとえば次のように、多数のサブディレクトリを指定したマップがあるとします。

```
john        willow:/home/john
mary        willow:/home/mary
joe         willow:/home/joe
able        pine:/export/able
baker       peach:/export/baker
```

この場合、文字列の置換が使えます。アンパサンド文字 (&) を使用して、任意の位置に記述されたこのキーを置換することができます。アンパサンドを使用した場合、上記のマップは次のようになります。

```
john        willow:/home/&
mary        willow:/home/&
joe         willow:/home/&
able        pine:/export/&
baker       peach:/export/&
```

キー置換はまた、次のような直接マップでも使用できます。

```
/usr/man      willow,cedar,poplar:/usr/man
```

これは次のようにも記述できます。

```
/usr/man      willow,cedar,poplar:&
```

なお、アンパサンドによる置換ではキー文字列全体を使用するため、直接マップでキーが / で始まる場合、そのスラッシュは引き継がれます。したがって、次のような指定はできません。

```
/progs    &1,&2,&3:/export/src/progs
```

これは、autofs が次のように解釈するためです。

```
/progs     /progs1,/progs2,/progs3:/export/src/progs
```

アスタリスク *(*)*

任意のキーを一致させるのに、任意の文字を表す置換文字であるアスタリスク (*) を使用できます。このマップエントリを使用して、すべてのホストから /export ファイルシステムをマウントできます。

```
*       &:/export
```

ここでは、各アンパサンドは特定のキーの値によって置換されています。autofs はこのアスタリスクをファイルの終わりとして解釈します。

特殊文字

特殊文字が含まれているマップエントリがある場合、autofs のマップ構文解析機能を混乱させるような名前のディレクトリについてはマウントする必要があります。autofs の構文解析機能は、名前に含まれるコロン、コンマ、スペースなどを認識します。これらの名前は二重引用符で囲んでください。

```
/vms    -ro    vmsserver: -  -  - "rc0:dk1 - "
/mac    -ro    gator:/ - "Mr Disk - "
```

第 **32** 章

メールサービスについてのトピック

第 33 章	メールサービスの概要
第 34 章	メールサービスの設定と障害回避方法 (トラブルシューティング)
第 35 章	メールサービスの背景情報について

第 33 章

メールサービスの導入

電子メールサービスの設定と維持管理は複雑な作業であり、ネットワークの日常の運用にとっても不可欠です。ネットワーク管理者は、既存のメールサービスを拡張したり、新規のネットワークやサブネットワークにメールサービスを設定することが必要なこともあります。ネットワークでのメールサービスの計画に役立つように、この章ではメールサービスの概念について説明し、典型的なメール構成の設定に必要な作業を手短かに述べます。

sendmail の新しい機能

Solaris 8 リリースには sendmail のバージョン 8.9.3 が付属しています。次にこの新しいバージョンに含まれている重要な変更およびユーザーの目につきやすい変更について説明します。

- MaxHeadersLength と呼ばれる新しい構成ファイルオプションは、指定されたメッセージのヘッダーラインすべての合計の長さを制限します。デフォルト値は 32,768 バイトです。受信したヘッダー付きメッセージでこの長さを超えるものは無視されます。
- /etc/default/sendmail と呼ばれる新しいファイルを使用すると、sendmail を起動する際に使用するオプションを init スクリプトに追加するのではなく、保

存することができます。このファイルを使用すると、init スクリプトを変更する必要がなくなるため、システムのグレードアップが容易になります。

- mail.local プログラムが拡張されてローカルメール転送プロトコルを使用できるようになりました。このプロトコルにより各受信側のエラーコードを返せるようになります。そのため、メッセージを受け取っていない受信側に対してだけ再送すればよくなり、すべての受信側に対してメッセージを再び待ち行列に入れる必要がなくなりました。Solaris 7 リリースではこのプロトコルが sendmail に追加されていました。
- /usr/bin/praliases という新しいコマンドを使用すると、エイリアスデータベース内のデータをテキストデータに変換することができます。コマンド行上で指定された引数がキーと一致した場合、キー: 値が出力されます。
- smrsh という新しいプログラムを使用すると、 sendmail の "|program" シンタックスを使用して実行することができるコマンドの個数を制限できます。この機能を有効にすると、/var/adm/sm.bin にあるプログラムのみを実行することができるようになります。主構成ファイルに FEATURE(`smrsh') を加えると、この機能が有効になります (詳しくは /usr/lib/mail/README を参照してください)。
- 不在返信プログラムに新しいオプションが追加されました。–f を使用すると ~/.vacation.*ext* の代わりのデータベースを選択できます。–m を使用すると ~/.vacation.msg の代わりのメッセージファイルを選択できます。–s を使用すると、受信するメッセージ内の UNIX From 行の代わりに応答アドレスを指定することができます。
- mailx プログラムへの変更は、送信側のベースとして、封筒送信側の代りに From: ヘッダーを使用することができるようになりました。この変更により mailx の機能は mailtool および dtmail と同じようになります。

Solaris 版の sendmail については、次のホームページを参照してください。http://www.sendmail.org/sun-specific/migration+sun.html

その他の sendmail の情報

次に、上記以外の sendmail 関連のホームページを示します。

- http://www.sendmail.org – sendmail のホームページ
- http://www.sendmail.org/faq – sendmail の FAQ
- http://www.sendmail.org/m4/readme.html – 新しい sendmail 構成ファイルの案内

- `http://www.sendmail.org/sun-specific/migration+sun.html` – Solaris 2.6 と Solaris 7 で提供される `sendmail` の違い

メールサービスに関する用語

メールサービスを確立するためには、多くのソフトウェアコンポーネントおよびハードウェアコンポーネントが必要になります。次の節ではこれらのコンポーネントとその説明に使用されるいくつかの用語について簡単に説明します。

最初の節では、メール配信システムのソフトウェア部分を説明するのに使用される用語を定義します。その次の節では、メール構成におけるハードウェアシステムの機能について取り上げます。

メールサービスソフトウェアコンポーネントの概要

次の表にメールシステムのソフトウェアコンポーネントを示します。ソフトウェアコンポーネントすべてに関する完全な説明については、772ページの「メールサービスソフトウェアの関連用語」を参照してください。

コンポーネント	説明
.forward ファイル	ユーザーのホームディレクトリ内で設定して、メールを自動的にリダイレクトしたり、プログラムに送ったりすることができるファイル
メールボックス	メールサーバー上にあり、電子メールメッセージの最終受信先であるファイル
メールアドレス	メールメッセージが配信される受信者またはシステムの名称が含まれる
メールエイリアス	メールアドレス内で使用されている代替名
メールキュー	メールサーバーによる処理を必要とするメールメッセージの集まり

コンポーネント	説明
ポストマスター	メールサービスについての問題を報告し質問を出すために使用される、特別のメールエイリアス
sendmail 構成ファイル	メールのルーティングに必要なすべての情報の入ったファイル

メール構成のハードウェア要素

メール構成では次の 3 つの要素が必要ですが、これらは同じシステムで組み合わせることも、別のシステムで提供することもできます。

- メールホスト – 解釈処理が困難なメールアドレスを扱うように構成されたシステム
- 少なくとも 1 台のメールサーバー – 1 つまたは複数のメールボックスを保持するように構成されたシステム
- メールクライアント – メールサーバーからメールにアクセスするシステム

ユーザーがドメイン外のネットワークと通信をするためには、4 番目の要素であるメールゲートウェイを追加する必要があります。

図 33–1 には、一般的な電子メール構成を示しますが、ここでは基本的な 3 つのメール要素とメールゲートウェイが使用されています。次の節では、各要素について説明します。

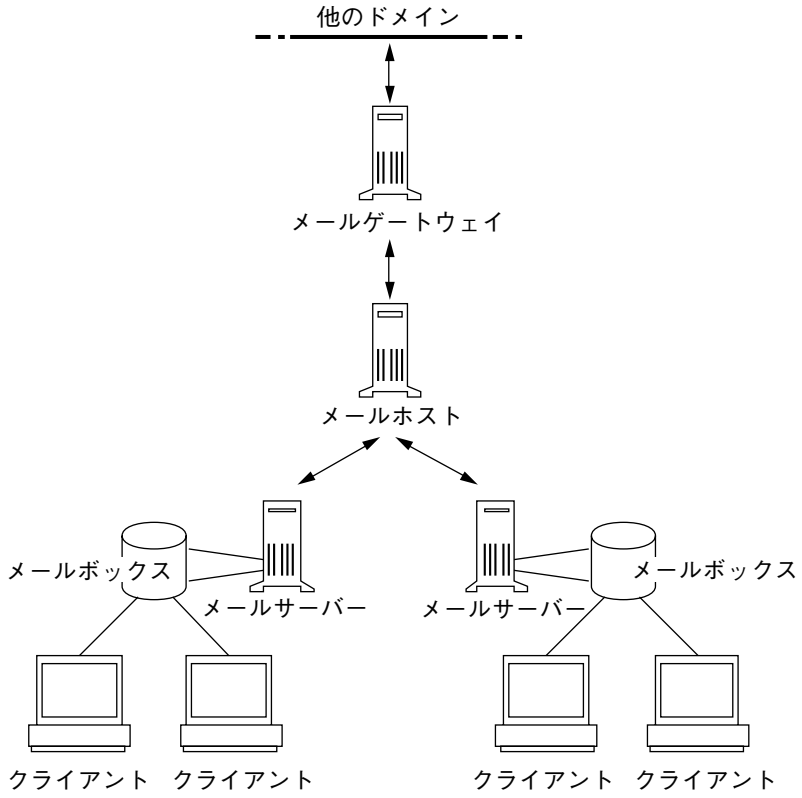


図 33–1　一般的な電子メール構成

各要素については、781ページの「メール構成のハードウェア要素」を参照してください。

第 34 章

メールサービスの設定と管理

この章ではメールサービスを設定し、管理する方法について説明します。メールサービスの管理についてまだ理解していない場合は、メールサービスの用語や構造の紹介、またメールサービス構成の説明について、第 33 章を参照してください。

メールシステムの計画

この節では、4 つの基本的なメール構成のタイプについて説明し、各構成を設定するために必要とされる作業について簡単に説明します。新しいメールシステムを設定する必要がある場合や、既存のものを拡張しようとする場合、この節が役立ちます。構成の最も基本的なケース (メールが完全にローカルで、外部との接続がない) から始め、メールゲートウェイを持つ 2 つのドメインの構成へと、順に複雑なものを取り上げます。まず最初に、これらの構成のどれを使用するのかを決める必要があります。次の構成が対象となります。

- 739ページの「リモートモードにおけるローカルメール」
- 739ページの「ローカルメールとリモート接続」
- 740ページの「2 つのドメインと 1 つのゲートウェイ」

システム管理者として、別名の更新とメールメッセージの転送の方法を決定してください。ユーザーがメールの転送要求やデフォルトのメール別名の変更要求を送る場所として、aliases メールボックスを設定できます。システムで NIS または NIS+ を使用する場合、転送の管理は、ユーザーではなく管理者が行うことができます。

ローカルメール専用

図 34–1 で示すように、最も単純なメール構成は、1 つのメールホストと、それに接続する複数のワークステーションです。メールは完全にローカルです。すべてのクライアントがローカルのディスクにメールを格納し、メールサーバーとして機能します。メールアドレスは /etc/mail/aliases ファイルを使用して構文解析されます。

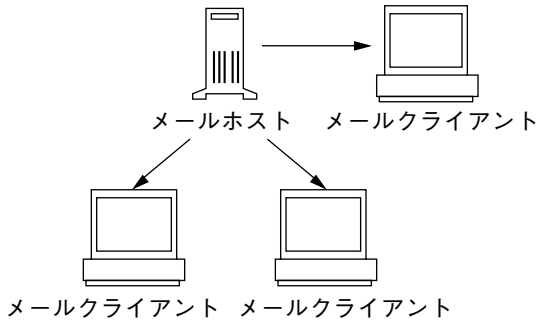


図 34–1　ローカルメール構成

この種のローカルメール構成を設定するには、以下が必要です。

- 各メールクライアントシステムでのデフォルトの /etc/mail/sendmail.cf ファイル (編集は不要)
- メールホストに指定されたサーバー (メールホストの /etc/hosts ファイルに mailhost.*domainname* を追加。NIS または NIS+ を実行していない場合は、すべてのメールクライアントの /etc/hosts ファイルにメールホスト IP アドレス行を追加)
- ローカルメールボックスを持つ任意のシステム上にある同じ内容の /etc/mail/aliases ファイル (NIS または NIS+ を実行していない場合)

- 各メールクライアントシステムでの /var/mail に、メールボックスを格納できるだけの十分な領域

リモートモードにおけるローカルメール

この構成では、各メールクライアントが、クライアントのメールボックスにメールのスプーリングが行える 1 つのメールサーバーから、メールをマウントします。このサーバーがメールホストになっても構いません。この構成では、各クライアントのメールボックスのバックアップが簡単に行えます。

この種類のメール構成を設定するには、次の設定が必要です。

- 各メールクライアントシステムでのデフォルトの /etc/mail/sendmail.cf ファイル (編集は不要)
- メールホストとして指定されたサーバー (メールホストの /etc/hosts ファイルに mailhost.*domainname* を追加。NIS または NIS+ を実行していない場合は、すべてのメールクライアントの /etc/hosts ファイルにメールホスト IP アドレス行を追加)
- ローカルメールボックスを持つ任意のシステムにある同じ内容の /etc/mail/aliases ファイル (NIS または NIS+ を実行していない場合)
- /var/mail ディレクトリをマウントするため、各メールクライアントの /etc/vfstab ファイルまたは /etc/auto_direct (autofs が使用されている場合) ファイルにエントリが必要
- メールサーバーの /var/mail でのクライアントのメールボックスを格納できるだけの十分な領域

ローカルメールとリモート接続

小規模のネットワークにおける最も一般的なメール構成を図 34–2 に示します。1 つのシステムが、メールサーバー、メールホスト、および外部へのメールゲートウェイを兼ねます。メールは /etc/mail/aliases ファイルを使用して配信されます。ネームサービスは必要ありません。

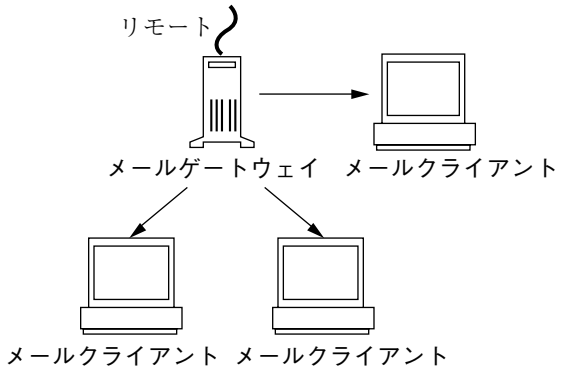


図 34–2　UUCP 接続を持つローカルメール構成

メールクライアントがメールホストの /var/mail からメールファイルをマウントする場合、この種のメール構成を設定するには、次の条件が必要です。

- メールゲートウェイの main.cf ファイル (MX レコードが使用される場合には編集は不要)
- 各メールクライアントシステムのデフォルト /etc/mail/sendmail.cf ファイル (編集は不要)
- メールホストに指定したサーバー (メールホストの /etc/hosts ファイルに mailhost.*domainname* を追加。NIS または NIS+ を実行していない場合は、すべてのメールクライアントの /etc/hosts ファイルにメールホスト IP アドレス行を追加)
- ローカルメールボックスを持つ任意のシステム上にある同じ内容の /etc/mail/aliases ファイル (NIS や NIS+ を実行していない場合)
- メールボックスがメールホストにあるときに、/var/mail ディレクトリをマウントするための各メールクライアントの /etc/vfstab ファイルか /etc/auto_direct (autofs が使用されている場合) にあるエントリ
- メールサーバーの /var/mail にクライアントメールボックスを格納できるだけの十分な領域

2 つのドメインと 1 つのゲートウェイ

図 34–3 に示したメール構成には、2 つのドメインと 1 つのメールゲートウェイがあります。この構成では、各ドメインのメールサーバー、メールホストおよびメールゲートウェイ (複数も可) が、異なるシステムの場合も多くなります。メールの管理や配信の過程をより簡単にするため、ネームサービスを使用します。

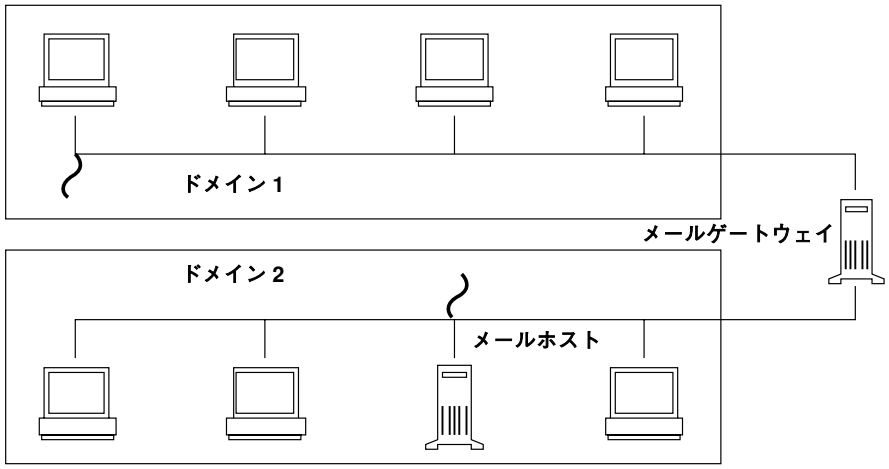


図 34–3　2 つのドメインと 1 つのゲートウェイ

メールクライアントがメールホストの /var/mail からメールファイルをマウントする場合、この種のメール構成を設定するには、以下の条件が必要です。

- 特別な規則が追加された、カスタマイズされた sendmail.cf ファイルを必要とする複雑なゲートウェイシステム
- メールゲートウェイに main.cf ファイル (MX レコードが使用される場合には編集は不要)
- メールホストとして指定されたサーバー (メールホストの /etc/hosts ファイルに mailhost.*domainname* を追加。NIS や NIS+ を実行していない場合は、すべてのメールクライアントの /etc/hosts ファイルにメールホスト IP アドレス行を追加)
- ローカルメールボックスを持つ任意のシステム上にある同じ内容の /etc/mail/aliases ファイル (NIS や NIS+ を実行していない場合)
- ユーザーにメールが格納される場所 (NIS+ では mail_aliases.org_dir、NIS では別名マップ) を示す別名エントリ
- 各メールクライアントシステムに、デフォルトの /etc/mail/sendmail.cf ファイル (編集は不要)
- メールボックスがメールホストにあるときに、/var/mail ディレクトリをマウントするための各メールクライアントの /etc/vfstab ファイルまたは /etc/auto_direct (autofs が使用されている場合) ファイルにエントリが必要
- メールサーバーの /var/mail に、クライアントのメールボックスを格納できるだけの十分な領域

メールサービスの設定

サイトが企業外の電子メール (email) サービスに接続していないか、または企業が 1 つのドメイン内にある場合は、メールサービスを比較的容易に設定できます。

メールでは、ローカルメール用に 2 種類の構成と、ドメイン外のネットワークとの通信用にさらに 2 つの構成が必要です。これらの構成は、同じシステムで組み合わせるか、または別のシステムで提供できます。機能を使用するには、サイト上のシステムを設定する必要があります。

- 743ページの「メールサーバーを設定する方法」
- 744ページの「メールクライアントを設定する方法」
- 746ページの「メールホストを設定する方法」
- 747ページの「メールゲートウェイを設定する方法」
- 748ページの「`sendmail` で DNS を使用する方法」

メールサービスの設定を始める前に、メールサーバー、メールホスト、およびメールゲートウェイとして機能するシステムを選びます。サービスを提供するすべてのメールクライアントのリストも作成し、メールボックスの位置を入れてください。このリストは、ユーザーのメール別名を作成するときに有用です。これらの各システムの持つ機能の詳細は、第 33 章を参照してください。便宜のために、どのシステムがメールサーバー、メールホスト、およびメールゲートウェイとして適しているかのガイドラインをこのあとの節でも再度説明します。

設定を簡単にするために、この章では、個々のメールサーバー、メールホスト、メールクライアント、およびリレーホストを設定するのに必要な操作を示します。メールサービス構成のシステムが複数の機能構成で動作する場合、システムのタイプごとに適切な操作説明に従ってください。たとえば、メールホストとメールサーバーの機能が同じシステムにある場合は、そのシステムをメールホストとして設定するための指示に従い、次に同じシステムをメールサーバーとして設定するための指示に従ってください。

注 - 次のメールサーバーとメールクライアントの設定の手順は、メールボックスが NFS でマウントされているときに適用されます。ただし、通常、メールボックスはローカルにマウントされた `/var/mail` ディレクトリで管理されます。この場合、このあとの手順は必要ありません。

▼ メールサーバーを設定する方法

メールサーバーはローカルユーザーにメールサービスを提供するだけなので、設定には特別な手順は必要ありません。ユーザーはパスワードファイルか名前空間にエントリが必要です。そのエントリは、メールの配信用にローカルのホームディレクトリが ~/.forward を確認できるために必要です。このためにホームディレクトリサーバーがしばしばメールサーバーとして設定されます。

メールサーバーは、クライアントからのすべてのメールのルーティングを行います。メールサーバーに必要な唯一のリソースは、クライアントメールボックスのための十分なスプール空間です。リモートのマウントでは、/var/mail ディレクトリを使用できるようにする必要があります。

この作業のために、/etc/dfs/dfstab ファイルを確認して /var/mail ディレクトリがエクスポートされていることを確認します。

1. メールサーバー上でスーパーユーザーになります。

2. /var/mail ディレクトリをリモートアクセスに使用できるかどうかを確認します。

   share と入力して Return キーを押します。/var/mail ディレクトリがリストになっていれば、これで終了です。/var/mail ディレクトリがリストされていなければ、次の手順に進みます。

3. /var/mail ディレクトリをリモートアクセスに使用できるようにします。

   以下のコマンドをキー入力します。

```
# share -F nfs /var/mail
```

4. ファイルシステムを永続的にリモートアクセスに使用できるようにします。

   /etc/dfs/dfstab を編集し、2. で使用したコマンド行を追加します。

```
# cat /etc/dfs/dfstab
 ..
share -F nfs -o rw /var/mail
```

注 - mail.local プログラムは、メッセージが初めて配信されたときに /var/mail ディレクトリでメールボックスを自動的に作成します。メールクライアントのためにメールボックスを作成する必要はありません。

▼ メールクライアントを設定する方法

メールクライアントは、メールボックスがメールサーバーにあり、/etc/mail/aliases ファイルのメール別名がメールボックスの位置を指しているメールサービスのユーザーです。

1. メールクライアントのシステム上でスーパーユーザーになります。

2. メールクライアントのシステムで /var/mail マウントポイントがあることを確認します。

   ls を使用するとファイルシステムが存在するかどうかがわかります。次の例はファイルシステムが作成されていない場合の応答を示しています。

```
# ls -l /var/mail
/var/mail not found
```

   メールファイルがこのディレクトリにある場合は、ファイルを移動させておくと、サーバーから /var/mail ディレクトリがマウントされるときにその対象となりません。

3. メールサーバーから /var/mail ディレクトリをマウントします。

   このメールディレクトリは自動的にマウントさせることも、起動時にマウントさせることもできます。

   a. /var/mail を自動的にマウントさせる場合

      /etc/auto_direct を編集し、次のようなエントリを加えます。

```
/var/mail -rw,hard,actimeo=0 server:/var/mail
```

   b. 起動時に /var/mail をマウントする場合

/etc/vfstab ファイルを編集し、/var/mail ディレクトリのためのエントリを追加して、そのディレクトリをローカルの /var/mail ディレクトリにマウントします。

```
server:/var/mail - /var/mail nfs - no rw,hard,actimeo=0
```

システムをリブートするたびに、クライアントのメールボックスが自動的にマウントされます。クライアントのメールボックスをマウントするには、システムをリブートするまでは、mountall と入力します。

注意 - NFS サーバーからメールをマウントするときは、メールボックスのロックとアクセスが適切に行われるように、actimeo=0 オプションを含める必要があります。

4. /etc/hosts を編集します。

   admintool を使用して /etc/hosts ファイルを編集し、メールサーバーのためのエントリを追加します。ネームサービスを使用している場合はこの作業は必要ありません。

5. 別名ファイルの 1 つにクライアントのエントリを追加します。

   異なるタイプのメール構成に対してメール別名を作成する方法については、750 ページの「メール別名ファイルの管理」を参照してください。

   注 - mail.local プログラムは、メッセージが初めて配信されるときに /var/mail ディレクトリでメールボックスを自動的に作成します。メールクライアントのために個々のメールボックスを作成する必要はありません。

6. **sendmail** を再起動します。

▼ メールホストを設定する方法

メールホストは、電子メールアドレスを解決し、ドメイン内でメールを再度ルーティングします。メールホストに適しているシステムは、ドメイン外または親ドメインに接続されているシステムです。

1. メールホストシステムでスーパーユーザーになります。

2. ホスト名の構成を確認します。

   次のように check-hostname スクリプトを実行し、sendmail が、このサーバーの完全指定ホスト名を識別できるかどうかを確認します。

```
% /usr/lib/mail/sh/check-hostname
hostname phoenix OK: fully qualified as phoenix.eng.acme.com
```

   このスクリプトによる完全指定ホスト名の識別ができなかった場合は、完全指定ホスト名を、/etc/hosts 内のホストの最初の別名として追加する必要があります。

3. /etc/hosts を更新します。

   admintool を使用して /etc/hosts ファイルを更新します。メールホストシステムの IP アドレスとシステム名のあとに mailhost および mailhost.*domainname* を入れます。このシステムはメールホストとして指定されます。この *domainname* は、次のコマンドの出力にサブドメイン名として指定されてる文字列と同じにする必要があります。

```
% /usr/lib/sendmail -bt -d0 </dev/null
Version 8.9.0+Sun
 Compiled with: MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7 NAMED_BIND
                NDBM NETINET NETUNIX NEWDB NIS NISPLUS QUEUE SCANF SMTP
                USERDB XDEBUG

============ SYSTEM IDENTITY (after readcf) ============
      (short domain name) $w = phoenix
  (canonical domain name) $j = phoenix.eng.acme.com
         (subdomain name) $m = eng.acme.com
              (node name) $k = phoenix
========================================================
```

   上記の変更のあとのホストファイルの例を示します。

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1       localhost
129.0.0.1       phoenix mailhost mailhost.eng.acme.com        loghost
```

4. ホストファイルに新規メールホストのエントリを作成します。

   NIS または NIS+ を使用する場合は、mailhost および mailhost.*domainname* と呼ばれるホスト別名を含むエントリを、新規メールホストのホストエントリに追加します。

   NIS または NIS+ を使用しない場合は、ネットワーク上の各システム用に、/etc/hosts にエントリを作成する必要があります。エントリでは、必ず *IP_address mailhost_name* mailhost mailhost.*domainname* というフォーマットを使用してください。

5. 正しい構成ファイルを変更します。

   このコマンドで /etc/mail/main.cf ファイルをコピーし、ファイルの名前を変更します。

```
# cp /etc/mail/main.cf /etc/mail/sendmail.cf
```

6. sendmail を再起動し、メール構成をテストします。

   詳細は、762ページの「メール構成をテストする方法」を参照してください。

▼ メールゲートウェイを設定する方法

メールゲートウェイは、ドメイン外のネットワークとの通信を管理します。送信側メールゲートウェイ上のメールプログラムは、受信側システムのメールプログラムと同じでなければなりません。

メールゲートウェイに適しているシステムは、Ethernet および電話回線に接続されているシステムか、またはインターネットへのルーターとして設定されているシステムです。メールホストをメールゲートウェイとして設定するか、または別のシステムをメールゲートウェイとして設定できます。複数のメールゲートウェイを自分

のドメイン用として設定できます。UUCP 接続がある場合は、UUCP 接続をメールゲートウェイとするシステム (1 つまたは複数のシステム) を設定してください。

1. メールゲートウェイでスーパーユーザーになります。

2. 構成ファイルを変更します。

   次のコマンドは `main.cf` ファイルをコピーし名前を変更します。

```
# cp /etc/mail/main.cf /etc/mail/sendmail.cf
```

3. ホスト名の構成を確認します。

   次のように `check-hostname` スクリプトを実行し、`sendmail` が、このサーバーの完全指定のホスト名を識別できるかどうかを確認します。

```
# /usr/lib/mail/sh/check-hostname
hostname phoenix OK: fully qualified as phoenix.eng.acme.com
```

   このスクリプトによる完全指定のホスト名の識別ができなかった場合は、完全指定のホスト名を、`/etc/hosts` 内のホストの最初の別名として追加する必要があります。

4. `sendmail` を再起動し、メール構成をテストします。

762ページの「メール構成をテストする方法」を参照してください。

▼ sendmail で DNS を使用する方法

DNS ネームサービスは個人別の別名をサポートしませんが、メール交換局 (MX) レコードおよび cname レコードを使用して、ホストおよびドメイン用の別名はサポートします。ホスト名とドメイン名は両方またはいずれか一方を DNS データベースで指定することができます。DNS の管理の詳細については、『*Solaris* ネーミングの設定と構成』を参照してください。

1. スーパーユーザーになります。

2. **DNS** ホストルックアップ機能を有効にします **(NIS** のみ**)**。

/etc/nsswitch.conf ファイルを編集し、dns フラグを含むホストの定義から # を削除します。使用される DNS ホスト別名のために、ホストエントリには次に示すように dns フラグを入れる必要があります。

```
# grep hosts /etc/nsswitch.conf
#hosts:      nisplus [NOTFOUND=return] files
hosts:      nisplus dns [NOTFOUND=return] files
```

3. mailhost と mailhost.*domainname* エントリを確認します。

   DNS データベース内に mailhost と mailhost.*domainname* のためのエントリが存在することを確認してください。

sendmail 構成ファイルの構築

sendmail 構成ファイルを作成するプロセスが変わりました。多くのサイトについて、この変更により構成ファイルの管理が容易になります。sendmail.cf ファイルの古いバージョンはまだ使用できますが、適当な時期に新しいシステムに移行することをお勧めします。新しいプロセスの詳細は、/usr/lib/mail/README で説明します。

▼ 新しい sendmail.cf ファイルを構築する方法

1. スーパーユーザーになります。

2. 変更しようとする構成ファイルのコピーを作成します。

```
# cd /usr/lib/mail/cf
# cp main-v7sun.mc myhost.mc
```

3. 必要に応じて新しい構成ファイルを編集します (たとえば、*myhost*.mc など)。

4. m4 を使用して構成ファイルを構築します。

```
# /usr/ccs/bin/make myhost.cf
```

5. –C オプションを使用して、新しい構成ファイルをテストし、新しいファイルを指定します。

```
# /usr/lib/sendmail -C myhost.cf -v testaddr </dev/null
```

このコマンドは testaddr にメッセージを送り、その一方で稼動時にメッセージを表示します。システム上で sendmail サービスを再起動せずに、送信メールだけがテストできます。まだメールを処理していないシステムでは、762ページの「メール構成をテストする方法」に記載される、完全なテスト手順を使用してください。

6. オリジナルのコピーを作成した後に新しい構成ファイルをインストールします。

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.save
# cp myhost.cf /etc/mail/sendmail.cf
```

7. sendmail サービスを再起動します。

```
# pkill -HUP sendmail
```

メール別名ファイルの管理

メール別名はドメイン独自にする必要があります。この節ではコマンド行を使用してメール別名テーブル内の別名を検索し、NIS+、NIS に対して、またはローカルシステム上で、メール別名を作成する方法について説明します。また、AdminTool のデータベースマネージャアプリケーションを使用して別名データベース上でこれらの作業を実行することもできます。

また、データベースファイルは、makemap を使用してローカルメールホスト用に作成できます。これらのデータベースファイルの使用により、NIS または NIS+ などの名前空間を使用する場合の利点をすべて提供できるわけではありませんが、データの検索速度は、ローカルファイルを使用するよりも速くなります。

▼ NIS+ 別名テーブルの内容を表示する方法

aliasadm コマンドを使用するには、root、mail_aliases テーブルを所有する NIS+ グループのメンバー、またはテーブルを作成したユーザーでなければなりません。

例 – NIS+ mail_aliases テーブル全体を表示する場合

♦ aliasadm -l と入力して Return キーを押します。

これにより、別名によるアルファベット順に別名テーブルの内容が表示されます。

注 - 大きな別名テーブルがある場合、全体の内容の表示には、やや時間がかかることがあります。特定のエントリを検索する場合は、grep 検索機能を使用して特定のエントリを探し出すことができるように、出力をパイプを通して grep コマンドに入力してください (aliasadm -l | grep *entry*)。

例 – NIS+ mail_aliases テーブル内の個々のエントリを表示する場合

♦ aliasadm -m ***alias*** と入力して Return キーを押します。

別名エントリが表示されます。

```
# aliasadm -m ignatz
ignatz: ignatz@saturn # Alias for Iggy Ignatz
```

注 - aliasadm -m オプションは、別名が完全に同じものだけを表示します。部分的に同じ文字列は表示しません。aliasadm -m オプションではメタキャラクタ (* および ? など) は使用できません。部分的に一致する文字列も表示したい場合は、aliasadm -l | grep *partial-string* と入力して Return キーを押してください。

▼ コマンド行から NIS+ mail_aliases テーブルへ別名を追加する方法

1. 各メールクライアント、各メールクライアントのメールボックスの位置、およびメールサーバーシステムの名前のリストを編集します。
2. 任意のシステム上のスーパーユーザーになります。
3. **(省略可能) :** 必要に応じて、**NIS+** テーブルを作成します。

   まったく新しい NIS+ mail_aliases テーブルを作成する場合は、最初に NIS+ テーブルを初期設定しなければなりません。aliasadm -I と入力して Return キーを押します。
4. それぞれの別名について、aliasadm -a ***alias expanded_alias*** [***options comments***] と入力して Return キーを押します。

   これにより、別名が NIS+ mail_aliases テーブルに追加されます。

```
# aliasadm -a iggy iggy.ignatz@saturn "Iggy Ignatz"
```

5. aliasadm -m ***alias*** と入力して Return キーを押します。

   これにより、作成したエントリが表示されます。
6. エントリが正しいことを確認します。

▼ NIS+ mail_aliases テーブルを編集してエントリを追加する方法

2 つまたは 3 つ以上の別名を追加する場合は、NIS+ テーブルを直接編集することもあります。

1. メールクライアント、メールボックスの位置、およびメールサーバーシステムの名前の各リストをコンパイルします。

2. 任意のシステムでスーパーユーザーになります。

3. `aliasadm -e` と入力して `Return` キーを押します。

   別名テーブルは、`$EDITOR` 環境変数で設定されているエディタを使用して表示されます。変数が設定されていなければ、`vi` がデフォルトのエディタです。

4. 次のフォーマットを使用してそれぞれの別名を別の行に入力します。

   a. 任意の順序でテーブルの任意の位置に別名を入力します。

      順序は NIS+ mail_aliases 別名テーブルにおいては任意です。`aliasadm -l` コマンドがリストをソートし、アルファベット順に表示します。

   b. フォーマット ***alias*: *expanded_alias*** `#["`***option***`"# "`***comments***`"]` を使用します。

      オプション列をブランクにする場合は、空の引用符 2 つ (`""`) を入力し、次にコメントを追加します。

   c. `Return` キーを押すことにより、それぞれの行を終了させます。

5. エントリが正しいことを確認します。

6. 変更を保存します。

▼ NIS+ mail_aliases テーブルのエントリを変更する方法

1. 任意のシステムでスーパーユーザーになります。

2. `aliasadm -m` ***alias*** と入力して `Return` キーを押します。

   別名の情報が表示されます。

3. `aliasadm -c` ***alias expanded_alias*** `[`***options comments***`]` と入力して `Return` キーを押します。

別名は、入力した新しい情報を使用して変更されます。

4. `aliasadm -m` ***alias*** と入力して `Return` キーを押します。
   作成したエントリが表示されます。

5. エントリが正しいことを確認します。

▼ NIS+ `mail_aliases` テーブルからエントリを削除する方法

1. 任意のシステムでスーパーユーザーになります。

2. `aliasadm -d` ***alias*** と入力して `Return` キーを押します。
   別名が NIS+ `mail_aliases` テーブルから削除されます。

▼ NIS `mail.aliases` マップを設定する方法

1. メールクライアント、メールボックスの位置、およびメールサーバーシステムの名前の各リストをコンパイルします。

2. **NIS** マスターサーバーでスーパーユーザーになります。

3. `/etc/mail/aliases` ファイルを編集し、次のようなエントリを作成します。
   a. メールクライアントごとにエントリを追加します。

   b. エントリ `Postmaster: root` をポストマスターとして指定された個人のメールアドレスに変更します。
      詳細は、757ページの「ポストマスター別名を設定する」を参照してください。

   c. メールサーバーの管理のためのメールボックスを作成した場合は、`root:`***mailbox@mailserver*** のエントリを作成します。

   d. 変更を保存します。

4. **NIS** マスターサーバーの /etc/hosts ファイルを編集し、メールサーバーごとにエントリを作成します。

5. **cd /var/yp** と入力して Return キーを押します。

6. **make** と入力して Return キーを押します。

   /etc/hosts と /etc/mail/aliases ファイルでの変更が NIS スレーブシステムに転送されます。別名が有効になるのに、数分かかることがあります。

▼ ローカルメール別名ファイルを設定する方法

1. メールクライアントとメールボックスの位置の各リストをコンパイルします。

2. メールサーバーでスーパーユーザーになります。

3. /etc/mail/aliases ファイルを編集し、次のようなエントリを作成します。

   a. メールクライアントごとにエントリを追加します。

   b. エントリ Postmaster: root をポストマスターとして指定された個人のメールアドレスに変更します。

      詳細は、757ページの「ポストマスター別名を設定する」を参照してください。

   c. メールサーバーの管理のためのメールボックスを作成した場合は、root: ***mailbox@mailserver*** のエントリを作成します。

   d. 変更を保存します。

4. newaliases と入力して Return キーを押します。

   これにより、sendmail が使用できるバイナリ形式で alias ファイルが作成されます。ファイルは、/etc/mail/aliases.dir と /etc/mail/aliases.pag ファイルに格納されます。

5. /etc/mail/aliases、/etc/mail/aliases.dir、および /etc/mail/aliases.pag ファイルを他の各システムにコピーします。

3 つのファイルをすべてコピーしたら、newaliases コマンドを他の各システムで実行する必要はありません。

rcp または rdist コマンドを使用してファイルをコピーするか、またはこの目的のために作成したスクリプトを使用してファイルをコピーできます。メールクライアントを追加または削除するたびにすべての /etc/mail/aliases ファイルを更新しなければならないので注意してください。

▼ キー付きマップファイルの作成方法

1. 選択したエディタを使用して、入力ファイルを作成します。

   エントリは以下のようになります。

```
sandy@newdomain.com     ssmith@newdomain.com
ssmith@olddomain.com    error:nouser No such user here
@olddomain.com          %1@newdomain.com
```

   この例では、最初のエントリにより、メールは新しい別名に転送されます。2 番目のエントリにより、不適切な別名が使用された時にメッセージが作成されます。さらに、最後のエントリにより、すべての着信メールは olddomain から newdomain へ転送されます。

2. データベースファイルを作成します。

```
# /usr/sbin/makemap -o dbm newmap < newmap
```

-o	ファイルを上書きする代わりに追加する。使用できるオプションのリストは、makemap(1M) のマニュアルページを参照
dbm	dbm データベースタイプを選択する。この他のマップタイプには、btree または hash がある
newmap	入力ファイル名とデータベースファイル名の最初の部分。dbm データベースタイプを選択すると、データベースファイルは接尾辞に .pag または .dir を使用して作成される。他の 2 つのデータベースタイプの場合、ファイル名には .db が付く

ポストマスター別名を設定する

各システムは postmaster メールボックスにメールを送信できなければなりません。postmaster の NIS または NIS+ 別名を作成するか、または各ローカル /etc/mail/aliases ファイルで別名を作成できます。次に、デフォルトの /etc/mail/aliases エントリを示します。

```
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's
# mail problems.
Postmaster: root
```

postmaster の別名を作成するには、各システムの /etc/mail/aliases ファイルを編集し、root をポストマスターとして機能する個人のメールアドレスに変更します。

ポストマスターがポストマスターメールと個人的メールとを区別するために、別のメールボックスを作成できます。別のメールボックスを作成する場合は、/etc/mail/aliases ファイルを編集するときに、ポストマスターのメールアドレスではなくメールボックスアドレスを使用してください。

▼ postmaster 用に別のメールボックスを作成する方法

1. postmaster として指定された個人のアカウントを作成し、アスタリスク (*) をパスワードフィールドに入れます。

2. メールが配信されたら mail -f ***postmaster*** と入力して Return キーを押します。

   mail プログラムはメールボックス名を読んだり、書き込んだりできます。

▼ postmaster メールボックスを別名に追加する方法

1. 各システムでスーパーユーザーになり、/etc/mail/aliases ファイルを編集します。

   ネットワークが NIS または NIS+ を実行しない場合は、/etc/mail/aliases ファイルを編集します。

2. ポストマスター別名を root から Postmaster: *postmastermailbox@postmasterhost* に変更し、その変更を保存します。

3. ポストマスターのローカルシステムで、別名の名前 (たとえば、sysadmin) を定義する /etc/mail/aliases ファイルにエントリを作成し、ローカルメールボックスへのパスを入れます。

4. **newaliases** と入力して Return キーを押します。

   あるいは、aliases ファイルで Postmaster: エントリを Postmaster: */usr/somewhere/somefile* エントリに変更することもできます。

メール待ち行列の管理

この節では、メールサービスをスムーズに動作させる方法について説明します。

▼ メール待ち行列を表示する方法

待ち行列の内容は mailq で印刷できます。このコマンドは sendmail に –bp フラグを指定するのと同じです。

- **/usr/bin/mailq | more** と入力して Return キーを押します。

待ち行列 ID のリスト、メッセージのサイズ、メッセージが待ち行列に入れられた日付、メッセージの状態、および発信者と受信者が表示されます。

▼ メール待ち行列を強制処理する方法

- **/usr/lib/sendmail -q -v** と入力して Return キーを押します。

これで待ち行列の処理が強制的に行われ、待ち行列の処理中にジョブの進行状況が表示されます。

▼ メール待ち行列のサブセットを実行する方法

- **/usr/lib/sendmail -qR***string* と入力して Return キーを押します。

-qR*string* (どれかの受信者名が *string* に一致する場合に待ち行列を実行) または -qI*nnnnn* (待ち行列 ID*nnnnn* の 1 つのメッセージを実行) でいつでも待ち行列のサブセットを実行できます。*string* はホスト名とも一致することができるので、*user@host.domain* のサブ文字列も一致します。

この例では、受信者 wnj の待ち行列にあるものをすべて処理します。

```
# /usr/lib/sendmail -qRwnj
```

▼ メール待ち行列を移動する方法

1. メールホストでスーパーユーザーになります。

2. **/etc/init.d/sendmail stop** と入力して Return キーを押します。

   これで古い sendmail デーモンは削除されるので、古い待ち行列ディレクトリが処理されることはありません。

3. **cd /var/spool** と入力して Return キーを押します。

4. **mv mqueue omqueue; mkdir mqueue** と入力して Return キーを押します。

   これでディレクトリの mqueue とその内容のすべてが omqueue ディレクトリに移動し、新規の空の Rmqueue ディレクトリを作成します。

5. **chmod 755 mqueue; chown daemon.daemon mqueue** と入力して Return キーを押します。

   これらのコマンドでディレクトリのアクセス権を設定し、所有者による読み込み、書き込み、実行、またグループや他のユーザーによる読み込み、実行が行えるようにします。またこれらのコマンドでは、所有者やグループを daemon に設定します。

6. **/etc/init.d/sendmail start** と入力し Return キーを押します。

   これで新規 sendmail デーモンが起動します。

▼ 古いメール待ち行列を処理する方法

1. **/usr/lib/sendmail -oQ/var/spool/omqueue -q** と入力して Return キーを押します。

-oQ フラグは代替待ち行列ディレクトリを指定し、-q フラグは待ち行列での各ジョブを処理するように指示します。詳細 (verbose) 表示にしたい場合は、-v フラグを使用します。

2. 待ち行列が最後に空になったら、**rmdir /var/spool/omqueue** と入力して Return キーを押します。

   これにより空のディレクトリが削除されます。

.forward ファイルの管理

この節では、.forward ファイルの管理に関する複数の手順を説明します。これらファイルはユーザーが編集できるので、ファイルが問題の原因になる場合があります。

▼ .forward ファイルを無効にする方法

この手順は特定のホスト用の .forward ファイルだけを無効にします。

1. スーパーユーザーになります。

2. /usr/lib/mail/domain/solaris-generic.m4 またはサイト固有のドメイン m4 ファイルのコピーを作成します。

```
# cd /usr/lib/mail/domain
# cp solaris-generic.m4 mydomain.m4
```

3. 次の行を作成したファイルに追加します。

```
define(‘confFORWARD_PATH’,‘’)dnl
```

   使用中のドメイン m4 ファイルにすでにこの行が存在する場合には、行を置き換えます。

4. 新しい構成ファイルを構築してインストールします。

   完全な手順については、749ページの「新しい sendmail.cf ファイルを構築する方法」を参照してください。

▼ .forward ファイルの検索パスを変更する方法

1. スーパーユーザーになります。

2. /usr/lib/mail/domain/solaris-generic.m4 またはサイト固有のドメイン m4 ファイルのコピーを作成します。

```
# cd /usr/lib/mail/domain
# cp solaris-generic.m4 mydomain.m4
```

3. 作成したファイルに次のような行を追加します。

```
define(‘confFORWARD_PATH’,‘~z/.forward:/var/forward/$u’)dnl
```

   使用中のドメイン m4 ファイルにすでにこの行が存在する場合には、行を置き換えます。

4. 新しい構成ファイルを構築してインストールします。

   完全な手順については、749ページの「新しい sendmail.cf ファイルを構築する方法」を参照してください。

▼ /etc/shells の作成および生成方法

このファイルは標準のリリースには含まれていないので、プログラムまたはファイルにメールを転送するためにユーザーが .forward ファイルを使用できるようにする場合には、追加する必要があります。このファイルは、grep を使用し、パスワード内にリストされたすべてのシェルを特定した後に、これらのシェルを入力してファイルを作成できますが、ダウンロードして入手できるスクリプトを使用した以下の手順を使用すると、より簡単に作成できます。

1. http://www.sendmail.org/sun-specific/gen-etc-shells.html からスクリプトをダウンロードします。

2. スーパーユーザーになります。

3. シェルのリストを作成するために、gen-etc-shells を実行します。

```
# ./gen-etc-shells.sh > /tmp/shells
```

   このスクリプトでは、getent コマンドを使用して、/etc/nsswitch.conf 内にリストされたパスワードファイルソースに組み込まれたシェルの名前を収集します。

4. /tmp/shells 内のシェルのリストを調べます。

   選択したエディタを使用し、組み込みたくないシェルを削除します。

5. ファイルを /etc/shells に移動します。

```
# mv /tmp/shells /etc/shells
```

メールに関する問題解決のヒント

この節では、メールサービスの問題解決に使用できるヒントとツールをいくつか示します。

▼ メール構成をテストする方法

1. 構成ファイルを変更したシステム上で、**sendmail** を再起動します。

```
# pkill -HUP sendmail
```

2. **/usr/lib/sendmail -v** *names* **</dev/null** と入力して Return キーを押すことにより、各システムからテストメッセージを送信します。

   names には、受信者の電子メールアドレスを指定します。

このコマンドは、指定された受信者に空のメッセージを送信し、動作している間メッセージを表示します。

3. 次のテストを実行します。

   a. メッセージを通常のユーザー名に送ることによってメールを自分自身またはローカルシステム上の他の人に送信します。

   b. **Ethernet** の場合は、別のシステムの誰かにメールを送信します。
   メインシステムからサブシステムへ、サブシステムからメインシステムへ、およびサブシステムから別のサブシステムへの 3 つの方向で送信します。

   c. メールゲートウェイがある場合、メールホストから別のドメインにメールを送信してリレーメールプログラムおよびホストが適切に設定されていることを確認します。

   d. 電話回線上に別のホストへの **UUCP** 接続を設定している場合は、そのホストの誰かにメールを送信し、その個人にメールを返信してもらうか、またはその個人がメッセージを受信したときに電話してもらいます。

   e. **UUCP** 接続を介してメールを送信するように他の人に頼みます。
   `sendmail` プログラムでは、メッセージが受信されたかどうかは通知されません。これは、メッセージが配信のために UUCP に渡されるためです。

   f. 異なるシステムの `postmaster` にメッセージを送り、自分のポストマスターのメールボックスに送られることを確認します。

▼ メール別名を確認する方法

別名と受信者にメールを配信できるかどうかを調べるには、次のようにします。

- **/usr/lib/sendmail -v -bv** *recipient* と入力して `Return` キーを押します。

このコマンドは別名を表示し、最終アドレスが配信可能かどうかを識別します。

次に出力例を示します。

```
% /usr/lib/sendmail -v -bv sandy
sandy... aliased to   ssmith
```

```
ssmith... aliased to              sandy@phoenix
sandy@phoenix... deliverable: mailer esmtp, host phoenix, user sandy@phoenix.eng.acme.com
%
```

ローカルとドメイン全体で有効な別名を両方使用するときには、ループしたりデータベースの一貫性が失われたりしないように十分に注意する必要があります。ユーザーをあるシステムから別のシステムに移動するときは、別名のループを作成しないように特に注意してください。

▼ sendmail ルールセットをテストする方法

1. **/usr/lib/sendmail -bt** と入力して Return キーを押します。
   情報が表示されます。
2. 最後のプロンプト (>) で、**3,0** とテストしたいメールアドレスを入力します。
3. Control-d キーを押してセッションを終了します。

次に出力例を示します。

```
% /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 sandy@phoenix
rewrite: ruleset   3   input: sandy @ phoenix
rewrite: ruleset  96   input: sandy < @ phoenix>
rewrite: ruleset  96 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset   3 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset   0   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 199   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 199 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset  98   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset  98 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 198   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 198 returns: $# local $: sandy
rewrite: ruleset   0 returns: $# local $: sandy
```

▼ 他のシステムへの接続を調べる方法

他のシステムへの接続を調べるには、mconnect プログラムを使用してネットワーク上のその他の sendmail システムへの接続をオープンします。mconnect プログラムは対話方式で動作します。さまざまな診断コマンドを実行できます。詳細は、mconnect(1) のマニュアルページを参照してください。次の例では、ユーザー名 shamira へのメールが配信可能かどうかを調べます。

```
$ mconnect phoenix
connecting to host phoenix (129.144.52.96), port 25
connection open
220 phoenix.Eng.Acme.COM Sendmail 8.9.0+Sun/8.9.0; Tue, 25 Jul 1998 10:45:28 -0700
vrfy sandy
250 Sandy Smith <sandy@phoenix.Eng.Acme.COM>
>
```

mconnect を使用して SMTP ポートに接続できない場合は、次の条件を確認してください。

- システム負荷が高すぎないか
- sendmail デーモンが動作しているか
- システムに適切な /etc/mail/sendmail.cf ファイルがあるか
- ポート 25 (sendmail が使用するポート) がアクティブであるか

システムログ

メールサービスは、syslogd プログラムを使用してほとんどのエラーを記録します。デフォルトでは、syslogd は loghost にメッセージを送ります。

/etc/hosts ファイルの loghost というシステムを、NIS ドメイン全体のすべてのログを管理するように定義できます。システムログは syslogd プログラムによってサポートされます。 /etc/hosts で loghost を指定します。loghost を指定しなければ、syslogd からのエラーメッセージはレポートされません。

例 34–1 にデフォルトの /etc/syslog.conf ファイルを示します。

例 **34–1**　デフォルトの /etc/syslog.conf ファイル

```
#ident "@(#)syslog.conf   1.3        93/12/09 SMI"  /* SunOS 5.0 */  #
# Copyright (c) 1994 by Sun Microsystems, Inc.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (‘’) names
# that match m4 reserved words. Also, within ifdef’s, arguments
# containing commas must be quoted.
#
# Note: Have to exclude user from most lines so that user.alert
#       and user.emerg are not included, because old sendmails
#       have no 4.2BSD based systems doing network logging, you
#       can remove all the special cases for "user" logging.
# *.err;kern.debug;auth.notice;user.none         /dev/console
*.err;kern.debug;daemon,auth.notice;mail.crit;user.none /var/adm/messages
*.alert;kern.err;daemon.err;user.none            operator
*.alert;user.none                                root
*.emerg;user.none                                *
# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice         ifdef(‘LOGHOST’, /var/log/authlog, @loghost)
mail.debug           ifdef(‘LOGHOST’, /var/log/syslog, @loghost)
#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(‘LOGHOST’, ,
user.err             /dev/console
user.err             /var/adm/messages
user.alert           ‘root, operator’
user.emerg           *
)
```

/etc/syslog.conf ファイルを編集することにより、デフォルト構成を変更できます。変更内容を有効にするには、syslog デーモンを再起動する必要があります。メールに関する情報収集のため、次の選択項目をファイルに追加できます。

- mail.alert – ここで訂正する必要のある状態メッセージ
- mail.crit – クリティカルメッセージ
- mail.warning – 警告メッセージ
- mail.notice – エラーではないが注意すべきメッセージ
- mail.info – 情報的なメッセージ
- mail.debug – デバッグメッセージ

次のエントリにより、重要なメッセージ、単なる情報としてのメッセージ、およびデバッグメッセージのすべてのコピーが、/var/log/syslog に送られます。

```
mail.crit;mail.info;mail.debug   /var/log/syslog
```

システムログの各行には、タイムスタンプ、ログを生成したシステム名、およびメッセージが入っています。syslog ファイルは、大量の情報を記録できます。

ログは、連続したレベルとして並べられます。最下位レベルでは、異常なイベントだけが記録されます。最上位レベルでは、もっとも必須なイベントと注目する必要のないイベントが記録されます。通常、10 以下のログレベルが「有用」とみなされます。10 を超えるログレベルは通常、デバッグに使用されます。loghost と syslogd プログラムについては、『*Solaris* のシステム管理 *(*第 *2* 巻*)*』を参照してください。

その他のメール診断情報

その他の診断情報については、次の情報源を確認してください。

- メッセージのヘッダーの Received 行を調べます。これらの行は、メッセージがリレーされるときにとった経路を追跡できます。UUCP ネットワークでは、多くのサイトがこれらの行を更新せず、またインターネットでは、行がしばしば再編成されるので注意してください。これらを適切に処理するには、各行の日時を調べてください。時間帯の違いを考慮するのを忘れないでください。
- MAILER-DAEMON からのメッセージを調べます。これらは通常、配信上の問題をレポートします。
- ワークステーショングループの配信上の問題を記録するシステムログを確認します。sendmail プログラムは常に、処理の内容をシステムログに記録します。crontab ファイルを修正してシェルスクリプトを夜間に実行できます。これは、ログから SYSERR メッセージのログを検索し、見つかったものをポストマスターに送信します。
- mailstats プログラムを使用してメールタイプをテストし、受信と送出メッセージの数を判定します。

第 35 章

メールサービスのリファレンス

sendmail プログラムは、構成ファイルを使用して「別名」変換と転送、ネットワークゲートウェイへの自動ルーティング、柔軟な構成を提供するメール転送エージェントです。Solaris オペレーティング環境では、ほとんどのサイトで使用できる標準構成ファイルが付属しています。第 34 章では、標準のファイルを使用して電子メールシステムを設定する方法について説明しています。この章では、sendmail の汎用バージョンと Solaris バージョンのいくつかの相違点について説明します。

Solaris sendmail の相違点

この節では、sendmail の Solaris バージョンに組み込まれたいくつかの変更について、汎用 Berkeley バージョンと比較して説明します。

sendmail のコンパイル時に使用するフラグ

次に、Solaris 8 に添付されている sendmail のバージョンをコンパイルするときに使用するフラグを示します。構成に他のフラグが必要な場合は、そのソースをダウ

ンロードし、バイナリにコンパイルし直してください。この処理については、http://www.sendmail.org に記載してあります。

フラグ	説明
SOLARIS=20800	Solaris 8 オペレーティング環境をサポートする
NDBM	ndbm データベースをサポートする
NEWDB	db データベースをサポートする
NIS	nis データベースをサポートする
NISPLUS	nisplus データベースをサポートする
LDAPMAP	LDAP のマップをサポートする
USERDB	ユーザーデータベースをサポートする
MAP_REGEX	正規表現のマップをサポートする
SUN_EXTENSIONS	Solaris のフラグで、sun_compat.o に組み込まれる Sun の拡張子
VENDOR_DEFAULT=VENDOR_SUN	Solaris のフラグで、Sun をデフォルトのベンダーとして選択する
USE_VENDOR_CF_PATH	Solaris のフラグで、このフラグを使用すると構成ファイルを /etc/mail 内に配置できる
_FFR_MAXALIASRECURSION_OPTION	Solaris のフラグで、このフラグを使用すると MaxAliasRecursion オプションを選択できる
_FFR_MAX_MIME_HEADER_LENGTH	Solaris のフラグで、このフラグを使用すると MaxMimeHeaderLength オプションを選択できる

sendmail の代替コマンド

Solaris リリースには、Berkley による汎用リリースで提供されているコマンドの同義語がすべて組み込まれているわけではありません。表 35–1 には、コマンドの別名

のリストと それが Solaris リリースに組み込まれているかどうか、および sendmail を使用して同じ動作を生成する方法を示しています。

表 35–1 代替 sendmail コマンド

代替名	**Solaris** に組み込まれているか	sendmail を使用したオプション
hoststat	組み込まれていない	sendmail –bh
mailq	組み込まれている	sendmail –bp
newaliases	組み込まれている	sendmail –bi
purgestat	組み込まれていない	sendmail –bH
smtpd	組み込まれていない	sendmail –bd

構成ファイルのバージョンの定義

sendmail の新版 (バージョン 8.9.3) には、sendmail.cf ファイルのバージョンを定義するための、新しい構成オプションがあります。このオプションを使用すれば、旧バージョンの構成ファイルをバージョン 8.9.3 の sendmail で使用できます。バージョンレベルには 0 から 8 の値を設定できます。また、ベンダーの定義もできます。Berkeley または Sun がベンダーとして選択できます。構成ファイルで V オプションが定義されていない場合は、V1/Sun がデフォルトの設定となります。ベンダーを定義せずに、バージョンレベルだけが設定されている場合は、Sun がデフォルトとして使用されます。表 35–2 に有効なオプションを示します。

表 35–2 構成ファイルのバージョン

定義	説明
V1/Sun	ネームサービスのサポートに Solaris の拡張機能を使用する。新バージョンの sendmail でも旧バージョンの構成ファイルを使用することができる。V オプションを何も定義していない場合はこれがデフォルトの設定
V7/Sun	sendmail のバージョン 8.8 に使用
V8/Sun	sendmail のバージョン 8.9.3 に使用。これは、Solaris 8 リリースの事前作成された構成ファイルに組み込まれた設定

表 35–2　構成ファイルのバージョン　続く

メールサービスの関連用語

メールファイルとプログラムに加え、メールサービスを構築するには、その他多数の構成要素が必要です。次の節ではこれらの構成要素と、それらを説明するのに使用する用語の一部を定義します。

最初の節では、メール配信システムのソフトウェア部分を説明するのに使用する用語を定義します。その次の節では、メール構成におけるハードウェアシステムの機能について取り上げます。

メールサービスソフトウェアの関連用語

ここでは、メールシステムのソフトウェアの構成要素について説明します。サービスには次のものがあります。

- メールユーザーエージェント
- メール転送エージェント
- メール配信エージェント

それ以外のソフトウェアの構成要素には、ドメイン名、メールアドレス、メールボックス、そしてメールの別名があります。

メールユーザーエージェント

「メールユーザーエージェント」は、ユーザーと `sendmail` プログラムなどのメール転送エージェントとの間のインタフェースとして機能します。Solaris オペレーティング環境に搭載されているメールユーザーエージェントは、`/usr/bin/mail`、`/usr/bin/mailx`、`$OPENWINHOME/bin/mailtool`、および `/usr/dt/bin/dtmail` です。

メール転送エージェント

「メール転送エージェント」は、メールメッセージのルーティングとメールアドレスの解釈を行います。Solaris オペレーティング環境ソフトウェアの転送エージェントは sendmail です。転送エージェントは次の機能を実行します。

- メールユーザーエージェントからメッセージを受信する
- 宛先アドレスを認識する
- 適切な配信エージェントを選択してメールを配信する
- 他のメール転送エージェントからのメールを受信する

メール配信エージェント

「メール配信エージェント」は、メールの配信プロトコルを実行するプログラムです。Solaris オペレーティング環境に搭載されているメール配信エージェントについては以下に述べます。

- UUCP メール配信エージェントは uux を使用してメールを配信します。
- 標準の Solaris リリースでは mail.local である、ローカルメール配信エージェントを配信します。

メールプログラム

「メールプログラム」は sendmail 独自の用語です。メール配信エージェントはカスタマイズできます。メールプログラムは sendmail によって使用され、カスタマイズしたメール配信エージェントまたはメール転送エージェントの特定のインスタンスを指定します。

ネットワークのすべてのシステムの sendmail.cf ファイルには、1 つ以上のメールプログラムを指定する必要があります。

smtp メールプログラムは SMTP を使用してメッセージを転送します。SMTP はインターネットで使用される標準のメールプロトコルです。SMTP メールヘッダーは次のようになります。

```
To: paul@phoenix.stateu.edu
From: Iggy.Ignatz@eng.acme.com
```

同じドメインの 2 人のユーザー間でメールが送信されると、ヘッダーは次のようになります。

```
To: Irving.Who@eng.acme.com
From: Iggy.Ignatz@eng.acme.com
```

ドメイン外にメールを送信するとき、特にインターネット経由でメールボックスに送信する必要がある場合は、SMTP を使用してください。

uucp-old メールプログラムはメッセージの配信に uux を使用しますが、ヘッダーをドメイン形式のアドレスでフォーマットします。To: 行と Cc: 行は SMTP ヘッダーとほぼ同様にドメインによってフォーマットされます。uucp ヘッダーは次のようになります。

```
To: paul@phoenix.stateu.com
From: ignatz@eng.acme.com
```

ドメイン形式の名前を処理し、理解できるシステムへの UUCP メールには uucp-uudom を使用してください。また、発信者はドメイン形式の名前を処理し、インターネットからの返信を受信できるようにしておく必要があります。

uucp-old メールプログラムはヘッダーでは感嘆符を用いるアドレスを使用します。これはオリジナルのメールプログラムの 1 つであり、ヘッダーは次のようになります。

```
To: edu!stateu!phoenix!paul
From: acme!ignatz
```

sendmail.cf ファイルにメールプログラム仕様を提供して、他のメール配信エージェントを定義できます。メールプログラムに関しては、/usr/lib/mail/README にも記載してあります。

ドメイン名

「ドメイン」は、ネットワークアドレスの命名のためのディレクトリ構造です。電子メールのアドレスにもドメインが使用されています。電子メールのアドレスは、次のようなフォーマットになっています。

```
user@subdomain. ... .subdomain2.subdomain1.top-level-domain
```

アドレスの @ 記号より左の部分はローカルアドレスです。ローカルアドレスには次の情報が含まれます。

- 別のメールトランスポートを使用するルーティング (たとえば、bob::vmsvax@gateway または smallberries%mill.uucp@gateway)
- 別名 (たとえば、iggy.ignatz)

受信側のメールプログラムでアドレスのローカル部分を解釈する必要があります。

アドレスの @ 記号より右の部分は、ローカルアドレスが位置するドメインアドレスを示します。ドットはドメインアドレスの各部分を区切ります。ドメインは、組織、物理的なエリア、地理的な領域などを表します。

ドメインアドレスは大文字と小文字を区別しません。アドレスのドメイン部分で大文字、小文字、またはそれらを混用しても相違はありません。

ドメイン情報の順序は階層的です。つまり、アドレスがローカルであるほど @ 記号に近づきます。

サブドメインの数が多いほど、宛先に関して提供される情報が詳細になります。ファイルシステム階層におけるサブディレクトリがその上のディレクトリの中にあると解釈されるのと同様に、メールアドレス内の各サブドメインは、その右にあるドメインの中にあると解釈されます。

表 35–3 に米国における最上位のドメインを示します。

表 35–3 米国の最上位のドメイン

ドメイン	説明
Com	企業
Edu	教育機関用
Gov	米国の政府機関
Mil	米国の軍事機関
Net	ネットワーク組織
Org	非営利組織

Donnalyn Frey および Rick Adams による『*A Directory of Electronic Mail Addressing and Networks*』(O'Reilly & Associates, Inc., 1993) には、国際的な最上位のドメインアドレスリストが載っており、定期的に更新されています。

メールの配信においては、名前空間のドメイン名とメールドメイン名は一致しないことがあります。しかし、DNS ドメイン名とメールドメイン名は同じでなければなりません。`sendmail` プログラムは、デフォルトでドメイン名から最初の構成要素を取り除き、メールドメイン名とします。たとえば、NIS+ ドメイン名が `bldg5.eng.acme.com` であれば、そのメールドメイン名は `eng.acme.com` となります。

注 - メールドメインアドレスは大文字と小文字の区別をしませんが、名前空間のドメイン名は異なります。メールと名前空間のドメイン名を設定するときは、小文字を使用するのが最善です。

メールアドレス

「メールアドレス」には、受信者の名前と、メールメッセージが配信されるシステムが含まれます。

ネームサービスを使用しない小さなメールシステムを管理する場合、メールのアドレス指定は簡単です。つまり、ログイン名がユーザーを一意に識別します。

ただし、複数のメールボックスと、複数のドメインを持つ複数のメールシステムを管理する場合、または外部に UUCP (またはその他の) メール接続がある場合は、メールアドレス指定はもっと複雑になります。メールアドレスには「経路依存型」と「経路非依存型」があり、2 つの混用も可能です。経路依存のアドレス指定は、古い仕様に基づいており、ほとんどの場合は必要なく、また望ましくありません。

経路に依存しないアドレス指定

経路に依存しないアドレス指定では、電子メールメッセージの発信者は、受信者の名前と最終の宛先アドレスを指定する必要があります。経路に依存しないアドレスは通常インターネットのような高速ネットワークで使用されます。さらに、新しい UUCP 接続はドメイン形式の名前を頻繁に使用します。経路に依存しないアドレスは次のようなフォーマットになります。

```
user@host.domain
```

UUCP 接続は次のアドレスフォーマットで使用できます。

```
host.domain!user
```

コンピュータのドメイン階層命名方式が普及したため、経路に依存しないアドレスがより一般的になってきました。実際、次に示すように、最も一般的な経路に依存しないアドレスはホスト名を省略し、電子メールメッセージの最終宛先の識別をドメインネームサービスにまかせています。

```
user@domain
```

ルートに依存しないアドレスでは、まず @ 記号を検索し、ドメイン階層を右 (最上位) から左 (@ 記号の右側にある最も固有なアドレス) へと読み取ります。

経路依存のアドレス指定

経路依存のアドレス指定では、電子メールメッセージの発信者が、ローカルアドレス (通常はユーザー名) とその最終の宛先、および最終の宛先に到達するためにメッセージが通らなければならない経路を指定する必要があります。経路依存のアドレスは、UUCP ネットワーク上では一般的に使用され、フォーマットは次のとおりです。

```
path!host!user
```

電子メールアドレスの一部に感嘆符がある場合は、常に経路のすべて (またはその一部) が発信者によって指定されています。経路依存のアドレスは常に左から右に読みます。

この場合、電子メールアドレスは、次のようになります。

```
venus!acme!sierra!ignatz
```

これは、ignatz というユーザーに送信されたメールは、venus というシステムにまず送られ、それに引き続いて、acme、sierra に転送されることを示しています (これはあくまでも実在する経路ではないので注意してください)。4 つのメールハンドラのいずれかが機能しないときは、メッセージは遅れるか、配信できないとして戻されます。

uucp メールプログラムを通してメールが送信される場合、アドレス指定は経路依存に制限されません。uucp メールプログラムによっては、経路に依存しないアドレス指定も処理します。

メールボックス

「メールボックス」は、電子メールメッセージの最終宛先であるメールサーバー内のファイルです。メールボックスの名前は、ユーザー名、またはポストマスターのような特定の職務を持つ人にメールを届ける場所の名前でもかまいません。メールボックスは、ユーザーのローカルシステムかリモートのメールサーバーのいずれかの /var/mail/*username* ファイルにあります。ただし、いずれの場合でも、メールボックスはメールが配信されるシステム上にあります。

ユーザーエージェントがメールスプールからメールを取り出し、ローカルメールボックスに容易に格納できるように、メールは常にローカルファイルシステムに配信される必要があります。ユーザーのメールボックスの宛先として、NFS でマウントされたファイルシステムを使用しないでください。特にリモートサーバーから /var/mail ファイルシステムをマウントしているメールクライアントには、直接メールを送信しないでください。この場合ユーザー宛のメールは、クライアントのホスト名ではなく、メールサーバーにアドレス指定する必要があります。 NFS でマウントされたファイルシステムは、メールの配信と処理に問題を起こすことがあります。/var/mail を NFS でマウントしたクライアントは「リモートモード」となり、サーバーにメールの送信と受信を行うように要求を出します。

/etc/mail/aliases ファイルと NIS や NIS+ といったネームサービスは、電子メールのアドレスに別名を作成するメカニズムを持っているため、ユーザーは、ユーザーのメールボックスの正確なローカル名を知る必要はありません。

表 35–4 に、特殊な目的のメールボックスに対する共通の命名規則をいくつか示します。

表 35–4 メールボックス名のフォーマットについての規則

フォーマット	説明
username	多くの場合、ユーザー名はメールボックス名と同じ
Firstname.Lastname *Firstname_Lastname* *Firstinitial.Lastname* *Firstinitial_Lastname*	ユーザー名は、ドット (または下線) でファーストネームとラストネームに区切ったフルネームか、またはファーストネームがイニシャルで、ドット (または下線) でイニシャルとラストネームを区切ったもの

表 35–4 メールボックス名のフォーマットについての規則 続く

フォーマット	説明
postmaster	ユーザーは、postmaster のメールボックスに質問を送ったり、問題点を報告したりできる。通常は各サイトとドメインに postmaster メールボックスがある
MAILER-DAEMON	sendmail は、MAILER-DAEMON 宛てのメールを自動的にポストマスターに送る
aliasname-request	-request で終わる名前は、配布リストの管理アドレス。このアドレスは、配布リストを管理する人にメールをリダイレクトする
owner-*aliasname*	owner- で始まる名前は、配布リストの管理アドレス。このアドレスは、メールエラーを処理する人にメールをリダイレクトする
owner-owner	この別名は、エラーを戻す先の owner-*aliasname* の別名がない場合に使用される。このアドレスは、メールエラーを処理する人にメールをリダイレクトし、大量の別名を管理する任意のシステムで定義される
local%*domain*	パーセント記号 (%) は、メッセージがその宛先に着くと展開されるローカルアドレスを示す。ほとんどのメールシステムは、% 記号つきのメールボックス名を全メールアドレスとして翻訳する。% は @ と置き換えられ、メールはそれに応じてリダイレクトされる。多くの人が % を使用するが、これは正式な標準ではない。電子メールの世界では「パーセントハック」と呼ばれている。この機能は、メールに問題が起こった場合にデバッグに使用されることが多い

バージョン 8 から、所有者別名が存在する場合には、グループ別名に送信されたメールの封筒の送信者は、所有者別名から拡張されたアドレスに変更されるようになりました。この変更によって、メールエラーは、送信者に返送されるのではなく、別名の所有者に送信されるようになりました。別名に送信したメールは、配信時に、別名の所有者から来たようにみえます。つまり、別名宛てではなく、直接返信が必要な場合には、ユーザーは自らを識別するように注意する必要があります。次の別名のフォーマットは、この変更に関連したいくつかの問題に対応します。

```
mygroup: :include:/pathname/mygroup.list
owner-mygroup: mygroup-request
mygroup-request: sandys, ignatz
```

この例では、mygroup の別名が、このグループの実際のメール別名です。owner-mygroup の別名は、エラーメッセージを受信します。mygroup-request の別名は、管理の要求に使用してください。この構造

は、mygroup の別名に送信されたメールでは、封筒の送信者が mygroup-request に変更されることを意味します。

メールの別名

別名 (alias) とは、もう 1 つの別の名前を指します。電子メールでは、メールボックスの位置を割り当てたり、メールリストを定義したりするために別名を使用できます。

大きなサイトでは通常、メール別名は、メールボックスの位置を定義します。メール別名を作成するのは、企業で個人のアドレスの一部としてメールストップを設定するのと似ています。メールストップを提供しない場合は、メールは中央アドレスに配信されます。建物内のどこにメールを配信するかを決定するには、別の作業が必要となり、ミスの可能性が増えます。たとえば、同じ建物に Kevin Smith という名前の人が 2 人いる場合、どちらの Kevin も、別の Kevin 宛のメールを受け取る可能性が高くなります。

メールリストを作成するときは、なるべくドメインの位置に依存しないアドレスを使用してください。別名ファイルの移植性と柔軟性を高めるため、別名エントリをできる限り一般的でシステムに依存しない形式にしてください。たとえば、システム mars のドメイン eng.acme.com に ignatz というユーザー名がある場合、別名は ignatz@mars ではなく、ignatz@eng としてください。ユーザー ignatz がシステム名を変更しても、eng ドメインには存在し続ける場合、システム名の変更を反映するように別名ファイルを更新する必要はありません。

別名エントリを作成するときは、1 行ごとに 1 つの別名を入力します。ユーザーのシステム名を含むエントリは 1 つだけにしてください。たとえば、ユーザー ignatz には、次のエントリを作成できます。

```
ignatz: iggy.ignatz
iggyi: iggy.ignatz
iggy.ignatz: ignatz@mars
```

ローカル名やドメインに別名を作成できます。たとえば、システム mars にメールボックスがあり、ドメイン planets 内のユーザー fred の別名エントリでは、NIS+ 別名テーブルに次のエントリを作成できます。

```
fred: fred@planets
```

ドメイン外のユーザーを含むメールリストを作成するときは、ユーザー名とドメイン名を持つ別名を作成してください。たとえば、システム privet のドメイン

mgmt.acme.com に smallberries というユーザー名がある場合、別名は smallberries@mgmt.acme.com とします。

送信者の電子メールアドレスは、メールがユーザードメイン外に発信されるときは、完全に修飾されたドメイン名に自動的に変換されます。

別名ファイルの使用

NIS+ mail_aliases テーブル、NIS aliases マップ、または、ローカルの /etc/mail/aliases ファイルでグローバルに使用するメール別名を作成します。また、同じ別名ファイルを使用してメールリストを作成して管理することができます。

メールサービスの構成に応じて、NIS または NIS+ ネームサービスを使用して別名を管理し、グローバル aliases データベースを維持したり、ローカルの /etc/mail/aliases ファイルをすべて同時に更新することにより、別名を同一にできます。

また、ユーザー自身が別名を作成して使用できます。ユーザーは、別名をユーザーだけが使用できるようにローカル ~/.mailrc ファイルで作成することも、誰でも使用できるようにローカル /etc/mail/aliases ファイルで作成することもできます。ユーザーは通常は、NIS または NIS+ 別名ファイルを作成または管理できません。

メール構成のハードウェア要素

メール構成では次の 3 つの要素が必要ですが、これらは同じシステムで組み合わせることも、別のシステムで提供することもできます。

- メールホスト
- メールサーバー (1 つ以上)
- メールクライアント

ユーザーがドメイン外のネットワークと通信をするためには、4 番目の要素であるメールゲートウェイを追加する必要があります。次の節では各ハードウェアコンポーネントについて説明しています。

メールホスト

「メールホスト」は、ネットワーク上でメインのメールマシンとして指定するマシンです。これはサイトにおいて、他のシステムでは配信できないメールを転送するためのマシンになります。hosts データベースにシステムをメールホストとして指定するには、ローカルの /etc/hosts ファイルか、ネームサービスのホストファイルで、IP アドレスの右に mailhost を追加します。メールホストシステムでは、main.cf ファイルもメール構成ファイルとして使用する必要があります。

メールホストとして適切なのは、ローカルエリアネットワーク上のシステムで、電話回線に PPP または UUCP リンクを設定するためのモデムがあるものです。またネットワークからインターネットのグローバルネットワークへのルーターとして構成されたシステムも適しています (PPP、UUCP、およびルーターの詳細は、第 21 章、第 25 章、114ページの「ルーターの構成」を参照)。ローカルネットワーク上のシステムにモデムがない場合は、システムの 1 つをメールホストに指定してください。

サイトの中には、タイムシェアリング構成でネットワークに接続されていないスタンドアロンのマシンを使用するものがあります。つまり、スタンドアロンのマシンが、シリアルポートに接続された端末として機能する場合です。このような構成では、スタンドアロンのシステムを 1 つのシステムネットワークのメールホストとして扱うことで、電子メールを設定できます。

メールサーバー

「メールボックス」は単独のファイルで、特定ユーザー用の電子メールが含まれています。メールはユーザーのメールボックスが置かれている場所のシステム、つまりローカルマシンかリモートサーバーに配信されます。「メールサーバー」は、/var/mail ディレクトリにユーザーのメールボックスを保持しているいずれかのシステムになります。

メールサーバーはクライアントからすべてのメールをルーティングします。クライアントがメールを送信するときに、メールサーバーは配信のためそのメールを待ち行列に入れます。メールが待ち行列に入れられたら、ユーザーはこれらのメールメッセージを失わずに、クライアントをリブートしたり、電源を切ったりすることができます。受信者がクライアントからメールを受けとると、メッセージの「From」行のパスには、メールサーバー名が含まれます。受信者が応答すると、その応答はユーザーのメールボックスに送られます。メールサーバーとして適しているのは、ユーザーにホームディレクトリを提供するシステムか、定期的にバックアップされるシステムです。

メールサーバーがユーザーのローカルシステムでない場合は、構成内で NFS ソフトウェアを使用するユーザーは、/etc/vfstab ファイル (ルートアクセスがある場合) を使用するか、オートマウンタを使用して、/var/mail ディレクトリをマウントできます。NFS サポートが利用できない場合、ユーザーはサーバーにログインしてメールを読み込めます。

ネットワーク上のユーザーが、PostScript ファイル、オーディオファイル、DTP システムからのファイルなど他の形式のファイルを送信する場合は、メールボックスのメールサーバーには、さらに多くの領域を割り当てる必要があります。

全メールボックス用に 1 台のメールサーバーを設定する利点の一つは、バックアップが簡単になることです。数多くのシステムにメールを分散すると、バックアップが難しくなります。1 つのサーバーに多くのメールボックスを格納する際の欠点は、そのサーバーの故障が多くのユーザーに影響することですが、バックアップの簡便さは、この危険性を補って余りあります。

メールクライアント

「メールクライアント」は、メールサーバーでメールを受信し、ローカルの /var/mail のないシステムです。これはリモートモードとして知られています。リモートモードは、デフォルトでは /etc/mail/subsidiary.cf で使用することができます。

メールクライアントには、/etc/vfstab ファイルに適切なエントリがあり、メールサーバーからメールボックスをマウントするマウント先があることを確認する必要があります。またクライアントの別名の宛先が、クライアントではなく、メールサーバーのホスト名になっていることを確認してください。

メールゲートウェイ

「メールゲートウェイ」は、異なる通信プロトコルを実行するネットワーク間の接続を処理したり、同じプロトコルを使用する異なるネットワーク間の通信を処理したりするマシンです。たとえば、メールゲートウェイでは、Systems Network Architecture (SNA) プロトコルセットを実行するネットワークに、TCP/IP ネットワークを接続する場合もあります。

設定の最も簡単なメールゲートウェイは、同じプロトコルかメールプログラムを使用する 2 つのネットワークを接続するものです。このシステムでは、sendmail がドメインで受信者を見つけられないアドレスのあるメールを処理します。メール

ゲートウェイがある場合、sendmail はこれを使用して、ドメイン外でメールの送受信を行います。

2 つのネットワーク間には、図 35–1 に示すように内容の異なるメールプログラムを使用してメールゲートウェイを設定できます。これをサポートするには、メールゲートウェイシステムで sendmail.cf ファイルをカスタマイズする必要がありますが、これは困難で時間のかかる作業になる場合もあります。

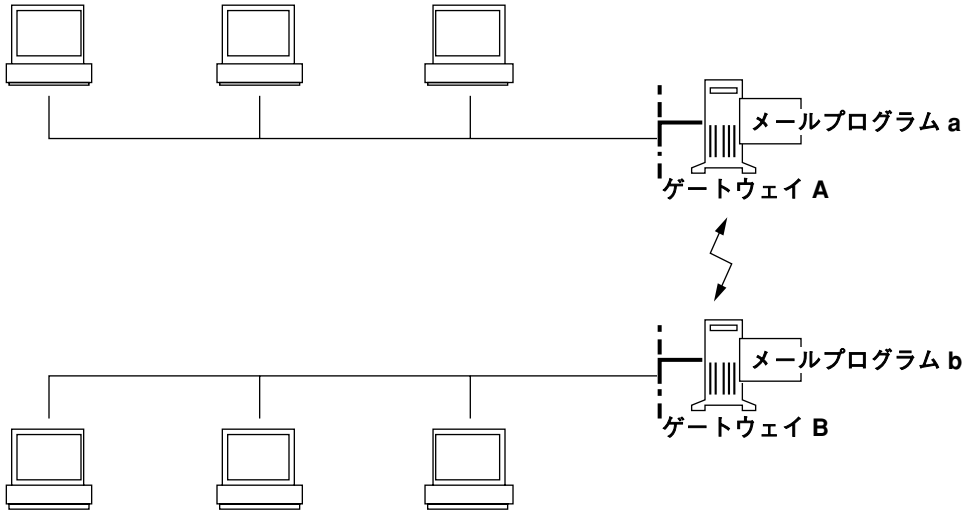


図 35–1　異なる通信プロトコル間のゲートウェイ

メールゲートウェイを設定する場合に、必要とするものに最も近いゲートウェイ構成ファイルを見つけ、状況に合わせて修正する必要があります。

インターネットに接続できるマシンがある場合は、そのマシンをメールゲートウェイとして構成できます。メールゲートウェイを構成するときは、まずサイトのセキュリティ要件を慎重に考慮する必要があります。社内ネットワークを外部と接続するには、ファイアウォールゲートウェイを構築し、メールゲートウェイとして設定する必要がある場合があります。

メールサービスのプログラムとファイル

メールサービスには、相互に対応する数多くのプログラムやデーモンが含まれています。この節では、電子メールの管理に関するプログラムや用語、あるいは概念

について述べます。表 35–5 には、メールサービスに使用する /usr/bin ディレクトリの内容を示します。

表 35–5 メールサービスに使用する /usr/bin ディレクトリの内容

名前	形式	説明
aliasadm	ファイル	NIS+ 別名マップを処理するプログラム
mail	ファイル	ユーザーエージェント
mailcompat	ファイル	メールを SunOS 4.1 メールボックスフォーマットに格納するフィルタ
mailq	リンク	/usr/lib/sendmail へのリンクで、メール待ち行列の表示に使用
mailstats	ファイル	/etc/mail/sendmail.st ファイルに格納されたメール統計情報の読み込みに使用するプログラム (存在する場合のみ)
mailx	ファイル	ユーザーエージェント
mconnect	ファイル	アドレスの検証とデバッグのためメールプログラムに接続するプログラム
newaliases	リンク	/usr/lib/sendmail へのリンクで、別名ファイルのバイナリ形式を作成するのに使用
praliases	ファイル	エイリアスデータベースを表示するコマンド
rmail	リンク	/usr/bin/mail へのリンクで、メールの送信だけを許可するのによく使用されるコマンド
vacation	ファイル	メールへの自動応答を設定するコマンド

表 35–6 に、/etc/mail ディレクトリの内容を示します。

表 35–6 /etc/mail ディレクトリの内容

名前	形式	説明
Mail.rc	ファイル	mailtool ユーザーエージェントのデフォルトの設定値
aliases	ファイル	メール転送情報
aliases.dir	ファイル	メール転送情報のバイナリ形式 (newaliases の実行によって作成される)
aliases.pag	ファイル	メール転送情報のバイナリ形式 (newaliases の実行によって作成される)
mailx.rc	ファイル	mailx ユーザーエージェントのデフォルトの設定値
main.cf	ファイル	メインシステム用の構成ファイルの例
relay-domains	ファイル	リレーが可能なドメインの全リストが含まれている。デフォルトでは、ローカルドメインだけが使用できる
sendmail.cf	ファイル	メールルーティング用の構成ファイル
sendmail.cw	ファイル	メールホスト用の別名の数が多すぎるときに作成可能なオプションファイル
sendmail.hf	ファイル	SMTP HELP コマンドで使用するヘルプファイル
sendmail.pid	ファイル	リスニングデーモンの PID を表示するファイル
sendmail.st	ファイル	sendmail 統計情報ファイル。このファイルが存在すると、sendmail は各メールプログラムのトラフィック量をログする
sendmailvars	ファイル	sendmail.cf からの名前空間の検索用のマクロとクラス定義を格納する
subsidiary.cf	ファイル	下位システムに対する構成ファイルの例

表 35–7 にメールサービスに使用する /usr/lib ディレクトリの内容を示します。

表 35–7 メールサービスに使用する /usr/lib ディレクトリの内容

名前	形式	説明
mail.local	ファイル	メールボックスにメールを配信するメールプログラム
sendmail	ファイル	メール転送エージェントとしても知られるルーティングプログラム
smrsh	ファイル	sendmail が実行できるプログラムを、/var/adm/sm.bin 内にあるプログラムに限定するシェルプログラム

/usr/lib ディレクトリ内は、sendmail.cf ファイルの構築に必要なファイルをすべて含むサブディレクトリです。このディレクトリの内容は、表 35–8 に示すとおりです。

表 35–8 メールサービスに利用する /usr/lib/mailディレクトリの内容

名前	形式	説明
README	ファイル	構成ファイルを説明する文書
cf	ディレクトリ	ホストのサイトに依存する、およびサイトに依存しない説明
cf/main-v7sun.mc	ファイル	主要な構成ファイル
cf/makefile	ファイル	新しい構成ファイルを作成する場合の規則が含まれている
cf/subsidiary-v7sun.mc	ファイル	/var/mail を別のホストから NFS マウントするホストの構成ファイル
domain	ディレクトリ	サイトに依存するサブドメインの説明
domain/generic.m4	ファイル	Berkeley からのジェネリックドメインファイル

表 35–8　メールサービスに利用する /usr/lib/mailディレクトリの内容　続く

名前	形式	説明
domain/solaris-antispam.m4	ファイル	sendmail 関数を以前の Solaris 版のようにする変更を伴うドメインファイル。リレーがまったく使用できない場合を除いて、ホスト名が指定されていない送信側アドレスは拒否され、また解決されないドメインは拒否される
domain/solaris-generic.m4	ファイル	sendmail 関数を以前の Solaris 版のようにする変更を伴うドメインファイル (デフォルト)
feature	ディレクトリ	特定のホスト用の特別な機能の定義 (機能の詳細な説明は README を参照)
m4	ディレクトリ	サイトに依存しないインクルードファイル
mailer	ディレクトリ	ローカル、smtp、および uucp を含むメールプログラムの定義
ostype	ディレクトリ	いろいろなオペレーティングシステム環境を説明する定義
ostype/solaris2.m4	ファイル	ローカルメールプログラムを mail に定義する
ostype/solaris2.ml.m4	ファイル	ローカルメールプログラムを mail.local に定義する (デフォルト)
sh	ディレクトリ	m4 作成プロセスと移行支援プログラムで使用するシェルスクリプト
sh/check-permissions	ファイル	include: エイリアスと .forward ファイルのアクセス権、および正確なアクセス権に必要なこれらの親ディレクトリのパスを確認する
sh/check-hostname	ファイル	sendmail が完全指定のホスト名を判別できることを確認する

表 35–8　メールサービスに利用する /usr/lib/mailディレクトリの内容　続く

メールサービスは、その他のいくつかのファイルおよびディレクトリを使用します。これらを表 35–9 に示します。

表 35–9　メールサービスに使用するその他のファイル

名前	形式	説明
sendmailvars.org_dir	テーブル	sendmailvars ファイルの NIS+ バージョン
/etc/default/sendmail	ファイル	sendmail 用の環境変数を示す
/etc/shells	ファイル	有効なログインシェルをリストする
/usr/sbin/in.comsat	ファイル	メール通知デーモン
/usr/sbin/makemap	ファイル	入力されたマップのバイナリフォーマットを構築する
/usr/sbin/syslogd	ファイル	sendmail が使用するエラーメッセージログをとるデーモン
/usr/dt/bin/dtmail	ファイル	CDE メールユーザーエージェント
/var/mail/mailbox1、/var/mail/mailbox2	ファイル	配信されたメールのメールボックス
/var/spool/mqueue	ディレクトリ	配信されないメール用の記憶領域
$OPENWINHOME/bin/mailtool	ファイル	ウィンドウベースのメールユーザーエージェント

これらのプログラムの組み合わせによるメールサービスが提供されていますが、その相互作用を図 35–2 に簡略に示します。

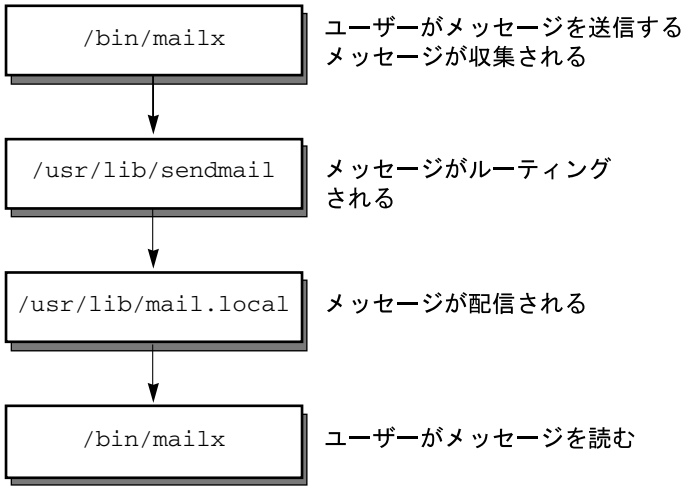


図 35–2　メールプログラムの相互作用

ユーザーは、mailx や mailtool などのプログラムを使用してメッセージを送信します。これらのプログラムについては、mailx(1) または mailtool(1) のマニュアルページを参照してください。

メッセージは、メッセージを生成するのに使用されたプログラムにより収集され、sendmail デーモンに渡されます。sendmail デーモンは、メッセージのアドレスを「解釈」し (識別可能なセグメントに分割)、構成ファイル /etc/mail/sendmail.cf からの情報を使用して、ネットワークの名前構文、別名、転送情報、およびネットワークトポロジを決定します。sendmail はこの情報を使用して、メッセージが受信者に到達する経路を決定します。

sendmail デーモンはメッセージを適切なシステムに渡します。ローカルシステムの /usr/lib/mail.local プログラムは、メッセージの受信者の /var/mail/*username* ディレクトリのメールボックスにメールを配信します。

受信者は、メールが届いたことが通知されるので、mail、mailx、mailtool などのプログラムを使用してこれを受け取ります。

sendmail プログラム

sendmail プログラムは、TCP/IP や UUCP などの異なる通信プロトコルを使用できます。また SMTP サーバー、メッセージキュー、メーリングリストも実装します。名前の解釈は、ドメインベースのネーミングとその環境で指定されている規則の両方を処理できるパターンマッチングシステムで制御されます。

sendmail プログラムは、ドメインベースのネーミングと任意の (古い) 名前構文を受け入れて、指定されている補完方法を使用して曖昧さを解決します。sendmail は共通点のないネーミングスキーマ間でメッセージを変換することもできます。ドメインの手法は、物理的なネーミング対論理的なネーミングの問題を分離します。インターネットドメインのネーミングの規則の詳細は、91ページの「ドメイン名」を参照してください。

他のネットワーク上のホストに対してローカルのように見えるネットワーク名を提供するなど、その環境で指定されている技法によって特殊な場合を処理できます。

Solaris オペレーティング環境では、sendmail プログラムをメールルーターとして使用します。sendmail は、電子メールメッセージの受信と配信を担当します。これは、mail、mailx、mailtool といったメール読み取りプログラムと、uucp のようなメールトランスポートプログラムの間のインタフェースです。sendmail プログラムは、ユーザーが送った電子メールメッセージを制御し、受信者のアドレスを判断し、適切な配信プログラムを選び、配信エージェントが処理できるフォーマットにアドレスを書き直し、必要に応じてメールヘッダーをフォーマットし直し、最後に変換したメッセージを配信のためのメールプログラムに渡します。

注 - Solaris 2.4 以前の旧リリース版には、sendmail.mx と呼ばれるバイナリが含まれていました。現在このプログラムは sendmail プログラムに含まれており、これを有効にするには、/etc/nsswitch.conf のホストエントリに dns フラグを追加します。詳細は、748ページの「sendmail で DNS を使用する方法」 を参照してください。

---

sendmail プログラムでは、メールルーティングに必要な 3 つのメカニズムをサポートしています。どのメカニズムを選択するかは、サーバーまたはドメイン全体の変更なのか、または単に 1 人のユーザーの変更であるかによって決まります。また、異なる再ルーティングメカニズムを選択することにより、必要な管理レベルに変更できます。

1 つ目の再ルーティングメカニズムはエイリアシングです。エイリアシングとは、使用するファイルのタイプに基づいて、サーバー全体、または名前空間全域ごとに名前をアドレスに対応させるメカニズムです。名前空間の別名ファイルを使用すると、メール再ルーティングの変更を単一のソースで管理できますが、この変更が伝達されるときに、遅延時間が発生する可能性があります。また、名前空間管理は、通常、システム管理者の選択グループに限定されるため、一般ユーザーが実行できる変更ではありません。サーバーの別名ファイルを通じて処理された再ルーティングは、そのサーバーのスーパーユーザーによって管理されます。通常、この変更の伝達に関連した遅延時間はほとんどみられませんが、この変更はローカルサーバー

にしか反映されません。この制約事項は、メールのほとんどが 1 つのサーバーに送信される場合には問題ありませんが、この変更を多数のメールサーバーに配信する場合には、ネームサービスを使用した方が簡単です。これも一般ユーザーが実行できる変更ではありません。

次のメカニズムは、転送と取り込みです。このメカニズムを使用すると、ユーザーはメールの再ルーティングを実行できます。転送を使用すると、ローカルユーザーは、着信メールを他のメールボックス、別のメールプログラム、あるいは他のメールホストにルーティングし直すことができます。このメール再ルーティングの形式は、`.forward` ファイルを使用することによりサポートされます。これらのファイルの詳細は、800ページの「`.forward` ファイル」を参照してください。

最後の再ルーティングメカニズムは取り込みで、これを使用すると、別名リストを、ルートアクセスを要求する代わりに、ユーザーによって保守できます。このメカニズムを提供するには、スーパーユーザーは、サーバー上の別名ファイル内に適切なエントリを作成する必要があります。このエントリが作成されると、ユーザーは必要に応じてメールをルーティングし直すことができるようになります。取り込みの詳細は、796ページの「`/etc/mail/aliases`」を参照してください。

図 35–3 は、`sendmail` がユーザー別名をどのように使用するかを示します。`/usr/bin/mailx` のようなメールを読み取るプログラムは、プログラム自身の別名を持つことができ、それらはメッセージが `sendmail` に達する前に展開されます。`sendmail` の別名は、多くの名前空間ソース (ローカルファイル、NIS、NIS+) からのものでも構いません。検索順序は `nsswitch.conf` ファイルによって決定されます。`nsswitch.conf(4)` のマニュアルページを参照してください。

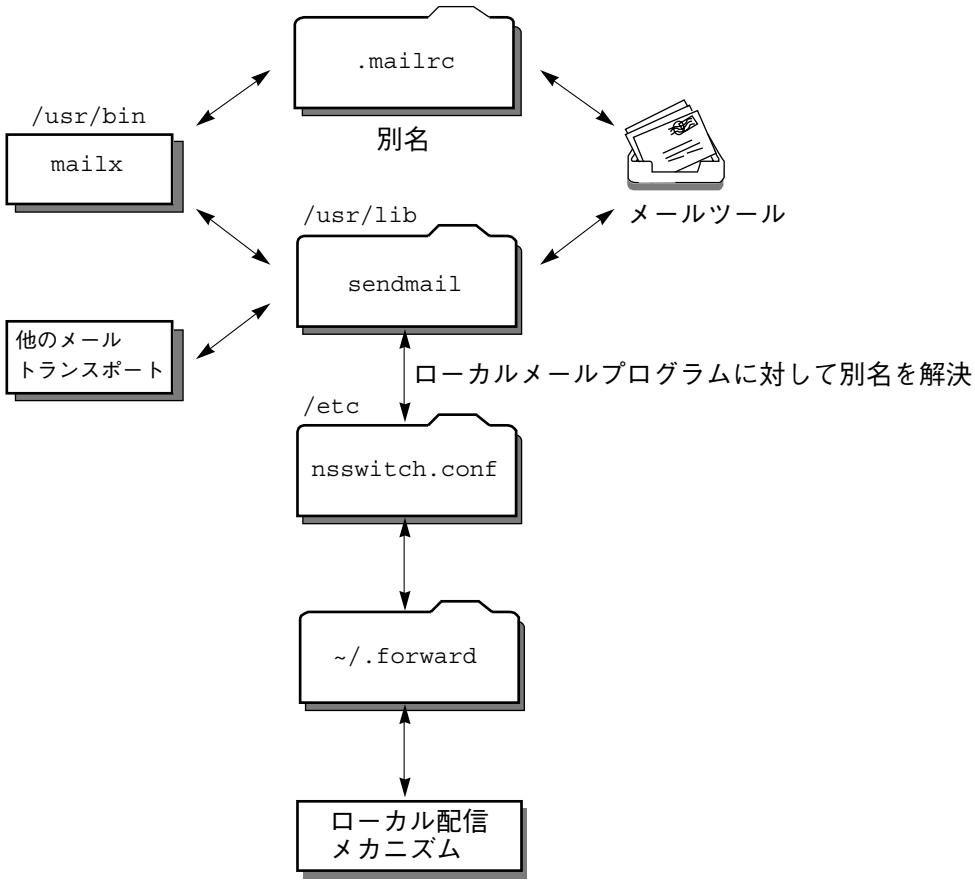


図 35–3　sendmail が別名を使用する方法

sendmail プログラムの機能

sendmail プログラムには、次のような機能があります。

- sendmail には高い信頼性があります。すべてのメッセージを正しく配信するように設計されています。どんなメッセージも完全に失われることはありません。
- sendmail は、既存のソフトウェアを配信に随時使用します。
- sendmail は、1 つのネットワークタイプ (UUCP や Ethernet など) に複数の接続を行う場合なども含め、複雑な環境を処理するように構成できます。sendmail は、名前とその構文を確認し、どのメールプログラムを使用するかを判断します。

- 構成情報をコードにコンパイルする代わりに、構成ファイルを使用してメール構成を制御します。
- ユーザーは独自のメーリングリストを管理できます。各ユーザーは、ドメイン全体で有効な別名ファイル (通常、NIS または NIS+ によって管理されるドメイン全体の別名の中にある) を修正することなく自分自身のメール転送を指定できます。
- 各ユーザーはカスタムメールプログラムを指定して着信メールを処理することができます。こうすると、たとえば、「I am on vacation」というメッセージを返すといった機能を設定できます。`vacation(1)` のマニュアルページを参照してください。
- 1 つのホストでアドレスを処理し、ネットワークトラフィックを削減します。

図 35–4 には、`sendmail` がメールシステムで他のプログラムと対話する方法を示します。

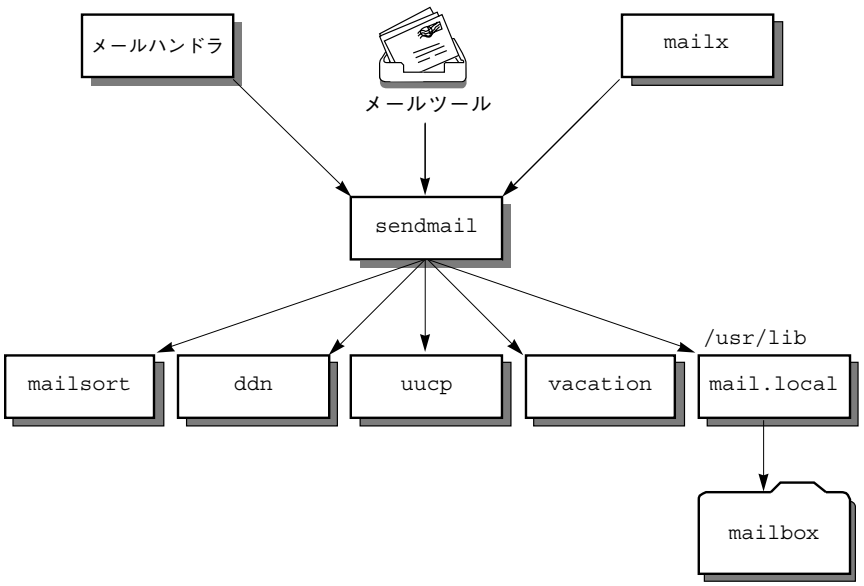


図 35–4　`sendmail` と他のメールプログラムとの相互作用

ユーザーは、メール生成プログラムおよび送信プログラムと対話します。メール送信が依頼されると、メール生成プログラムは `sendmail` を呼び出し、`sendmail` は適切なメールプログラムにメッセージを送ります。発信者の一部はネットワークサーバーであったり、またメールプログラムの一部はネットワーククライアントであるため、`sendmail` は、インターネットメールゲートウェイとしても使用できます。

sendmail 構成ファイル

「構成ファイル」は、sendmail がその機能を実行する方法を制御します。構成ファイルにより、配信エージェント、アドレスの変換の規則、およびメールヘッダーのフォーマットが選択されます。

sendmail プログラムは、/etc/mail/sendmail.cf ファイルの情報を使用して、その機能を実行します。各システムには、/etc/mail ディレクトリにインストールされたデフォルトの sendmail.cf ファイルがあります。メールサーバーまたはメールクライアントのためにデフォルト構成ファイルを編集または変更する必要はありません。カスタマイズされた構成ファイルを必要とするシステムは、メールホストとメールゲートウェイだけです。

Solaris オペレーティング環境には、以下に示すように、/etc/mail ディレクトリに 2 つのデフォルト構成ファイルがあります。

1. メールホストまたはメールゲートウェイとして使用する 1 つのシステム (または複数のシステム) を指定するための main.cf という名前の構成ファイル
2. subsidiary.cf という名前の構成ファイル (デフォルト sendmail.cf ファイルの複製コピー)

システムで使用する構成ファイルは、システムがメールサービスで果たす役割によって異なります。

- メールクライアントまたはメールサーバーについては、デフォルト構成ファイルを設定または編集する必要はありません。
- メールホストやゲートウェイを設定するには、main.cf ファイルをコピーし、それを (/etc/mail ディレクトリで) sendmail.cf と名称変更します。次に、sendmail.cf ファイルを再構成して、リレーメールプログラムを設定して、メール設定に必要なホストパラメータをリレーします。

次に、サイトの要求に応じて変更が可能な構成パラメータをいくつか説明します。

- 時間値の指定
  - 読み取りのタイムアウトを指定します。
  - メッセージが待ち行列内にあり送信者に戻されるまでの未配信状態の時間を指定します。
- 配信 (delivery) モードは、メールが配信される速さを指定します。

- 長いメッセージ、多くの受信者へのメッセージ、および長時間ダウンしているサイトへのメッセージを配信しないことにより、ロードを制限してロード時間内の無駄を省きます。
- ログレベルは、どのような種類の問題をログするかを指定します。

メール別名ファイル

下記の任意のファイルを使用して、別名を管理できます。使用するファイルのタイプは、別名を使用する人と別名を変更する必要がある人によって決まります。別名ファイルのタイプにはそれぞれ固有の形式要件があります。これについては、以下で定義します。

.mailrc の別名

.mailrc ファイルのリストに入っている別名には、ファイルを所有するユーザーだけしかアクセスできません。これにより、ユーザーは自分で制御し、所有者だけが使用できる別名を作成できます。.mailrc ファイルの別名は、次のようになります。

```
alias aliasname value value value ...
```

aliasname は、ユーザーがメールの送信時に使用する名前であり、*value* は有効な電子メールアドレスです。

ユーザーが scott に個人的な別名を作成し、それが名前空間の scott の電子メールアドレスと一致しない場合、そのユーザーが作成したメールに他のユーザーが返信しようとするときに、メールが間違ったユーザーに転送されることになります。これを回避するには、別の別名命名方式を使用する以外にありません。

/etc/mail/aliases

/etc/mail/aliases ファイルで作成したいずれの別名も、その別名の名前とファイルを含んでいるシステムのホスト名を知っているユーザーなら誰でも使用できます。ローカルの /etc/mail/aliases ファイルの配布リストは、以下のようになります。

```
aliasname: value,value,value...
```

aliasname は、ユーザーがこの別名にメールを送信するときに使用する名前で、*value* は有効な電子メールアドレスになります。

ご使用のネットワークがネームサービスを実行していない場合は、各システムの /etc/mail/aliases ファイルにすべてのメールクライアントのエントリを入れておく必要があります。各システムのファイルを編集するか、1 つのシステムのファイルを編集してからそのファイルを他のシステムに個々にコピーします。

/etc/mail/aliases ファイルの別名は、テキスト形式で保存されます。/etc/mail/aliases ファイルを編集するときに、newaliases プログラムを実行してデータベースを再コンパイルし、sendmail プログラムでその別名がバイナリ形式で使用できるようにします。あるいは Administration Tool の Database Manager を使用して、ローカルの /etc ファイルに保存されているメール別名を管理できます。

エイリアスを作ることができるのは、ローカル名、つまり現在のホスト名に対してのみ、またはホスト名は指定できません。たとえば、システム saturn 上にメールボックスを持っているユーザー ignatz に対するエイリアスエントリは、下記エントリを /etc/mail/aliases ファイル内に持っています。

```
ignatz: ignatz@saturn
```

各メールサーバー上で管理用アカウントを作ると便利です。このアカウントを作成する場合は、メールサーバー上にメールボックスのルートを割り当て、ルートについての /etc/mail/aliases ファイルにエントリを追加します。たとえば、システム saturn がメールボックスサーバーの場合は、エントリ root: sysadmin@saturn を /etc/mail/aliases ファイルに追加します。

通常、このファイルを編集できるのはスーパーユーザーだけです。admintool を使用する場合は、sysadmin グループであるグループ 14 のすべてのユーザーが、ローカルファイルを変更できます。別のオプションとしては、以下のようなエントリが作成できます。

aliasname: :include:/*path*/*aliasfile*

aliasname は、ユーザーがメールを送信するときに使用する名前であり、/path/aliasfile は別名リストを含むファイルへの完全なパスになります。別名ファイルには、各行に 1 つの電子メールエントリを入れ、その他の表記は付けないでください。

```
user1@host1
user2@host2
```

/etc/mail/aliases に追加のメールファイルを定義して、ログやバックアップコピーの管理もできます。以下のエントリでは、*filename* の *aliasname* に送信されるすべてのメールを格納します。

```
aliasname: /home/backup/filename
```

また、メールを他のプロセスにルーティングすることもできます。次のように入力すると、メールメッセージのコピーが *filename* 内に格納され、コピーが出力されます。

```
aliasname: "|tee -a /home/backup/filename |lp"
```

NIS 別名マップ

NIS 別名マップに含まれているエントリは、ローカルドメインのすべてのユーザーが利用できます。sendmail プログラムは、ローカルの /etc/mail/aliases ファイルの代わりに NIS 別名マップを使用して、メールアドレスを決定できます。詳細は、nsswitch.conf(4) のマニュアルページを参照してください。

NIS 別名 マップの別名は、以下のようになります。

```
aliasname: value,value,value...
```

aliasname は、ユーザーがメールを送信するときに使用する名前であり、*value* は有効な電子メールアドレスです。

NIS 別名マップには、すべてのメールクライアント用のエントリを含めてください。一般にこれらのエントリを変更できるのは、NIS マスターのスーパーユーザーだけです。このタイプの別名は、頻繁に変更される別名としては適していないかもしれませんが、次の構文例のように、別名が他の別名ファイルを指している場合は便利です。

```
aliasname: aliasname@host
```

aliasname はユーザーがメールを送信するときに使用する名前であり、*host* は /etc/mail/alias ファイルを含むサーバー用のホスト名です。

NIS+ mail_aliases テーブル

NIS+ mail_aliases テーブルには名前が含まれていて、それによってローカルドメインにおけるシステムや個人が登録されています。sendmail プログラムは、ローカルの /etc/mail/aliases ファイルの代わりに NIS+ mail_aliases テーブルを使用して、メールアドレスを決定できます。詳細は、aliasadm(1M) と nsswitch.conf(4) のマニュアルページを参照してください。

NIS+ mail_aliases テーブルの別名は次のようになります。

```
alias:    expansion      [options   # "comments"]
```

表 35–10 に 4 つの列を記載します。

表 35–10 NIS+ mail_aliases テーブルの列

列	説明
alias	別名の名前
expansion	sendmail /etc/mail/aliases ファイルに現れる別名の値または別名のリスト
options	将来の使用のために確保
comments	個々の別名に関するコメント

NIS+ mail_aliases テーブルには、すべてのメールクライアントのエントリを含めてください。NIS+ aliases テーブルでは、aliasadm コマンドで、エントリの表示、作成、変更、および削除ができます。あるいは admintool の Database Manager を使用して、NIS+ メール別名を管理できます。

新規の NIS+ 別名テーブルを作成する場合は、エントリを作成する前にテーブルを初期設定する必要があります。テーブルが存在するときは、初期設定は不要です。

aliasadm コマンドを使用するには、別名テーブルを所有する NIS+ グループのメンバーか、テーブルを作成したユーザーでなければなりません。

.forward ファイル

ユーザーは、システム管理者の手を借りることなくプログラムのカスタムセットにメールをリダイレクトまたは送信するために、ホームディレクトリに、sendmail が使用する .forward ファイルを作成できます。メールの問題、特に所定のアドレスに配信されないメールに関する問題の解決の際、ユーザーのホームディレクトリに .forward ファイルがあるかどうかを常に確認してください。

よくある間違いは、host1 上のホームディレクトリの .forward ファイルに、user@host2 にメールを転送する設定を入れてしまうことです。メールが host2 に送られると、sendmail は NIS や NIS+ 別名で user を検索し、user@host1 にメッセージを送り返すので、ループが発生し、メールは返送されてしまいます。

注 - root および bin アカウントは、.forward ファイルを所有できません。.forward ファイルを作成すると、セキュリティ上の問題が生じます。必要な場合には、代わりに別名ファイルを使用してメールを転送してください。

メールの配信中に .forward ファイルを調べるためには、このファイルを、ファイルの所有者によってのみ書き込み可能な状態にしておく必要があります。これにより、他のユーザーによるファイルへのアクセスを防ぎます。また、ホームディレクトリのパスは、root だけが所有し、書き込める状態にしておく必要があります。特に、.forward ファイルが /export/home/terry 内にある場合には、/export と /export/home は root だけが所有し、書き込める状態にしておかなければなりません。また実際のホームディレクトリに書き込めるのは、そのユーザーだけである必要があります。.forward ファイルにはこの他にも制約があります。このファイルはシンボリックリンクにすることはできず、また複数のハードリンクも実行できません。

標準の .forward ファイルに加えて、.forward.*hostname* ファイルを作成し、特定のホストに送信されたメールを転送できます。たとえば、ユーザーの別名を sandy@phoenix.eng.acme.com から sandy@eng.acme.com に変更した場合、sandy のホームディレクトリ内に .forward.phoenix ファイルがあると便利です。

```
% cat .forward.phoenix
sandy@eng.acme.com
"|/usr/bin/vacation sandy"
% cat .vacation.msg
From: sandy@eng.acme.com (via the vacation program)
```

(続く)

```
Subject: my alias has changed

My alias has changed to sandy@eng.acme.com.
Please use this alias in the future.
The mail that I just received from you
has been forwarded to my new address.

Sandy
```

こうすることにより、メールを適切な場所に転送すると同時に、別名の変更を送信者に通知できます。vacation プログラムではメッセージファイルは 1 つしか使用できないため、この場合 1 回につき 1 つのメッセージしか実行できません。ただし、メッセージがホスト固有のものではない場合には、1 つの vacation メッセージファイルを、複数のホストの .forward ファイルで使用できます。

転送メカニズムの拡張機能にはこの他に、.forward+*detail* ファイルがあります。*detail* は、オペレータ文字以外の文字を自由に並べることができます。オペレータ文字とは、.:%&!^[]+ です。このようなファイルを使用すると、第三者によって自分の電子メールアドレスが使用されたかどうかを判別することが可能になります。たとえば、あるユーザーが、誰かに電子メールアドレス sandy+test1@eng.acme.com を使用するように指示した場合、ユーザーは、この別名に配信されるメールを、アドレスに送信されるメールの中から識別できます。デフォルトにより、sandy+test1@eng.acme.com の別名に送信されたメールはすべて、この別名と .forward+*detail* ファイルと突き合わせて検査されます。ここで一致しない場合は、そのメールは最終的に sandy@eng.acme.com に配信されますが、ユーザーは、これらのメールの To: ヘッダー内の変更箇所を調べることができます。

/etc/default/sendmail

このファイルは sendmail のための初期設定用オプションを保存し、ホストをアップグレードしたときに除去されないようにするために使用します。次の変数を使用することができます。

MODE=-bd

sendmail を起動するためのモードを選択します。–bd オプションを使用するか、未定義のままにしておきます。

QUEUEINTERVAL=#

実行するメールキューのための間隔を設定します。# は正の整数とし、その後に秒の場合は s、分の場合は m、時の場合は h、日の場合は d、週の場合は w を付けます。この構文は sendmail の起動前に確認されます。この間隔が負の場合、またはエントリの最後の文字が不適当な場合、この間隔は無視され、sendmail は 15 分のキュー間隔で起動します。

OPTIONS=*string*

sendmail コマンドで使用する追加のオプションを選択します。構文の確認は行われないため、この変数を変更するときは間違えないように注意してください。

メールアドレス指定の動作

配信時にメールメッセージが辿る経路は、クライアントシステムの設定とメールドメインのトポロジによって異なります。メールホストやメールドメインの各追加レベルでは、別名の解釈をさらに 1 回追加できますが、ルーティングプロセスは基本的にほとんどのホストで同じになります。

クライアントシステムを設定してメールをローカルで受信したり、リモートでクライアントシステムのためのメールを受信したりできます。メールをローカルで受信することは、ローカルモードでの sendmail の実行として知られています。すべてのメールサーバーと一部のクライアントでは、ローカルがデフォルトモードです。クライアントがサーバーから /var/mail をマウントしている場合、クライアントはリモートモードで sendmail を実行します。

sendmail.cf ファイルで設定したデフォルト規則を使用している場合の、電子メールメッセージが辿る経路を以下に示します。

リモートモードのメールクライアントでは、メールメッセージは以下のルーティングプロセスを経由して送信されます。

1. 可能な場合メール別名を展開し、ローカルのルーティングプロセスを再起動します。

   /etc/nsswitch.conf のエントリに応じて、名前空間でメール別名を検索し、新しい値が見つかった場合に置換することで、メールアドレスが展開されます。この新しい別名が次に再度確認されます。

2. アドレスを展開できない場合、メールサーバーに転送します。

   メールアドレスを展開できない場合、アドレスに問題があるか、アドレスがローカルでない可能性があります。どちらの場合も、メールサーバーで問題を解決する必要があります。

3. 展開した別名が元の宛先にループバックすると、メールはメールサーバーに転送されます。

   このプロセスでは検索のすべての履歴が維持され、元の別名が再生成されると、メールはメールサーバーに転送されて解釈が行われます。

ローカルモードのメールサーバーやメールクライアント上で、メールメッセージは以下のルーティングプロセスを経由して送信されます。

1. 可能な場合はメール別名を展開し、ローカルのルーティングプロセスを再起動します。

   名前空間でメール別名を検索し、見つかった場合に新しい値と置換することで、メールアドレスが展開されます。次にこの新しい別名が再度確認されます。

2. メールがローカルの場合、`/usr/lib/mail.local` に配信されます。

   メールはローカルのメールボックスに配信されます。

3. メールアドレスがこのメールドメインにホストを含んでいると、そのホストにメールを配信します。

4. アドレスがこのドメインにホストを含んでいない場合、メールホストにメールを転送します。

   メールホストはメールサーバーと同じルーティングプロセスを使用しますが、メールホストはホスト名に加え、ドメイン名が宛先になっているメールも受信できます。

sendmail とネームサービスとの相互作用

「メールドメイン」は、標準の sendmail.cf ファイルによって使用される概念で、メールを直接配信するか、またはメールホストによって配信するかを判断します。ドメイン内メールは直接 SMTP 接続経由で配信され、ドメイン間メールはメールホストに送られます。

セキュリティの高いネットワークでは、ほんの少数の選ばれたホストだけが、外部宛てのパケットを生成する権限を与えられています。ホストがメールドメイン外のリモートホストの IP アドレスを持っていても、これで SMTP 接続が確立できるとは限りません。標準の sendmail.cf では次のことを仮定しています。

- 現在のホストは、パケットを直接メールドメイン外のホストに送信する権限がない
- メールホストは、パケットを直接外部ホストに送信することが可能な認定ホストにメールを転送できる (実際には、メールホスト自身が認定ホストとなりうる)

このように仮定すると、ドメイン間メールの配信または転送はメールホスト側の責任です。

ネームサービスに対する sendmail 要件を設定する方法

sendmail は各種の要件をネームサービスに課します。次の節で、これらの要件とその要件を満たす方法を説明します。詳細は、in.named(1M)、nis+(1)、nisaddent(1M)、および nsswitch.conf(4) のマニュアルページを参照してください。

ネームサービスによるメールドメイン名の設定

メールドメイン名はネームサービスドメイン名の接尾辞の 1 つでなければなりません。たとえば、ネームサービスのドメイン名が「A.B.C.D」ならば、メールドメイン名は次のうちのいずれかです。

- A.B.C.D
- B.C.D
- C.D
- D

メールドメイン名は、最初に設定されたときには、多くの場合ネームサービスドメインと同じになります。ネットワークが大きくなれば、ネームサービスドメインを小さく分割してネームサービスを管理しやすくすることができます。ただし、メールドメインは、一貫した別名を提供するために分割されないまま残ることがあります。

ホスト名前空間データ

ネームサービスにおけるホストテーブルまたはマップは、次の 3 種類の gethostbyname() による問い合わせをサポートするように設定しなければなりません。

- mailhost – いくつかのネームサービスの構成では、自動的にこの要件を満たします。
- 完全なホスト名 (たとえば、smith.admin.acme.com) – 多くのネームサービスの構成がこの要件を満たします。
- 短いホスト名 (たとえば、smith) – sendmail はメールホストに接続し、外部へのメールを転送します。メールアドレスが現在のメールドメイン内であるかどうかを判定するために、gethostbyname() が完全なホスト名で呼び出されます。エントリが見つかると、アドレスは内部にあるとみなされます。

  NIS、NIS+、および DNS はすべて、短いホスト名を引数にする gethostbyname() をサポートします。したがって、この要件は自動的に満たされます。

名前空間内で sendmail サービスを適切に確立するには、さらにホスト名空間に関する以下の 2 つのルールに従う必要があります。

1. 完全なホスト名による gethostbyname() と短いホスト名による gethostbyname() で、一致した結果を生じるようにします。たとえば、両関数がメールドメイン admin.acme.com から呼び出される限り、gethostbyname (smith.admin.acme.com) と gethostbyname (smith) は同じ結果になるようにします。
2. 共通のメールドメイン下のすべてのネームサービスドメインに対しては、短いホスト名による gethostbyname() で同じ結果を生じるようにします。たとえば、メールドメイン smith.admin.acme.com があるとして、gethostbyname(smith) は、ebb.admin.acme.com または esg.admin.acme.com のいずれのドメインから呼び出されても同じ結果になるようにします。主なメールドメイン名は通常ネームサービスドメインより短く、このために各種ネームサービスにとって特別な意味のあるものになっています。

NIS と sendmail を使用する場合の設定の問題点

ネームサービスとして NIS だけを使用するときは、sendmail 使用時に事前に解決しておかなければならない設定項目を以下に示します。

メールドメイン名 – NIS をプライマリネームサービスとして設定している場合に、sendmail は、自動的に NIS ドメイン名の最初の構成要素を取り除いた結果をメールドメイン名として使用します。たとえば、ebs.admin.acme.com は、admin.acme.com となります。

mailhost ホスト名 – NIS のホストマップには、mailhost エントリが必要になります。

完全なホスト名 – 通常の NIS の設定では、完全なホスト名は認識されません。NIS に完全なホスト名を認識させようとするよりは、sendmail.cf ファイルを編集し %l を %y で置き換えて、sendmail 側からこの要件をなくしてください。こうすることによって、sendmail のドメイン間のメール検出機能をオフにできます。ターゲットとするホストの IP アドレスを取得できれば、SMTP による直接配信が試みられます。NIS のホストマップに現在のメールドメインの外部のホストのエントリが含まれていないことを確認してください。もし、そのエントリがあれば、さらに sendmail.cf ファイルをカスタマイズする必要があります。

ホストの完全名および短縮名のマッチング – 前述した手順を参考にして、完全なホスト名による gethostbyname() をオフにしてください。

1 つのメールドメイン内の複数の NIS ドメイン – 共通のメールドメインの NIS のホストマップは、ホストのエントリは同じである必要があります。たとえば、ebs.admin.acme.com ドメインのホストマップは、esg.admin.acme.com のホストマップと同じものにします。異なる場合には、ある NIS ドメインで有効なアドレスが他の NISドメインでは無効になることがあります。

sendmail と同時に NIS と DNS を使用する場合の設定の問題

ネームサービスとして NIS と DNS を同時に使用する場合に、sendmail を使用する前に解決しておかなければならない設定上の問題を以下に示します。

メールドメイン名 – NIS をプライマリネームサービスとして設定している場合には、sendmail は、自動的に NIS ドメイン名の最初の構成要素を取り除いた結果をメールドメイン名として使用します。たとえば、ebs.admin.acme.com は、admin.acme.com となります。

mailhost ホスト名 – DNS の転送機能がオンになっていれば、NIS で解決できない照会は DNS に転送されるため、NIS ホストマップに mailhost エントリは必要ありません。

完全なホスト名 – NIS が完全なホスト名を認識できなくても、DNS が認識します。NIS と DNS の通常の設定手順を踏んでいる場合には、完全なホスト名の要件は満たされます。

ホストの完全名および短縮名のマッチング – NIS のホストテーブルにおけるすべてのホストエントリに対して、DNS にも対応するホストエントリが必要です。

1 つのメールドメイン内の複数の NIS ドメイン – 共通のメールドメインの NIS のホストマップ中のホストのエントリは同じである必要があります。たとえば、ebs.admin.acme.com ドメインのホストマップは、esg.admin.acme.com のホストマップと同じものにします。異なる場合には、ある NIS ドメインで有効なアドレスが他の NIS ドメインでは無効になることがあります。

NIS+ と sendmail を使用する場合の設定の問題点

使用するネームサービスが NIS+ だけの場合、sendmail を使用する前に解決しておかなければならない設定上の問題点を以下に記します。

メールドメイン名 – プライマリネームサービスとして、NIS+ を設定していれば、sendmail は、NIS+ の sendmailvars テーブル (キーと値から構成される 2 列の NIS+ テーブル) からメールドメインを検索します。メールドメインを設定するには、このテーブルにエントリを 1 つ追加する必要があります。このエントリは、キーの列に文字列 maildomain が、値の列には自分のドメイン名 (たとえば、admin.acme.com) が設定されている必要があります。NIS+ では、sendmailvars テーブルにどのような文字列でも設定できますが、メールシステムが正常に機能するように接尾辞の規則が適用されます。nistbladm を使用して、maildomail エントリを sendmailvars テーブルに追加できます。たとえば、次のようになります。

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

メールドメインは NIS+ ドメインの接尾辞となることに注意してください。

mailhost ホスト名 – NIS+ ホスト名には、mailhost エントリが必要です。

完全なホスト名 – NIS+ は、完全なホスト名を認識することができます。通常の NIS+ の設定手順を行えば、この完全なホスト名の要件は満たされます。

ホストの完全名および短縮名のマッチング – この要件を満たすには、すべてのホストテーブルでエントリをコピーするか、ユーザーネームサービスのドメイン中の全ホストのエントリをメールドメインレベルのマスターホストテーブルに入力する必要があります。

1 つのメールドメイン内の複数の NIS ドメイン – この項目を満たすには、すべてのホストテーブルのエントリをコピーするか、ユーザーネームサービスのドメイン中の全ホストのエントリをメールドメインレベルのマスターホストテーブルに入力する必要があります。これは、(論理的または物理的に) 複数のホストテーブルを 1 つのホストテーブルに結合することになるので、メールドメインを共有する複数のネームサービスドメインで同じホスト名を再使用することはできません。

sendmail と同時に NIS+ と DNS を使用する場合の設定の問題点

ネームサービスとして NIS+ と DNS を同時に使用する場合に、sendmail 使用前に解決しておかなければならない設定上の問題点を以下に記します。

メールドメイン名 – プライマリネームサービスとして、NIS+ を設定していれば、sendmail は、NIS+ の sendmailvars テーブル (キーと値から構成される 2 列の NIS+ テーブル) からメールドメインを検索します。メールドメインを設定するには、1 つのエントリをこのテーブルに追加する必要があります。このエントリは、キーの列に文字列 maildomain が、値の列に自分のドメイン名 (たとえば、admin.acme.com) が設定されている必要があります。NIS+ では、sendmailvars テーブルに、どのような文字でも設定できますが、メールシステムが正常に機能するように接尾辞の規則が適用されます。nistbladm を使用して、maildomail エントリを sendmailvars テーブルに追加できます。たとえば、次のようになります。

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

メールドメインは NIS+ ドメインの接尾辞となることに注意してください。

mailhost ホスト名 – ネットワークがホストデータベースのソースとして NIS+ と DNS の両方を使用しているときは、mailhost エントリを NIS+ あるいは DNS ホストテーブルのいずれかに置くことができます。NIS+ と DNS をホストデータベースのソースとして /etc/nsswitch.conf ファイルで指定するようにしてください。

完全なホスト名 – NIS+ も DNS も完全なホスト名を認識します。通常の NIS+ と DNS の設定手順を踏めば、この項目の要件は満たされます。

ホストの完全名および短縮名のマッチング – NIS+ ホストテーブルの全ホストエントリに対して、対応するエントリが DNS に必要です。

1 つのメールドメイン内の複数の NIS ドメイン – この要件を満たすには、全ホストテーブルエントリをコピーするか、ネームサービスのドメイン中の全ホストのエントリをメールドメインレベルのマスターホストテーブルに入力する必要があります。

第 36 章

ネットワークサービスの監視についてのトピック

第 37 章	ネットワークサービスの監視の手順

第 37 章

ネットワーク性能の監視 (作業)

この章ではネットワークの性能を監視する方法について説明します。以下の項目ごとに説明します。

ネットワーク性能の監視

表 37–1 に、ネットワークの性能を監視するために使用できるコマンドを示します。

表 37–1 ネットワーク監視コマンド

コマンド	用途
`ping`	ネットワーク上でホストの応答を調べる
`spray`	送信したパケットサイズの信頼性を検査する。パケットが遅延されていないか、落とされていないか判定できる
`snoop`	ネットワークからパケットを捕捉し、各クライアントから各サーバーへの呼び出しを追跡する

表 37–1　ネットワーク監視コマンド　続く

コマンド	用途
netstat	TCP/IP トラフィックに使用されるインタフェースや IP ルーティングテーブルなどに関するネットワーク状態と、UPD、TCP、ICMP、および IGMP についてのプロトコル別の統計情報を表示する
nfsstat	NFS の問題を解析するのに使用できる、サーバーおよびクライアントの統計情報の要約を表示する

▼ ネットワーク上でホストの応答を検査する方法

ping コマンドを使用して、ネットワーク上のホストの応答を検査します。

```
$ ping hostname
```

物理的な問題が発生していると思われる場合は、ping コマンドを使用して、ネットワーク上にあるホストの応答時間を調べることができます。あるホストからの応答が期待していたものと異なる場合は、そのホストについて調査します。物理的な問題の原因としては次の理由が考えられます。

- ケーブルまたはコネクタの緩み
- 接地不良
- 終端処理の欠落
- 信号の反射

このコマンドの詳細については ping(1M) のマニュアルページを参照してください。

例 – ネットワーク上のホストの応答を検査する

最も簡単な ping コマンドの使い方は、ネットワーク上のホストへ 1 つのパケットを送信することです。正しい応答が受信されると、"*host* is alive" というメッセージが表示されます。

```
$ ping elvis
elvis is alive
```

-s オプションを指定すると、ping は 1 秒ごとにデータグラムをホストへ送り、次に各応答と、往復に要した時間を表示します。たとえば、次のように表示されます。

```
$ ping -s pluto
64 bytes from pluto (123.456.78.90): icmp_seq=0. time=10. ms
64 bytes from pluto (123.456.78.90): icmp_seq=5. time=0. ms
64 bytes from pluto (123.456.78.90): icmp_seq=6. time=0. ms
^C
----pluto PING Statistics----
8 packets transmitted, 8 packets received, 0% packet loss

round-trip (ms) min/avg/max = 0/2/10
```

▼ ネットワーク上でホストへパケットを送信する方法

spray コマンドを使用すると、送信したパケットサイズの信頼性を検査できます。

```
$ spray [ -c count -d interval -l packet_size] hostname
```

–c *count*	送信するパケット数
–d *interval*	パケットの送信ごとに一時停止するマイクロ秒数。遅延を使用しないと、バッファーを使い果たす可能性がある
–l *packet_size*	パケットサイズ
hostname	パケットを送信するシステム

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

例 – ネットワーク上のホストへパケットを送信する

次の例では、各パケットサイズが 2048 バイト (-l 2048) のパケット 100 個 (-c 100) を、ホストへ送信します。パケットは、各バースト間に 20 マイクロ秒の遅延時間 (-d 20) を入れて送信されます。

```
$ spray -c 100 -d 20 -l 2048 pluto
sending 100 packets of length 2048 to pluto ...
no packets dropped by pluto
279 packets/sec, 573043 bytes/sec
```

▼ ネットワークからパケットを捕捉する方法

ネットワークからパケットを捕捉し、各クライアントから各サーバーへの呼び出しを追跡するには、snoop コマンドを使用します。このコマンドは、ネットワークの性能の問題を素早く解析するための、正確なタイムスタンプを提供します。詳細は、snoop(1M) のマニュアルページを参照してください。

```
# snoop
```

パケットが落とされるのは、バッファーの領域不足か、CPU の過負荷が原因となっている場合があります。

▼ ネットワークの状態を調べる方法

netstat コマンドを使用すると、ネットワークインタフェースやルーティングテーブルなどに関するネットワーク状態と、各種プロトコルについての統計情報を表示できます。

```
$ netstat [-i] [-r] [-s]
```

–i	TCP/IP インタフェースの状態を表示する
–r	IP ルーティングテーブルを表示する
–s	UDP、TCP、ICMP、および IGMP プロトコルについての統計情報を表示する

詳細は、netstat(1M) のマニュアルページを参照してください。

例 – ネットワークの状態を調べる

次の表示例は、netstat -i コマンドの出力を示したものです。このコマンドは、TCP/IP トラフィックに使用されるインタフェースの情報を表示します。

```
$ netstat -i
Name  Mtu  Net/Dest    Address      Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 software    localhost     1280   0     1280     0       0    0
le0   1500 loopback    venus      1628480   0   347070    16   39354    0
```

上記の表示例は、マシンが各インタフェース上で送受信したパケット数を示しています。有効なネットワークトラフィックが存在するマシンでは、Ipkts と Opkts が継続的に増加しています。

ネットワーク衝突率は、衝突カウント (Collis) を出力パケットの数 (Opkts) で割ることにより算出できます。上記の例では、衝突率は 11%です。ネットワーク全体の衝突率が 5 ～ 10 %を超える場合には、問題が発生している可能性があります。

入力パケットエラー率 (Ierrs/Ipkts) は、入力エラー数を合計入力パケット数で割ることにより算出できます。出力パケットエラー率 (Oerrs/Opkts) は、出力エラー数を合計出力パケット数で割ることにより算出できます。入力エラー率が高い場合 (0.25% を超えている場合)、ホストがパケットを落としている可能性があります。

次に、netstat -s コマンドの出力例を示します。このコマンドは、UPD、TCP、ICMP、および IGMP についてプロトコル別の統計情報を表示します。

```
UDP
 udpInDatagrams      =196543 udpInErrors         =     0
 udpOutDatagrams     =187820

TCP
 tcpRtoAlgorithm     =     4 tcpRtoMin           =   200
 tcpRtoMax           = 60000 tcpMaxConn          =    -1
 tcpActiveOpens      = 26952 tcpPassiveOpens     =   420
 tcpAttemptFails     =  1133 tcpEstabResets      =     9
 tcpCurrEstab        =    31 tcpOutSegs          =3957636
 tcpOutDataSegs      =2731494 tcpOutDataBytes     =1865269594
 tcpRetransSegs      = 36186 tcpRetransBytes     =3762520
 tcpOutAck           =1225849 tcpOutAckDelayed    =165044
 tcpOutUrg           =     7 tcpOutWinUpdate     =   315
 tcpOutWinProbe      =     0 tcpOutControl       = 56588
 tcpOutRsts          =   803 tcpOutFastRetrans   =   741
 tcpInSegs           =4587678
 tcpInAckSegs        =2087448 tcpInAckBytes       =1865292802
 tcpInDupAck         =109461 tcpInAckUnsent      =     0
 tcpInInorderSegs    =3877639 tcpInInorderBytes   =-598404107
 tcpInUnorderSegs    = 14756 tcpInUnorderBytes   =17985602
 tcpInDupSegs        =    34 tcpInDupBytes       = 32759
 tcpInPartDupSegs    =   212 tcpInPartDupBytes   =134800
 tcpInPastWinSegs    =     0 tcpInPastWinBytes   =     0
 tcpInWinProbe       =   456 tcpInWinUpdate      =     0
 tcpInClosed         =    99 tcpRttNoUpdate      =  6862
```

(続く)

```
 tcpRttUpdate        =435097 tcpTimRetrans       = 15065
 tcpTimRetransDrop   =    67 tcpTimKeepalive     =   763
 tcpTimKeepaliveProbe=     1 tcpTimKeepaliveDrop =     0

IP
 ipForwarding        =     2 ipDefaultTTL        =   255
 ipInReceives        =11757234 ipInHdrErrors     =     0
 ipInAddrErrors      =     0 ipInCksumErrs       =     0
 ipForwDatagrams     =     0 ipForwProhibits     =     0
 ipInUnknownProtos   =     0 ipInDiscards        =     0
 ipInDelivers        =4784901 ipOutRequests       =4195180
 ipOutDiscards       =     0 ipOutNoRoutes       =     0
 ipReasmTimeout      =    60 ipReasmReqds        =  8723
 ipReasmOKs          =  7565 ipReasmFails        =  1158
 ipReasmDuplicates   =     7 ipReasmPartDups     =     0
 ipFragOKs           = 19938 ipFragFails         =     0
 ipFragCreates       =116953 ipRoutingDiscards   =     0
 tcpInErrs           =     0 udpNoPorts          =6426577
 udpInCksumErrs      =     0 udpInOverflows      =   473
 rawipInOverflows    =     0

ICMP
 icmpInMsgs          =490338 icmpInErrors        =     0
 icmpInCksumErrs     =     0 icmpInUnknowns      =     0
 icmpInDestUnreachs  =   618 icmpInTimeExcds     =   314
 icmpInParmProbs     =     0 icmpInSrcQuenchs    =     0
 icmpInRedirects     =   313 icmpInBadRedirects  =     5
 icmpInEchos         =   477 icmpInEchoReps      =    20
 icmpInTimestamps    =     0 icmpInTimestampReps =     0
 icmpInAddrMasks     =     0 icmpInAddrMaskReps  =     0
 icmpInFragNeeded    =     0 icmpOutMsgs         =   827
 icmpOutDrops        =   103 icmpOutErrors       =     0
 icmpOutDestUnreachs =    94 icmpOutTimeExcds    =   256
 icmpOutParmProbs    =     0 icmpOutSrcQuenchs   =     0
 icmpOutRedirects    =     0 icmpOutEchos        =     0
 icmpOutEchoReps     =   477 icmpOutTimestamps   =     0
 icmpOutTimestampReps=     0 icmpOutAddrMasks    =     0
 icmpOutAddrMaskReps =     0 icmpOutFragNeeded   =     0
 icmpInOverflows     =     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
```

次に、netstat -r コマンドの出力例を示します。このコマンドは、IP ルーティングテーブルを表示します。

```
Routing Table:
  Destination        Gateway           Flags  Ref   Use    Interface
------------------ -------------------- ----- ----- ------ ---------
localhost          localhost             UH       0   2817  lo0
earth-bb           pluto                 U        3  14293  le0
224.0.0.0          pluto                 U        3      0  le0
default            mars-gate             UG       0  14142
```

表 37–2 は、netstat -r コマンドが出力するレポート中のフィールドを説明します。

表 37–2 netstat -r コマンドの出力

フィールド名		説明
Flags	U	ルートが正常に動作している
	G	ルートはゲートウェイを経由する
	H	ルートはホスト宛である
	D	ルートはリダイレクトを使用して動的に作成された
Ref		同じリンク層を共有している現在のルート数を示す
Use		送信されたパケット数を示す
Interface		ルートに使用されるネットワークインタフェースを表示する

▼ NFS サーバーとクライアントの統計情報を表示する方法

NFS 分散型ファイルサービスは、ローカルコマンドをリモートホストへの要求に変換する、リモート手続き呼び出し (RPC) 機能を使用します。リモート手続き呼び出しは同期型の呼び出しです。つまり、サーバーが呼び出しを完了してその結果を返すまで、クライアントアプリケーションはブロックまたは中断されます。NFS の性能に影響を与える主要な要素の 1 つに再伝送率があります。

ファイルサーバーがクライアントの要求に応答できない場合、そのクライアントは、指定された回数だけ要求を再伝送して終了します。再伝送のたびにシステムにオーバーヘッドがかかり、ネットワークトラフィックが増加します。過度の再伝送はネットワークの性能を低下させます。再伝送率が高い場合は、次のような問題が発生していないか調べます。

- サーバーが過負荷になっており、要求の処理に時間がかかりすぎていないか
- Ethernet インタフェースがパケットを落としていないか
- ネットワークの輻輳によりパケットの伝送が低下していないか

表 37–3 に、クライアントとサーバーの統計情報を表示するための nfsstat コマンドのオプションとその説明を示します。

表 37–3 クライアントとサーバーの統計情報を表示するためのコマンド

指定するオプション	表示される情報
nfsstat -c	クライアントの統計情報
nfsstat -s	サーバーの統計情報
netstat -m	ファイルシステムごとのネットワーク統計情報

クライアントの統計情報を表示するには nfsstat -c を使用し、サーバーの統計情報を表示するには nfsstat -s を使用します。また、ファイルシステムごとのネットワークの統計情報を表示するには、nfsstat -m を使用します。詳細は、nfsstat(1M) のマニュアルページを参照してください。

例 – NFS サーバーとクライアントの統計情報を表示する

次の例は、クライアント pluto の RPC と NFS データを表示します。

```
$ nfsstat -c

        Client rpc:
Connection oriented:
calls    badcalls  badxids  timeouts newcreds  badverfs   timers
1595799  1511      59       297      0         0          0
cantconn nomem     interrupts
```

(続く)

```
1198      0         7
Connectionless:
calls    badcalls  retrans  badxids  timeouts  newcreds   badverfs
80785    3135      25029    193      9543      0          0
timers   nomem     cantsend
17399    0         0

Client nfs:
calls    badcalls  clgets   cltoomany
1640097  3112      1640097  0
Version 2: (46366 calls)
null     getattr   setattr  root     lookup     readlink  read
0 0%     6589 14%  2202 4%  0 0%     11506 24%  0 0%      7654 16%
wrcache  write     create   remove   rename     link      symlink
0 0%     13297 28% 1081 2%  0 0%     0 0%       0 0%      0 0%
mkdir    rmdir     readdir  statfs
24 0%    0 0%      906 1%   3107 6%
Version 3: (1585571 calls)
null    getattr    setattr  lookup     access     readlink  read
0 0%    508406 32% 10209 0% 263441 16% 400845 25% 3065 0%  117959 7%
write    create     mkdir    symlink    mknod    remove   rmdir
69201 4% 7615 0%    42 0%    16 0%      0 0%     7875 0%  51 0%
rename   link       readdir  readdir+   fsstat   fsinfo   pathconf
929 0%   597 0%     3986 0%  185145 11% 942 0%   300 0%   583 0%
commit
4364 0%
Client nfs_acl:
Version 2: (3105 calls)
null       getacl     setacl     getattr    access
0 0%       0 0%       0 0%       3105 100%  0 0%
Version 3: (5055 calls)
null       getacl     setacl
0 0%       5055 100%  0 0%
```

表 37–4 に、nfsstat -c コマンドの出力とその説明を示します。

表 37–4 nfsstat -c コマンドの出力とその説明

フィールド	説明
calls	送信された合計呼び出し数
badcalls	RPC によって拒否された合計呼び出し数
retrans	再伝送の合計数。このクライアントの場合、再伝送回数は 1% 未満 (6888 回の呼び出しのうち、10 回のタイムアウト)。これらの再伝送の原因としては一時的な障害の発生が考えられる。1% 以上の再伝送率の場合は、問題が発生している可能性がある

表 37–4　nfsstat -c コマンドの出力とその説明　続く

フィールド	説明
badxid	1 つの NFS 要求に対して重複する承認を受信した回数
timeout	タイムアウトした呼び出しの回数
wait	利用可能なクライアントハンドルがないために呼び出しが待機した回数
newcred	認証情報を書き換えなければならなかった回数
timers	タイムアウト値が、呼び出しに対して指定されたタイムアウト値以上であった回数
readlink	シンボリックリンクに対して読み取りが行われた回数。この値が大きすぎる (10% を超える) 場合は、シンボリックリンクが多すぎる可能性がある

次に、nfsstat -m コマンドの出力例を示します。

```
pluto$ nfsstat -m
/usr/man from pluto:/export/svr4/man
Flags: vers=2,proto=udp,auth=unix,hard,intr,dynamic,
       rsize=8192, wsize=8192,retrans=5
 Lookups: srtt=13 (32ms), dev=10 (50ms), cur=6 (120ms)
 All:     srtt=13 (32ms), dev=10 (50ms), cur=6 (120ms)
```

表 37–5 に、ミリ秒単位で表示される nfsstat -m コマンドの出力を示します。

表 37–5　nfsstat -m コマンドの出力

フィールド	説明
srtt	平準化された平均往復時間
dev	平均偏差
cur	現在の「予測」応答時間

ネットワークのハードウェアに問題の原因があると思われる場合は、ケーブルおよびコネクタをチェックしてください。

付録 **A**

PCNFSpro の障害追跡

次の障害追跡手法は Windows クライアントとして作動している PCNFSpro 専用です。次にこの付録の内容を示します。

障害追跡

次の障害追跡手法は、Windows クライアントとして動作している PCNFSpro 専用です。PCNFSpro および Windows クライアントについては、『*SolarNet PC-Admin Administrator's Guide*』を参照してください。次の手順について説明します。

PC の再起動

PC がサーバーに接続を試みた際に接続できないか、エラーメッセージが表示される場合は、まず再起動してください。マシンを再起動すると、ネットワークのハー

ドウェアとソフトウェアがリセットされます。期限が切れた一時ライセンスが問題の原因である場合には、再起動によってライセンスが 30 分間更新されます。

Windows クライアントの場合は、以下のファイルを削除します。

```
c:\pcnfspro\dhcp\interface.bin
```

interface は、使用中の実際のインタフェース名に置換します。たとえば、次のようにします。

```
c:\pcnfspro\dhcp\pk0.bin
```

デバッグモードでの実行

DHCP デバッグモードで実行すると、クライアントとサーバーとの間で進行中の大部分のダイアログが明らかになります。このダイアログは、ネットワークの問題を解決する有用な手がかりを与えてくれます。

▼ Windows クライアントをデバッグモードで実行する方法

1. **DHCP** サーバーを終了して、デバッグモードで再起動します。
2. 「**Configuration Tool**」**(**構成ツール**)** の「**Service applet**」**(**サービスアプレット**)** の中の「**Network Event Log**」**(**ネットワークイベントログ**)** を有効にします。
3. 「**Configuration Tool**」**(**構成ツール**)** を閉じます。
4. プログラムグループから「**Network Event Log**」**(**ネットワークイベントログ**)** を開始します。
5. 「**Display**」**(**表示**)** メニューを選択して、すべての優先レベルを強調表示します。
6. 「**Save**」**(**保存**)** を選択します。選択後は、`nfswdhcp.exe` がネットワークイベントログへの記録を行います。

7. **Windows** を終了して再起動します。

クライアントが **DHCP/BOOTP** サーバーとの接続に失敗する場合

新規クライアントをインストールまたは追加したが、そのマシンがサーバーへの接続に失敗してエラーメッセージが表示される場合は、マシンのケーブルとアダプタを検査します。アダプタに診断プログラムがある場合は、そのプログラムを実行して、考えられる問題を特定します。

PCNFSpro ディレクトリ内の構成アプリケーションを開始します。「Services」(サービス) を選択し、「Start Network Event Log」(ネットワークイベントログの開始) を有効にします。アイコンから直接ネットワークイベントログを開始することもできます。ネットワークイベントログの開始後、「Display」(表示) を選択し、次に「Configure」(構成) を選択します。最下位の「Debug」(デバッグ) まで、すべての優先レベルを選択します。「Save」(保存) を選択して構成を保存します。次の内容に類似するイベントログのエントリは、マシンが構成を求める要求を伝送済みであることを示しています。

```
DHCP: Attempting to configure interface using DHCP
```

サーバーの応答が後続します。

次に、サーバー上で in.dhcpd デーモンを終了し、**in.dhcp -d** と入力してサーバーを診断モードで動作させます。

出力を受け取った後で、マシンのサブネット上に DHCP サーバーまたはリレーエージェントが存在するかどうかを検査します。クライアントのブート時に、マシンと同じサブネット上にある任意のサーバーで以下を実行します。

```
snoop udp port 67 or udp port 68
```

システムが応答するかどうかを確認します。Windows クライアントの snoop の出力は、次のようになります。

```
OLD-BROADCAST -> BROADCAST UDP D=67 S=68 LEN=311
glrr -> BROADCAST UDP D=68 S=67 LEN=490
```

Windows クライアントの場合は、クライアント名は表示されません。DHCP の稼働は BROADCAST によって表されます。

アプリケーションが従来のメモリーを使い切った場合

CONFIG.SYS ファイルは、(デフォルトでは) Windows クライアントのアダプタドライバ (パケットドライバ、NDIS、ODI のいずれか) をアッパーメモリーへ読み込みません。この結果として、アプリケーションの開始時に、「メモリー不足です (not enough conventional memory)」または類似のメッセージが報告されることがあります。

DOS 5.x またはそれ以降を動作させている場合は、CONFIG.SYS 内の以下の行を変更します。

```
DEVICE=C:\PCNFSPRO\filename
```

上記の行を以下に示すように変更します。

```
DEVICEHIGH=C:\PCNFSPRO\filename
```

さらに、NFSWAUTO.BAT 内の、以下の TSR を読み込む行を変更します。

```
tsrname
```

上記の行を以下に示すように変更します。

```
LH tsrname
```

ホームディレクトリをマウントする場合

ソフトウェアのデフォルトとしてホームディレクトリをドライブ H 上にマウントすると、同じくドライブ H を使用する MS-DOS 6.2 の DoubleSpace ユーティリティが衝突します。この衝突を解決するには、DoubleSpace ユーティリティが使用するドライブの割り当てを変更する (dblspace h: /host=*new_drive*) か、またはユーザーサイトのログインスクリプトを変更してホームディレクトリを異なるドライブにマウントします。ユーザーサイトのログインスクリプトは、/opt/SUNWpcnet/1.5/site/pcnfspro/login.snc です。

サイトのログインスクリプトを変更する場合は、HOME 環境変数も新しいドライブに変更する必要があります。

ping の使用法

任意のマシンから、net use コマンドを使用してリモートファイルをマウントすることができない場合は、アクセスしたいと希望するファイルシステムへのパス名が適正に入力されていることを確認します。そのあとで ping コマンドを使用します。

ネットワークドライブをコピー先またはコピー元としてユーザーがファイルのコピーを行なっていて、コピーが完了する前に処理が停止する場合は、ping コマンドを使用します。

これまでライセンスを受け取ってきたマシンが、電源投入時または再起動時にライセンスを受け取ることができず、代わりにエラーメッセージを受け取る場合は、ping コマンドを使用します。

SNC スクリプト

マシンの起動時、または新規ユーザーがログインした際に、ソフトウェアによりアプリケーションおよびその他のネットワークサービスへのアクセスが配付されます。ソフトウェアは、store\login.snc ファイル内の命令を解釈します。%SNDRIVE% は、Windows クライアント用のサイトの SNC スクリプトディレクトリである /opt/SUNWpcnet/1.5/site/pcnfspro に展開されます。このスクリプトは UNIX のアクセス権によって保護されているため、ユーザーがアクセスすることはできません。

ディレクトリ /opt/SUNWpcnet/1.5/site/pcnfspro にはデフォルトのクライアントスクリプト (SNC スクリプト) が格納されていて、ブート時、ログイン時、およびログアウト時に使用されてマシン上の資源を制御します。ディレクトリは一時的にマウントされ、スクリプトが実行後にアンマウントされます。Windows クライアント上では、SNC コマンドがネットワークに対するユーザーの一意な関係を確立する役割を果たします。

Windows クライアントの構成プログラムは、SNC スクリプトディレクトリ /opt/SUNWpcnet/1.5/site/pcnfspro 内の起動スクリプト (デフォルトの名前は boot.snc) を処理します。スクリプトと SNC スクリプトディレクトリの名前は、ネットワークごとに異なることがあります。

Windows クライアントの構成プログラムは総合的なグラフィカルツールで、各種の構成パラメータの参照と変更を行う場合に使用します。このプログラムを使用すると、Windows クライアントのデフォルトの構成を変更してカスタマイズ、およびカスタマイズした構成を保存することができます。この構成プログラムを使用して制御できるパラメータには、TCP/IP、ローカルエリアネットワーク (LAN) のユーザー名、NFS システム、プリンタクライアント、NetBIOS、および SNMP があります。さらに、Windows クライアントの構成プログラムがユーザーサイトの複数のユーザーに対して使用可能に設定する構成のレベルを標準化して制御することもできます。構成プログラムの使用法については、プログラムのオンラインヘルプを参照してください。

新規グループの `login.snc` スクリプトに対する INCLUDE 命令を追加すると、サイトの SNC スクリプトディレクトリが拡張されます。INCLUDE 命令を `store\logout.snc` ファイル内に追加した場合にも、サイトの SNC スクリプトディレクトリは拡張されます。

スクリプトディレクトリを使用して、ネットワーク上のアプリケーションを独自に表示する各ユーザーの UNIX ログイン名を名前にしたディレクトリを作成することができます。次に、ユーザー固有の `login.snc` スクリプトと `logout.snc` スクリプトを作成して、それらのスクリプトを個々のディレクトリにコピーすることができます。以下のコマンドを使用します。

```
cd/opt/SUNWpcnet/1.5/site/pcnfspro
mkdir user1 user2 user3 user4 user5 user6 user7
```

この例では、*user8*、*user9*、*user10*、*user11* を除くすべてのユーザーが、その他のどのユーザーとも共有しないアプリケーション、または一部のユーザーとは共有しているがすべてのユーザーと共有してはいないアプリケーションを使用します。

SNC スクリプトディレクトリ `/opt/SUNWpcnet/1.5/site/pcnfspro` 内には、クライアントの種類ごとに、デフォルトの SNC スクリプトが 3 つ用意されています。用意されているスクリプトは以下のとおりです。

- `boot.snc` – Windows クライアントの構成プログラムおよびログインアプリケーション・ログアウトアプリケーションが使用する、サイトの起動スクリプト。
- `login.snc` – Windows クライアントのログインアプリケーション・ログアウトアプリケーションが使用する、サイトのログインスクリプト。
- `logout.snc` – Windows クライアントのログインアプリケーション・ログアウトアプリケーションが使用する、サイトのログアウトスクリプト。

デフォルトのサイトの logout.snc スクリプトは、サイトの login.snc スクリプトの後に再配置します。サイトの logout.snc スクリプトをコピーし、名前を変更して、独自のログインスクリプトに対応するように修正することができます。すべてのログアウトスクリプト logout.snc の名前を、ログインスクリプトと平行して付けてください。ログアウトスクリプトを、SNC スクリプトディレクトリ /opt/SUNWpcnet/1.5/site/pcnfspro の下に作成したユーザー固有ディレクトリとグループ固有ディレクトリ内に配置します。

DHCP データベース

NIS+ ドメインごとに dhcptab テーブルが 1 つあります。このテーブルには、DHCP (または BOOTP) クライアントに戻す構成パラメータが定義されます。/opt/SUNWpcnet/1.5/site/pcnfspro スクリプトディレクトリもエントリの 1 つで、ユーザーの表示の定義とアプリケーションの配付用に使用します。

ライセンスのアップグレード

ライセンスのアップグレードファイルが作成されたことを検査して確認する場合には、Windows クライアントでは特定の手順が必要です。最初に C:\pcnfspro\upgrade を実行します。ライセンスのアップグレードファイルとその内容を検査してから、install プログラムを再実行します。その後で以下の手順を実行します。

1. C:\windows\pcnfswin.ini の名前を変更します。
2. C:\pcnfspro\binnfswpupg プログラムを実行します。
3. 最後に、名前を変更した pcnfswin.ini ファイルを、元の名前の pcnfswin.ini に戻します。

アップグレードされたライセンスファイルが存在して適正に動作していることを確認するためのその他の手順を実行した後で、古い DHCP 構成ファイルを削除し、マシンを再起動します。ディレクトリ C:\pcnfspro\dhcp 内の、インタフェースに対応するファイルを削除します。

ホスト名と IP アドレスが失われた場合

ホスト名と IP アドレスがマシンから失われた場合は、類似した手順を実行する必要があります。最初に、「Control Panel」(コントロールパネル) アプリケーションを実行して「Network」(ネットワーク) アイコンを選択します。現在のホスト名と IP アドレス、さらに関連情報を参照します。`dhcp_table` にエントリを作成する手順を実行します。その後で `C:\pcnfspro\dhcp` を実行し、前述の手順を実行します。

次に、サーバー上のインストールディレクトリ内のアップグレードディレクトリにアップグレードファイルをコピーします。`snaddpcs` スクリプトが実行されたことを検査します。その後で、古い DHCP 構成ファイルを削除し、マシンを再起動します。DHCP 構成ファイルの削除は、ディレクトリ `C:\pcnfspro\dhcp` 内の、インタフェースに対応するファイルを削除することによって行います。

アプリケーションの配付

マシンの起動時、または新規ユーザーがログインした際に、ソフトウェアによりアプリケーションおよびその他のネットワークサービスへのアクセスが配付されます。Windows クライアントの場合、これらのアプリケーションとサービスはログインアプリケーションによって各ユーザーへ供給されます。ログインアプリケーションは、Windows の開始時に自動的に開始するアプリケーションです。

Windows ベースの DHCP アプリケーションが DHCP を開始し、マシンの IP アドレスと関連ネットワーク情報を受け取り、マシンのスタックとサービスを構成します。その後で、クライアントの (SNC) スクリプトの処理を開始してファイルシステムをマウントし、共有される資源、グループ、個別ユーザー、および個別マシンの各検索パスを設定します。

ログインおよびログアウト

Windows クライアントの場合、ネットワークへのログインとネットワークからのログアウトは、ログインアプリケーションやログアウトアプリケーションのアイコンをクリックすると表示される Windows ダイアログボックスに記入を行うことによって行われます。ユーザーがダイアログボックスにユーザー名とパスワードを記入して OK をクリックするか、`Enter` キーを押すと、クライアントのサイトの起

動スクリプト内の以下に示す INCLUDE 命令によって、Windows クライアントのサイトの logon.snc が起動します。

```
INCLUDE  %SNDRIVE%\PCNFSPRO\LOGIN.SNC
```

このスクリプトはディレクトリをマウントし、特定の環境変数を設定し、ログインするために使用しているマシンとは関係なく、すでにログインしているユーザー用にその他のサービスを提供します。このスクリプトは、ネットワーク内のすべてのユーザーが従うログイン手順を確立します。

Windows ベースの同じダイアログボックスを使用して Windows クライアントがログアウトすると、クライアントのサイトの logout.snc スクリプトが処理されます。このスクリプトは、 login.snc スクリプトが実行したことをリセットします。ログアウトアプリケーションは、ファイルシステムとプリンタをアンマウントし、マシンの環境変数をリセットしてログイン前の値に戻します。

Windows クライアントのログインアプリケーションおよびログアウトアプリケーションは、ユーザーのネットワーク設定についての詳細情報を持っているため、これらのアプリケーションを使用して設定を変更することができます。変更できる設定には、バージョンおよびライセンス番号、ユーザー名およびユーザー ID、グループ ID、NIS ドメインおよび DNS ドメイン、サブネットマスク、MAC アドレス (Ethernet 通信アダプタを特定します)、タイムゾーン、終端ドライブ、使用可能なサーバー名と IP アドレスがあります。

付録 **B**

NFS の調整機能

NFS サービスの機能を高めるため、いくつかのパラメータを設定できます。これらのパラメータは /etc/system で定義することができ、システムのブート時に読み込まれます。各パラメータは、それが入っているカーネルモジュール名と、それを識別する記号名で識別できます。

注 - 記号の名前、それらが常駐するモジュールおよびデフォルト値は、リリースごとに異なる場合があります。変更を加えたり、前のリリース版の値を適用するときは、あらかじめ適切な SunOS バージョンの関連ドキュメントを確認してください。

次にこの章の内容を示します。

表 B–1 nfs モジュール用の NFS パラメータ

記号名	説明	デフォルト設定
nfs_32_time_ok	この記号は NFS クライアントまたはサーバーが Y2038 以上のファイル時刻表示を可能にするかどうかを決定する	デフォルトはオフ (0)。いずれかのファイルの時刻表示が負の場合で、それでもファイルにアクセスしたい場合はこの記号を使用する。この時刻表示が負の場合は日付が 1970 年より前であることを示している
nfs_acl_cache	この記号は NFS_ACL プロトコルを使用中のクライアントで、ACL をキャッシュするかどうかを制御する	デフォルトはオフ (0)。これは安全に有効 (1) 設定でき、Solaris の将来のリリースではオンがデフォルトになる予定
nfs_cots_timeo	NFS バージョン 2 クライアントによる接続指向トランスポートに対する操作の、デフォルトのタイムアウト	600 × 1/10 秒
nfs3_cots_timeo	NFS バージョン 3 クライアントによる接続指向トランスポートに対する操作の、デフォルトのタイムアウト	600 × 1/10 秒
nfs_do_symlink_cache	この記号は NFS バージョン 2 を使用してマウントしたファイルシステムで、シンボリックリンクをキャッシュするかどうかを制御する	デフォルトはオン (1)。システムで amd などを使用する場合には無効 (0) にできる。これが無効に設定されていると、クライアントシステムの性能が低下する場合がある
nfs3_do_symlink_cache	この記号は NFS バージョン 3 を使用してマウントしたファイルシステムで、シンボリックリンクをキャッシュするかどうかを制御する	デフォルトはオン (1)。これは無効 (0) に設定できるが、クライアントシステムの性能が低下する可能性がある
nfs_dynamic	この記号は NFS バージョン 2 を使用してマウントしたファイルシステムで、動的再転送のサポートを使用するかどうかを制御する	デフォルトはオン (1)。 安全にオフ (0) にできるが、処理の遅いサーバーや 8K バイトの読み書き転送を完全にサポートできないサーバーでは、相互運用に問題が生じる可能性がある

表 B–1 nfs モジュール用の NFS パラメータ 続く

記号名	説明	デフォルト設定
nfs3_dynamic	この記号は NFS バージョン 3 を使用してマウントしたファイルシステムで、動的再転送のサポートを使用するかどうかを制御する	デフォルトはオフ (0)。これは変更しないこと
nfs_lookup_neg_cache	この記号は NFS バージョン 2 を使用してマウントしたファイルシステムで、失敗した検索要求をキャッシュさせるかどうかを制御する	デフォルトはオフ (0)。安全に有効 (1) に設定できるが、正常なディレクトリ名のキャッシュ処理に悪影響が生じることもある
nfs3_lookup_neg_cache	この記号は NFS バージョン 3 を使用してマウントしたファイルシステムで、失敗した検索要求をキャッシュさせるかどうか制御をする	デフォルトはオフ (0)。安全に有効 (1) に設定できるが、正常なディレクトリ名のキャッシュ処理に悪影響が生じる可能性がある
nfs_max_threads	この記号は NFS バージョン 2 を使用してマウントしたファイルシステムごとに、非同期スレッドを起動する最大数を制御する	デフォルトは 8。この数字はファイルシステムごとのスレッド数に影響するので、ファイルシステムの多いクライアントでは、大きな変更を行うと性能を大幅に劣化させる原因になる
nfs3_max_threads	この記号は NFS バージョン 3 を使用してマウントしたファイルシステムごとに、非同期スレッドを起動させる最大数を制御する	デフォルトは 8。この数字はファイルシステムごとのスレッド数に影響するので、ファイルシステムの多いクライアントでは、大きな変更を行うと性能を大幅に劣化させる原因になる
nfs3_max_transfer_size	この記号は NFS バージョン 3 のクライアントファイルのブロックサイズを制御する	デフォルトは 32K バイト。変更はできるだけ行わないこと
nfs_nra	この記号は NFS バージョン 2 を使用してマウントしたファイルシステムで、読み込まれる先読みブロックの数を制御する	デフォルトは 4。値を大きくしても性能は向上せず、クライアント側でのメモリー利用が増大する

表 B–1 nfs モジュール用の NFS パラメータ 続く

記号名	説明	デフォルト設定
nfs3_nra	この記号は NFS バージョン 3 を使用してマウントしたファイルシステムについて、読み込まれる先読みブロックの数を制御する	デフォルトは 4。値を大きくしても性能は向上せず、クライアント側でのメモリー利用が増大する
nrnode	この記号はキャッシュされる NFS rnode の数を制御する	この記号に割り当てられる値はブート時に構成され、サーバーに適合するようスケーリングされる。1 に設定してキャッシュ処理を無効にできる
nfs_shrinkreaddir	この記号は回線上で、NFS バージョン 2 READDIR 要求を 1024 バイトに縮小するかどうかを制御する。一部古い NFS バージョン 2 のサーバーの中には、1024 バイト以上の READDIR 要求を正しく処理できないものがある	デフォルトはオフ (0) で、この場合 READDIR 要求を削減しない。これは安全に有効 (1) に設定できるが、ディレクトリの読み込み中に性能に悪影響を与える恐れがある
nfs_write_error_interval	この記号は NFS ENOSPC によるエラーメッセージの書き込みがログされる頻度を、秒単位で制御する	デフォルトは 5
nfs_write_error_to_cons_only	この記号は NFS 書き込みのエラーメッセージが、システムコンソールか、システムコンソールおよび syslog にログされるかどうかを制御する	デフォルトはオフ (0) で、この場合 NFS 書き込みエラーメッセージをすべて、システムコンソールと syslog にログする。この機能を有効 (1) に設定すると、NFS 書き込みエラーメッセージのほとんどが、システムコンソールでしか出力されないことになる

表 B–2 nfssrv モジュール用の NFS パラメータ

記号名	説明	デフォルト設定
nfs_portmon	この記号は IP ポート番号に基づき、NFS サーバーで要求のフィルタ処理を行うかどうかを制御する。予約済みポート番号のバークレー表記法を使用する	デフォルトはオフ (0)。有効 (1) にできるが、相互運用上の問題が発生する恐れがある
nfsreadmap	この記号は現在ではアクティブではなく、今ではマップの読み込みは実装していない。移行を容易にするため残されている	デフォルトはオフ (0)
rfs_write_async	この記号は書き込みの処理能力を安全に高めるため、NFS バージョン 2 サーバーが、書き込みのクラスタ化機能を使用するかどうかを制御する	デフォルトはオン (1)。無効 (0) に設定できるが、性能は低下する場合がある

表 B–3 rpcmod モジュール用の NFS パラメータ

記号名	説明	デフォルト設定
svc_ordrel_timeout	カーネルが接続を強制的に切断するまでの時間 (ミリ秒)。カーネル RPC に使用されている TCP 接続が、万一終了の途中でハングした場合に使用される。接続がハングする原因には、NFS サーバーが接続に対する正常なクローズ (FIN) を試み、クライアントがそのクローズの完了 (FIN 確認) ハンドシェークに失敗した場合が考えられる	デフォルトは 600,000 ミリ秒 (10 分)。値が小さすぎると、TCP 接続を閉じるのに十分な時間がクライアントに与えられない。大きすぎると、「欠陥のある」クライアントや害を及ぼすクライアントがサーバーと TCP 接続を結ぶことになる

表 B–4 rpcsec パラメータ用の NFS パラメータ

記号名	説明	デフォルト設定
authdes_cachesz	この記号は authdescache の大きさを決定する。これは性能を向上させる機能で、この機能を使用するとセキュリティ保護された RPC 要求が発生するたびにクライアントの資格を確認する必要がなくなる	デフォルトは 128
authdes_win	この記号は、AUTH_DES を使用しているときにサーバーとクライアントの間で許容されるクロックスキューの時間を決定する	デフォルトは 300 秒
authkerb_cachesz	この記号は authkerb キャッシュの大きさを決定する	デフォルトは 128。この値を高くし過ぎるとシステムの性能が低下する場合がある
authkerb_win	この記号は、AUTH_KERB を使用しているときにサーバーとクライアントの間で許容されるクロックスキューの時間を決定する	デフォルトは 300 秒
clnt_authdes_cachesz	この記号は、クライアント上の sec=ch 認証ハンドルのためのキャッシュテーブルの大きさを決定する	デフォルトは 64

カーネルパラメータの値を設定する方法

1. **root** になります。
2. /etc/system ファイルを編集し、行を追加してパラメータを設定します。

   それぞれのエントリは次の形式に従ってください。

   set *module*:*symbol*=*value*

   module は、必要なパラメータを含むカーネルモジュール名で、*symbol* はパラメータ名、*value* はパラメータに割り当てる数値です。以下に例を示します。

```
set nfs:nfs_nra=4
```

これで NFS バージョン 2 を使用してマウントしたファイルシステムについて、読み込まれる先読みブロック数が変更になります。/etc/system については、system(4) のマニュアルページを参照してください。

3. システムをリブートします。

用語集

この用語集は、本書で使用されている新しい用語だけを定義するものであり、汎用的な用語集ではありません。この用語集に含まれていない用語の定義については、`http://docs.sun.com:80/ab2/coll.417.1/GLOBALGLOSS/@Ab2TocView` にアクセスして、"Global Glossary" を参照してください。

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。

MTU 最大転送単位。リンク上で転送できるオクテット単位のパケットサイズ。たとえば、Ethernet の MTU は 1500 オクテット。

SADB セキュリティアソシエーションデータベース。データの伝送に使用される暗号キーおよびアルゴリズムを指定するテーブル。

SPI セキュリティパラメータインデックス。受信したパケットを複合化するために受信者が使用する、SADB 内の行を指定する整数。

VPN 仮想私設ネットワーク。インターネットなどの公衆ネットワークにまたがるトンネルを使用する、単一で安全な論理ネットワーク。

移動体 **(**モビリティ**)** **IP** IP アドレスを変更することなく、あるリンクから他のリンクへ移動することができるノード。

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。
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。
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	IPv6 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。
近傍探索	接続されているリンク上にあるルーター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	近傍ノードのデータリンク層アドレスを決定するため、あるいは、近傍ノードがキャッシュ内にあるデータリンク層アドレスの有効性を確認するためにノードが送る要求メッセージ。
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。
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	あるホストから他のホストへのセキュリティ特性を指定するための情報。

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 3 ホップ離れている、という。
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。
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。
認証ヘッダー	IPv6 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。
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。
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。
マルチキャストアドレス	インタフェースのグループを指定する IP アドレスで、マルチキャストアドレスに送られたパケットは、グループ内のすべてのインタフェースに配送される。
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次のホップノードを、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	(サブネットまたは加入者ネットワーク内で) 経路制御スコープがローカルに限定されているユニキャストアドレスで、ローカルまたはグローバルな一意性スコープを持つことができる。

索引

記号

数字

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

あ

か

き

く

け

こ

さ

し

す

な

に

ね

の

は

ほ

ま

も

る

れ

ろ

わ